UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
09/611,775	7013482	2134	9200

OC000000057515915

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 107299 on 11/06/2012

- **Correspondence Address**
- **Maintenance Fee Address**

The address of record for Customer Number 107299 is:

107299
Alan R. Loudermilk
511 N. Washington Ave
Marshall, TX 75670

PART 1 - ATTORNEY/APPLICANT COPY

page 1 of 1

10

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 12/29/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER
SIMITOSKI, MICHAEL J

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 12/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
09/611,775	7/7/00	Krumel	

EXAMINER
Michael J. Simitoski

ART UNIT	PAPER
2134	12192005

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

The information disclosure statement (IDS) submitted on 12/05/2005 was filed after the mailing date of the Notice of Allowance on 9/27/2005. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner. The drawings submitted 12/5/2005 are acceptable and overcome any previous objections. The amendments to the claims however, is not considered because the amendments constitute a change in the scope of the claims. For instance, regarding claims 1 & 31, "by the time the end portion of the packet is received" is considered to be substantially equivalent to "at the instant the end portion becomes fully received". However, "by a time when the end portion of the packet is received" can be any time after. Regarding claims 20, 37, 41 & 50, the amendatory language would require further search and consideration.

David Y. Jung
Primary Examiner

David Y. Jung
Primary Examiner

12/23/05

Sheet 1 of 1

Form PTO-1449 (REV. 7-92)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use several sheets if necessary)

Attorney's Docket Number: 802-001

Serial No.: 09/611,775

Applicant(s): Krumel

Filing Date: 7/7/00

Group Art Unit: 2134

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6	7	0	0	8	9	1	03/02/04	Wong	370	401	

EXAMINER

DATE CONSIDERED 12/19/05

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Do not enter. 12/19/05

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT PURSUANT TO RULE 312

Sir or Madam:

In response to the notice of allowance mailed September 27, 2005, please re-examine the above-identified application in view of the following amendment and remarks. The issue fee transmittal, substitute formal drawings and an IDS accompany this submission.

IN THE TITLE:

Please change the title to:

--METHODS FOR PACKET FILTERING INCLUDING PACKET INVALIDATION IF PACKET VALIDITY DETERMINATION NOT TIMELY MADE--

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 12/09/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 12/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Supplemental Notice of Allowability	**Application No.** 09/611,775	**Applicant(s)** KRUMEL, ANDREW K.	
	Examiner Michael J. Simitoski	**Art Unit** 2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *RCE of 7/28/2005*.

2. ☒ The allowed claim(s) is/are *1-66*.

3. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a) □ All b) □ Some* c) □ None of the:

   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. □ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

   (a) □ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

   1) □ hereto or 2) □ to Paper No./Mail Date _____.

   (b) □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. □ Notice of References Cited (PTO-892)
2. □ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. □ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. □ Notice of Informal Patent Application (PTO-152)
6. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. □ Examiner's Statement of Reasons for Allowance
9. □ Other _____.

SUPPLEMENTAL EXAMINER'S AMENDMENT

1. The IDS and response of 7/28/2005 was received and considered.

2. Claims 1-66 are allowed, a Notice of Allowance was mailed 9/27/2005.

3. An examiner's informal supplemental amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

The application has been amended as follows:

In claim 16: Please replace "The method of claim 16" (in line 1 of the claim) to "The method of claim 15".

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Simitoski whose telephone number is (571) 272-3841. The examiner can normally be reached on Monday - Thursday, 6:45 a.m. - 4:15 p.m.. The examiner can also be reached on alternate Fridays from 6:45 a.m. – 3:15 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached at (571) 272-3838.

Any response to this action should be mailed to:
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Or faxed to:
(571) 273-8300
(for formal communications intended for entry)
Or:
(571) 273-3841 (Examiner's fax, for informal or draft communications, please label "PROPOSED" or "DRAFT")

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MJS
November 28, 2005

GREGORY MORSE
[illegible]

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT PURSUANT TO RULE 312

Sir or Madam:

In response to the notice of allowance mailed September 27, 2005, please re-examine the above-identified application in view of the following amendment and remarks. The issue fee transmittal, substitute formal drawings and an IDS accompany this submission.

IN THE TITLE:

Please change the title to:

--METHODS FOR PACKET FILTERING INCLUDING PACKET INVALIDATION IF PACKET VALIDITY DETERMINATION NOT TIMELY MADE--

IN THE CLAIMS:

1. (currently amended) A method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets, the method comprising the steps of:

receiving a packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are ~~performing~~ performed in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet, wherein the packet is selectively altered to be invalid if a determination has not been made as to whether the packet is valid or invalid by ~~the~~ a time when the end portion of the packet is received.

2. (original) The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. (original) The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. (original) The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. (original) The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. (original) The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. (original) The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. (original) The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. (original) The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. (original) The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. (original) The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. (original) The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. (original) The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. (original) The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. (original) The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. (currently amended) The method of claim ~~16~~ 15, wherein the one or more network ~~address~~ addresses ~~comprises~~ comprise an IP address for the external computing system and/or the internal computing system.

17. (original) The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. (currently amended) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has ~~rejected~~ invalidated one or more packets.

21. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. (original) The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. (currently amended) The method of claim 23, wherein the one or more visual or audio feedback devices are reset by ~~the~~ a state of a physical switch.

25. (currently amended) The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of ~~the~~ an operation or status of the system performing the packet filter process.

26. (original) The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. (original) The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. (original) The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. (original) The method of claim 25, wherein the at least one light source comprises an LED.

30. (original) The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. (currently amended) A system for filtering packets of data between at least an external network and an internal network, wherein data is transmitted and received in the form of a plurality of packets, comprising:

a first interface circuit for coupling data packets to and from the external network;

a second interface circuit for coupling data packets to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria, wherein the packet is selectively altered to be invalid if a determination has not been made as to whether the packet is valid or invalid by ~~the~~ a time when the end portion of the packet is received.

32. (original) The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. (currently amended) The system of claim 31, wherein the programmable logic circuit includes at least ~~first~~ a logic portion for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the ~~plurality of~~ filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. (original) The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. (currently amended) The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and one or more signals from the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. (original) The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. (currently amended) The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by ~~a~~ the time when the end portion of

the packet ~~has been~~ is received, then the packet is selectively altered by the programmable logic device to be invalid.

40. (currently amended) The system of claim 31, wherein the packet is subjected to the ~~plurality of~~ filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid ~~by a~~ before the time when the end portion of the packet ~~has been~~ is received.

41. (currently amended) The system of claim 31, wherein the packet is subjected to the ~~plurality of~~ filtering criteria in real time ~~with~~ while the packet being received and transmitted between the first and second interface circuits.

42. (original) The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. (original) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. (previously amended) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a server mode.

45. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. (original) The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. (currently amended) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has ~~rejected~~ invalidated one or more packets.

51. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. (original) The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. (currently amended) The system of claim 53, wherein the one or more visual or audio feedback devices are reset by ~~the~~ a state of a physical switch.

55. (currently amended) The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of ~~the~~ an operation or status of the system.

56. (original) The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. (original) The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. (original) The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. (original) The system of claim 55, wherein the at least one light source comprises an LED.

60. (original) The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. (currently amended) The system of claim 36, wherein the one or more stateful filtering ~~criteria~~ criterion are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. (original) The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. (original) The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. (original) The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. (original) The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. (original) The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

REMARKS

Claims 1-66 are in the application and have been allowed. Herein Applicant is correcting certain typographical errors, informalities, etc., that were noted during a final review of the claims. Applicant also reviewed the drawings based on the originally-filed informal drawings and is herewith submitting substitute formal drawings.

No new matter has been added.

Applicant also is submitting an additional prior art reference cited in one of the related applications (09/745,599) (these applications were referenced in a previous amendment). The cited reference, however, among other distinctions, does not disclose or suggest the packet being selectively altered to be invalid if a determination has not been made as to whether the packet is valid or invalid by a time when the end portion of the packet is received. Thus, the presently pending claims remain allowable, and consideration of the IDS is respectfully requested.

Entry of this amendment is requested..

If there are any questions regarding the foregoing, Applicant's attorney requests an opportunity to discuss such matters with the Examiner by way of another interview, either in-person or by telephone.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251. No new matter has been added.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

December 2, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated above.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax **(571) 273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 09/27/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

12/06/2005 WABDELR3 00000082 500251 09611775

01 FC:2501 700.00 DA

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

ALAN R. LOUDERMILK	(Depositor's name)
	(Signature)
12/2/05	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

TITLE OF INVENTION: REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$0	$700	12/27/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
SIMITOSKI, MICHAEL J	2134	726-013000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 LOUDERMILK + ASSOCIATES
2 ____________
3 ____________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE
802 SYSTEMS LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY)
CHICAGO, IL

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual ☒ Corporation or other private group entity □ Government

4a. The following fee(s) are enclosed:

☒ Issue Fee
□ Publication Fee (No small entity discount permitted)
□ Advance Order - # of Copies ____________

4b. Payment of Fee(s):

□ A check in the amount of the fee(s) is enclosed.
□ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number 50-0251 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. □ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________ Date 12/2/05

Typed or printed name ALAN R. LOUDERMILK Registration No. 32,788

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel

Serial No.: 09/611,775

Filed: July 7, 2000

For: Real Time Firewall/Data Protection Systems and Methods

Examiner: Simitoski, Michael J.

Group Art Unit: 2134

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMISSON OF FORMAL DRAWINGS

Sir or Madam:

In response to the notice of allowance mailed September 27, 2005, Applicant is herewith submitting 14 sheets of substitute formal drawings for the above-identified application.

No new matter has been added.

Please contact the undersigned if there are any questions regarding the foregoing.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

December 2, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated above.

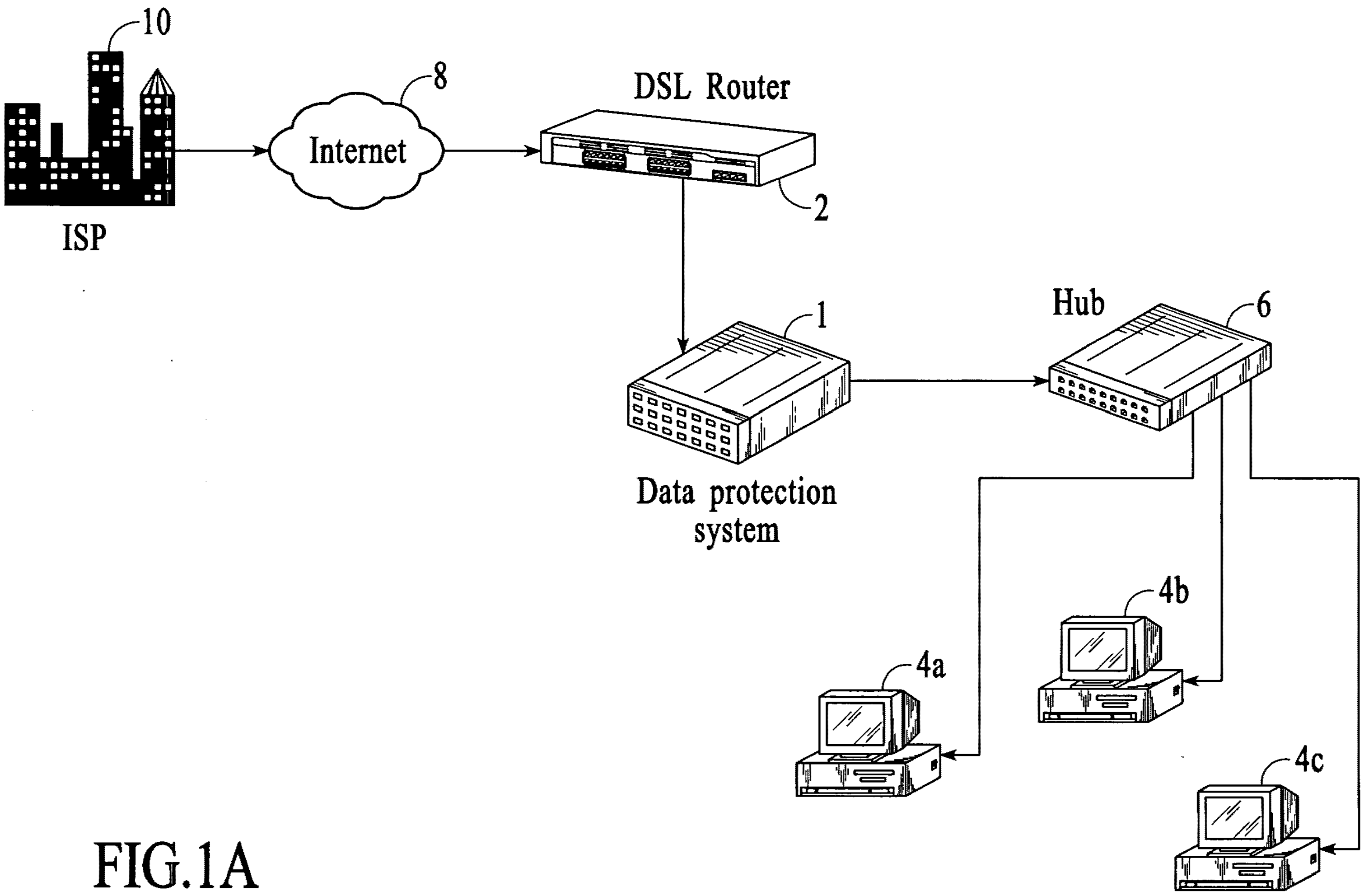

10
ISP
8
Internet
DSL Router
2
1
Data protection system
Hub
6
4a
4b
4c

FIG.1A

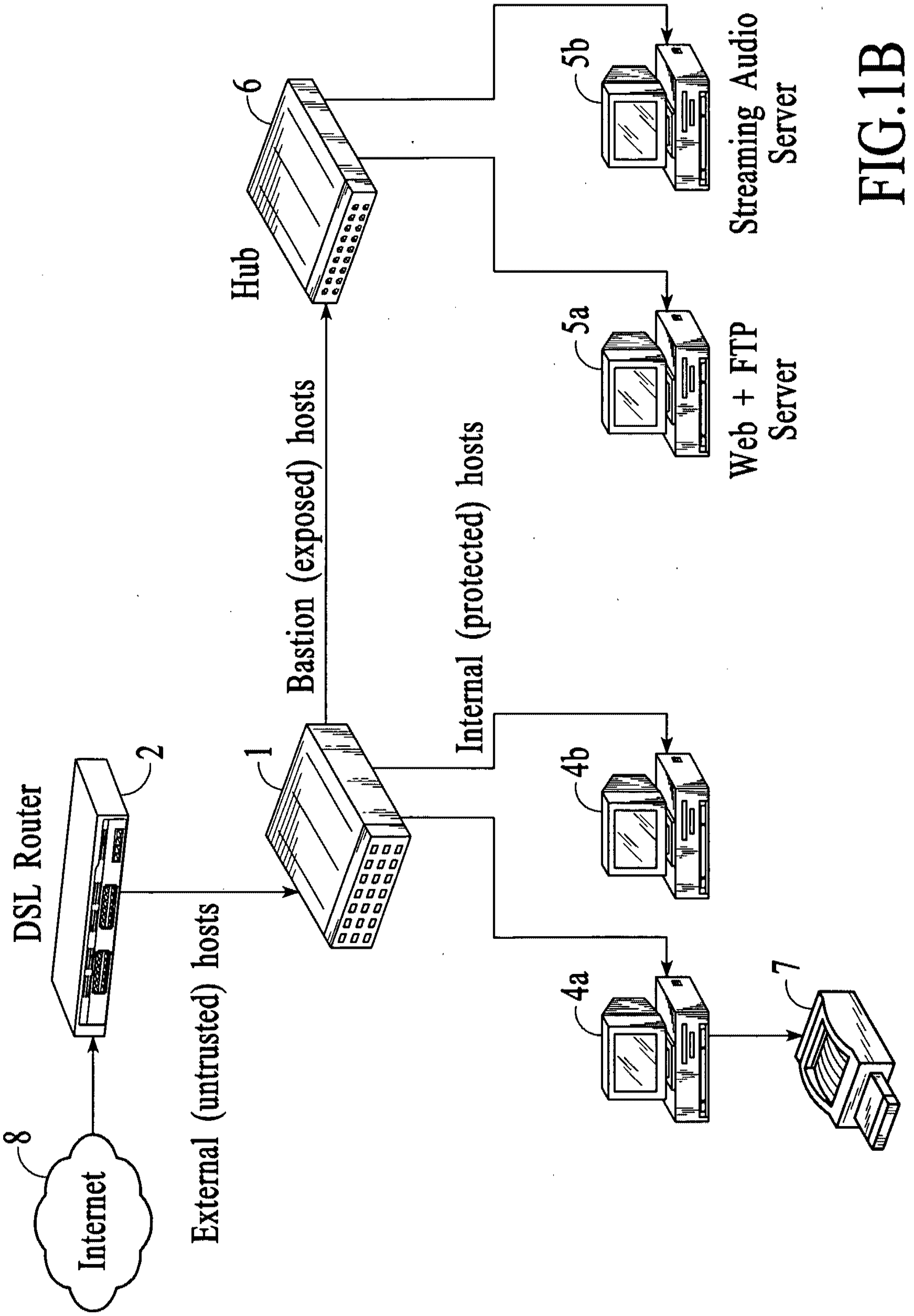

Internet
8
DSL Router
2
External (untrusted) hosts
1
Bastion (exposed) hosts
Internal (protected) hosts
Hub
6
4a
7
4b
5a
Web + FTP Server
5b
Streaming Audio Server

FIG.1B

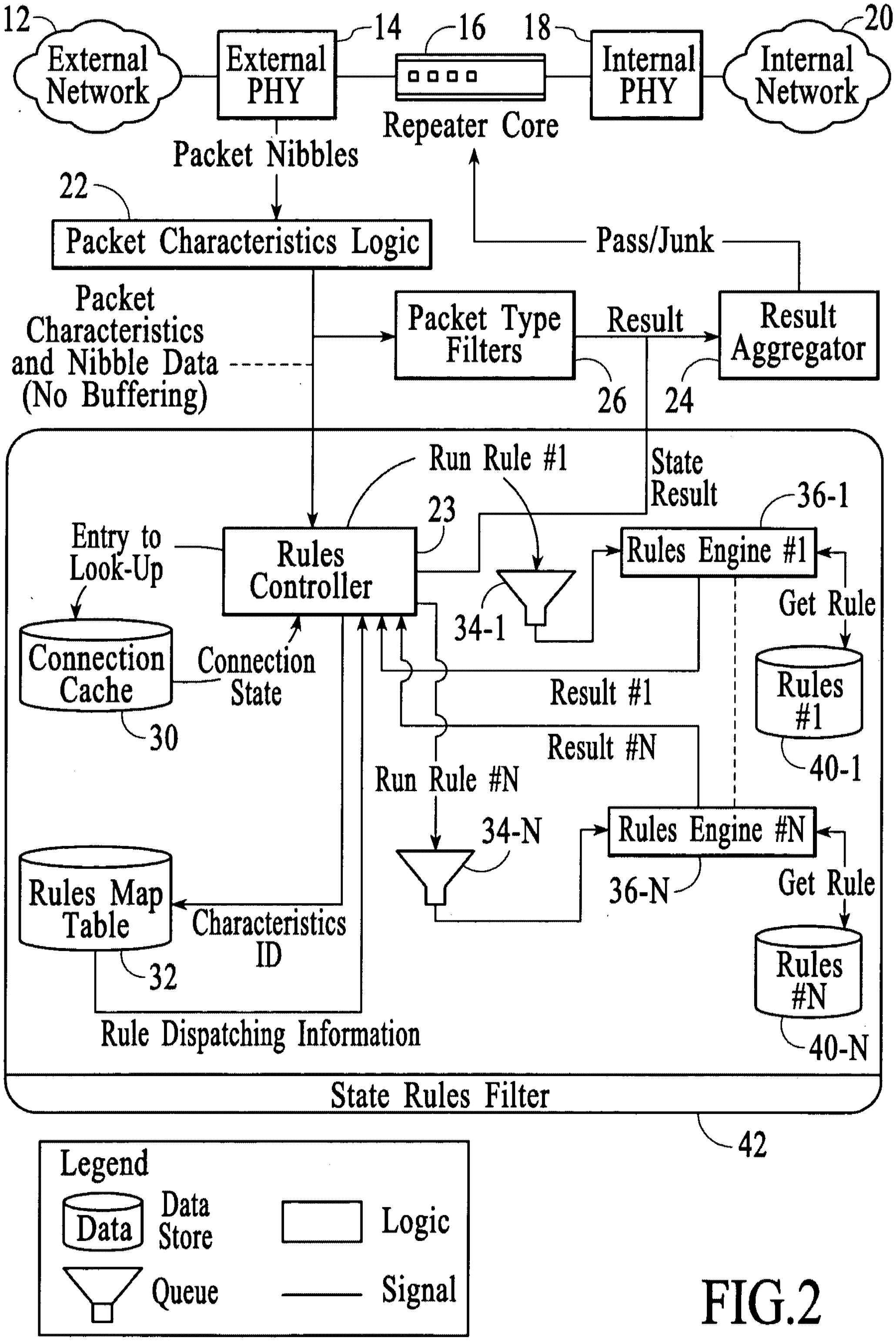

12
External Network
External PHY
14
16
18
Repeater Core
Internal PHY
Internal Network
20
Packet Nibbles
22
Packet Characteristics Logic
Pass/Junk
Packet Characteristics and Nibble Data (No Buffering)
Packet Type Filters
Result
Result Aggregator
26
24
Run Rule #1
State Result
23
36-1
Entry to Look-Up
Rules Controller
Rules Engine #1
34-1
Get Rule
Connection Cache
Connection State
Rules #1
30
Result #1
Result #N
40-1
Run Rule #N
34-N
Rules Engine #N
Rules Map Table
Characteristics ID
36-N
Get Rule
32
Rules #N
Rule Dispatching Information
40-N
State Rules Filter
42
Legend
Data
Data Store
Logic
Queue
Signal

FIG.2

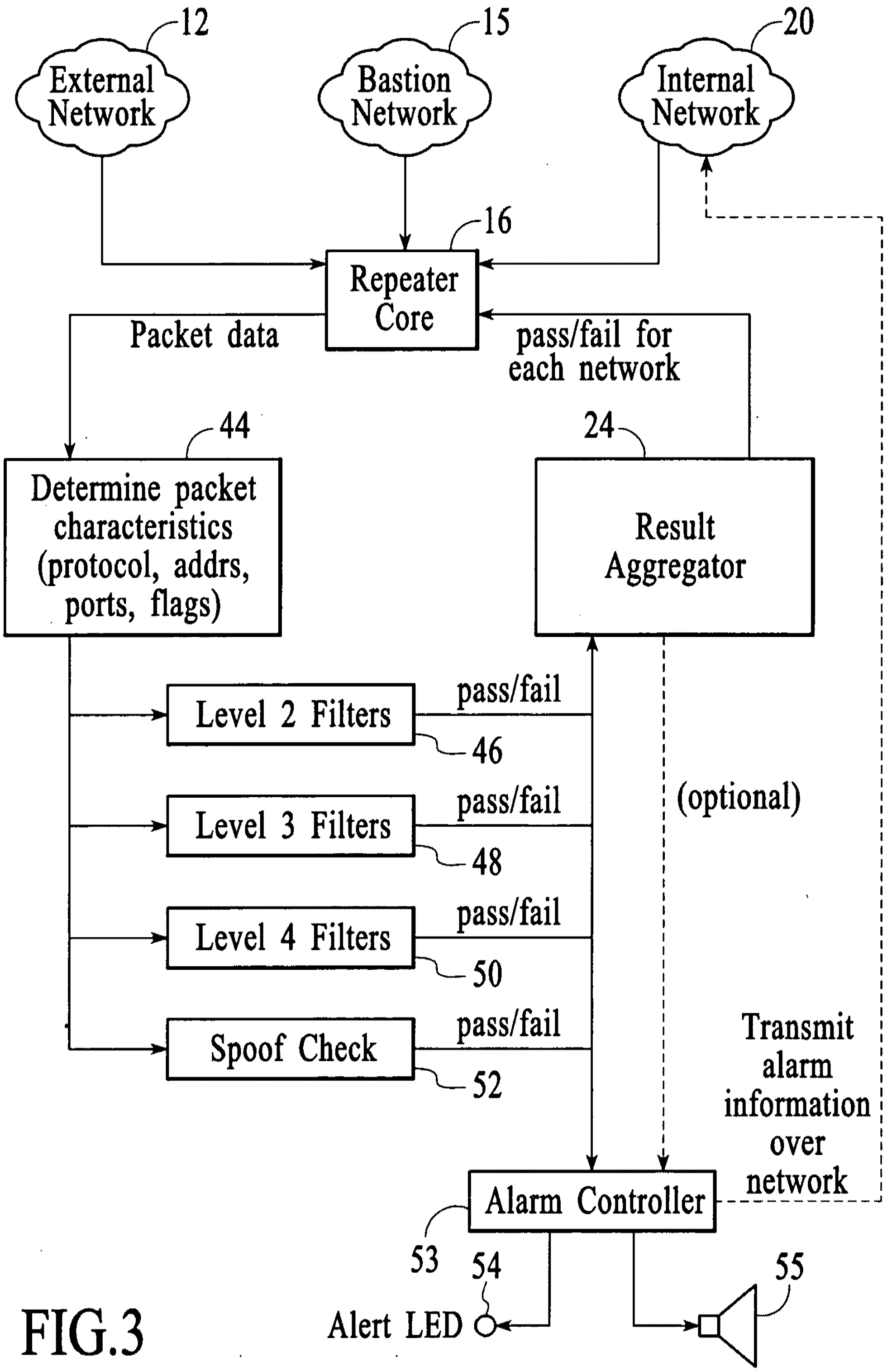
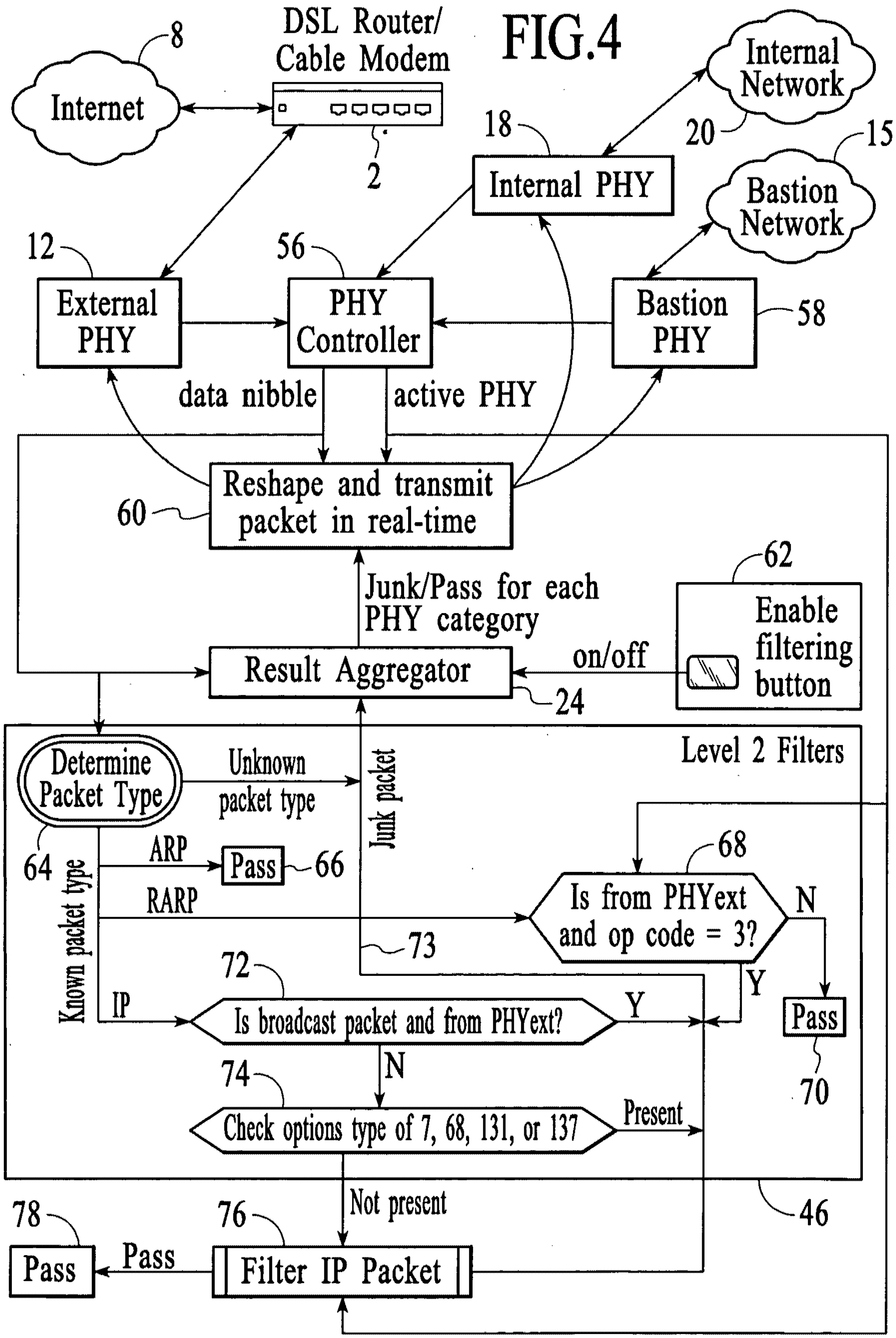

12
External Network
15
Bastion Network
20
Internal Network
16
Repeater Core
Packet data
pass/fail for each network
44
Determine packet characteristics (protocol, addrs, ports, flags)
24
Result Aggregator
Level 2 Filters
pass/fail
46
Level 3 Filters
pass/fail
48
Level 4 Filters
pass/fail
50
Spoof Check
pass/fail
52
(optional)
Transmit alarm information over network
Alarm Controller
53
54
Alert LED
55

FIG.3

FIG.4



8
Internet
DSL Router/
Cable Modem
2
Internal Network
20
18
Internal PHY
Bastion Network
15
12
External PHY
56
PHY Controller
Bastion PHY
58
data nibble
active PHY
Reshape and transmit packet in real-time
60
Junk/Pass for each PHY category
62
Enable filtering button
on/off
Result Aggregator
24
Level 2 Filters
Determine Packet Type
64
Unknown packet type
Junk packet
Known packet type
ARP
Pass
66
RARP
68
Is from PHYext and op code = 3?
N
Y
Pass
70
73
72
IP
Is broadcast packet and from PHYext?
Y
N
74
Check options type of 7, 68, 131, or 137
Present
Not present
46
78
Pass
Pass
76
Filter IP Packet

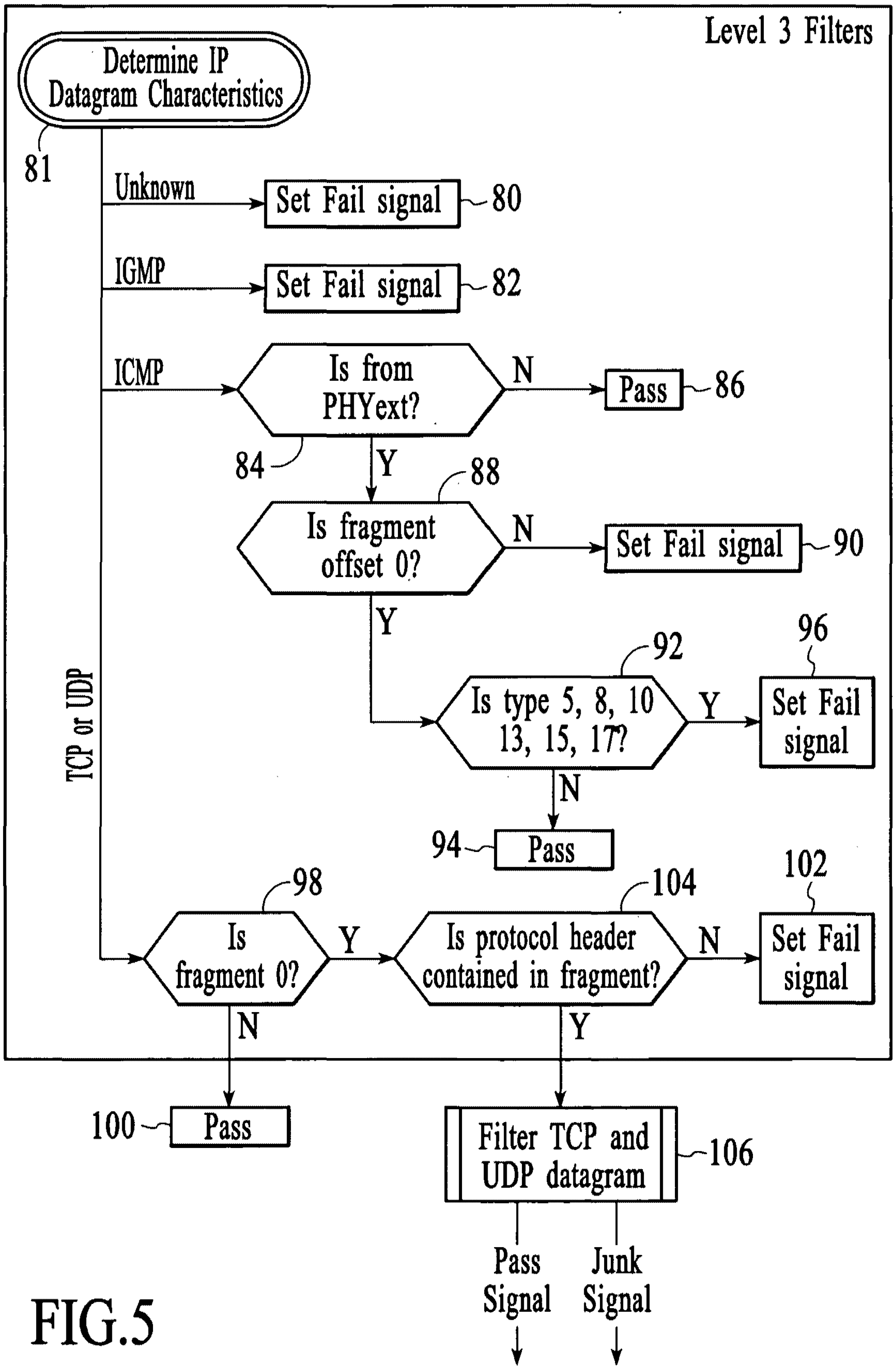
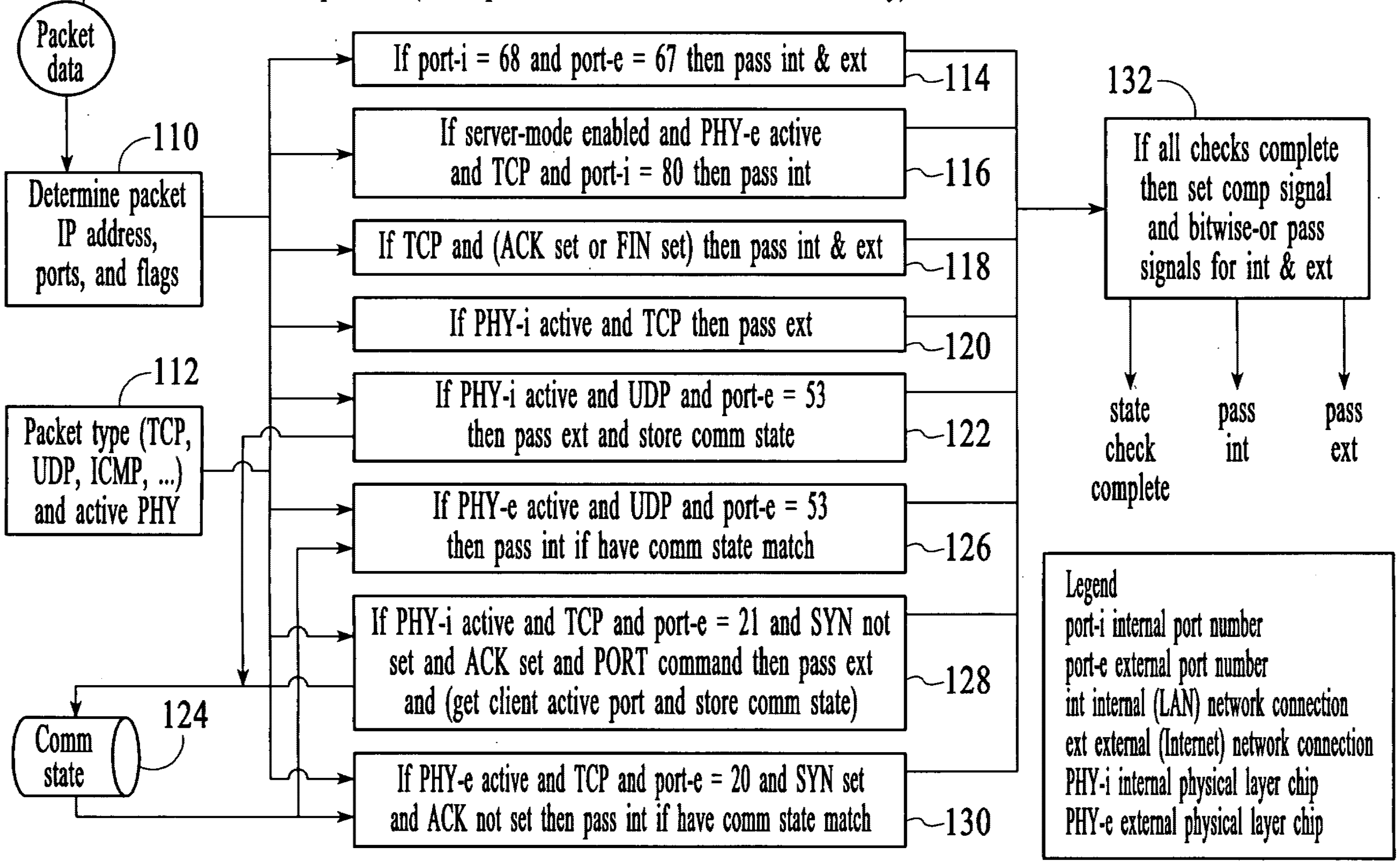

Level 3 Filters
Determine IP Datagram Characteristics
81
Unknown
Set Fail signal
80
IGMP
Set Fail signal
82
ICMP
Is from PHYext?
N
Pass
86
84
Y
88
Is fragment offset 0?
N
Set Fail signal
90
Y
92
96
Is type 5, 8, 10 13, 15, 17?
Y
Set Fail signal
N
94
Pass
TCP or UDP
98
104
102
Is fragment 0?
Y
Is protocol header contained in fragment?
N
Set Fail signal
N
Y
100
Pass
Filter TCP and UDP datagram
106
Pass Signal
Junk Signal

FIG.5

FIG.6

TCP and UDP packets are evaluated for pass or fail in parallel (other protocols also handled simultaneously)


108
Packet data
110
Determine packet IP address, ports, and flags
112
Packet type (TCP, UDP, ICMP, ...) and active PHY
124
Comm state
If port-i = 68 and port-e = 67 then pass int & ext
114
If server-mode enabled and PHY-e active and TCP and port-i = 80 then pass int
116
If TCP and (ACK set or FIN set) then pass int & ext
118
If PHY-i active and TCP then pass ext
120
If PHY-i active and UDP and port-e = 53 then pass ext and store comm state
122
If PHY-e active and UDP and port-e = 53 then pass int if have comm state match
126
If PHY-i active and TCP and port-e = 21 and SYN not set and ACK set and PORT command then pass ext and (get client active port and store comm state)
128
If PHY-e active and TCP and port-e = 20 and SYN set and ACK not set then pass int if have comm state match
130
132
If all checks complete then set comp signal and bitwise-or pass signals for int & ext
state check complete
pass int
pass ext
Legend
port-i internal port number
port-e external port number
int internal (LAN) network connection
ext external (Internet) network connection
PHY-i internal physical layer chip
PHY-e external physical layer chip

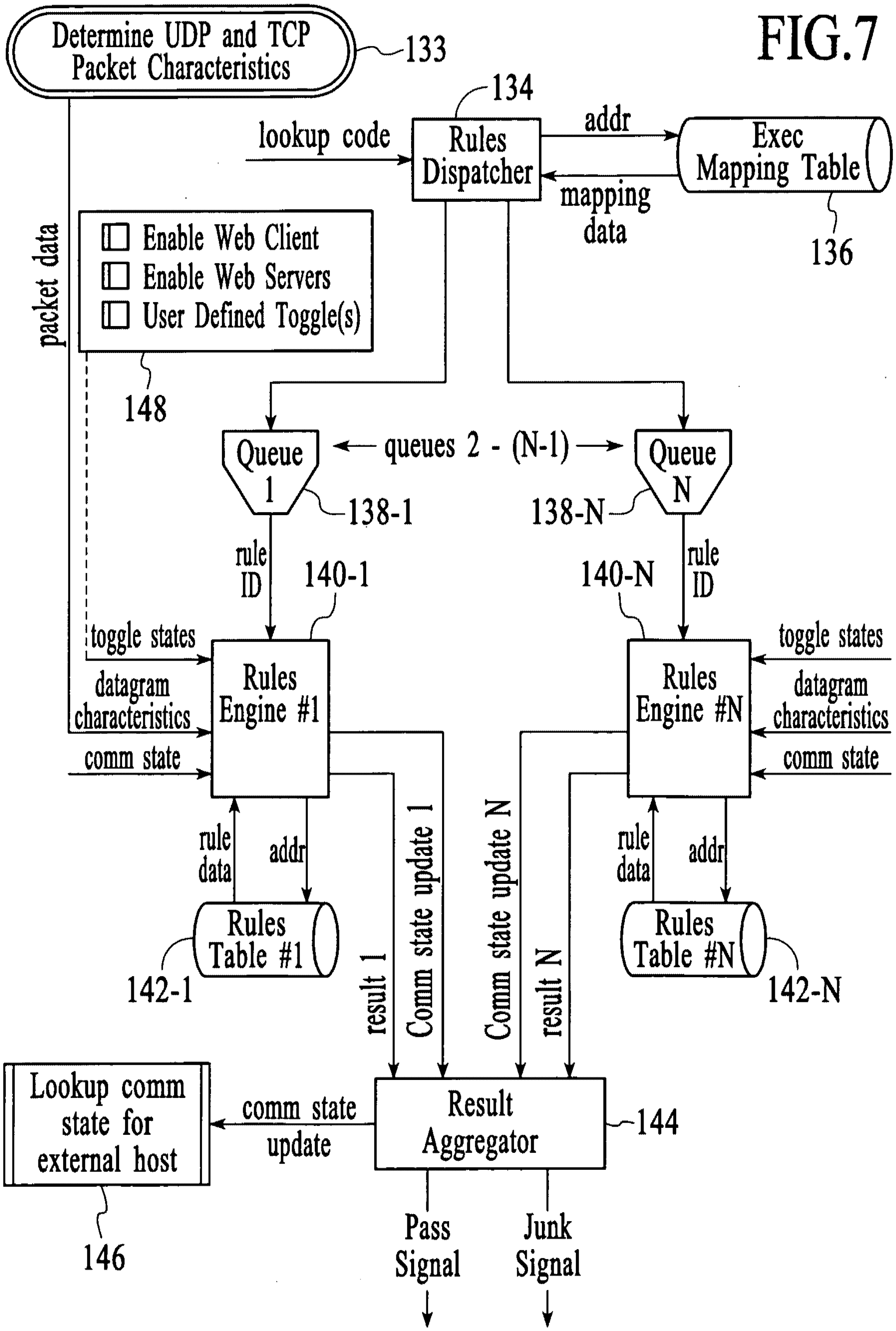

Determine UDP and TCP Packet Characteristics
133
134
lookup code
Rules Dispatcher
addr
Exec Mapping Table
mapping data
136
packet data
□ Enable Web Client
□ Enable Web Servers
□ User Defined Toggle(s)
148
Queue 1
queues 2 - (N-1)
Queue N
138-1
138-N
rule ID
140-1
140-N
rule ID
toggle states
datagram characteristics
comm state
Rules Engine #1
Rules Engine #N
toggle states
datagram characteristics
comm state
rule data
addr
Rules Table #1
142-1
result 1
Comm state update 1
Comm state update N
result N
rule data
addr
Rules Table #N
142-N
Lookup comm state for external host
146
comm state update
Result Aggregator
144
Pass Signal
Junk Signal

FIG.7

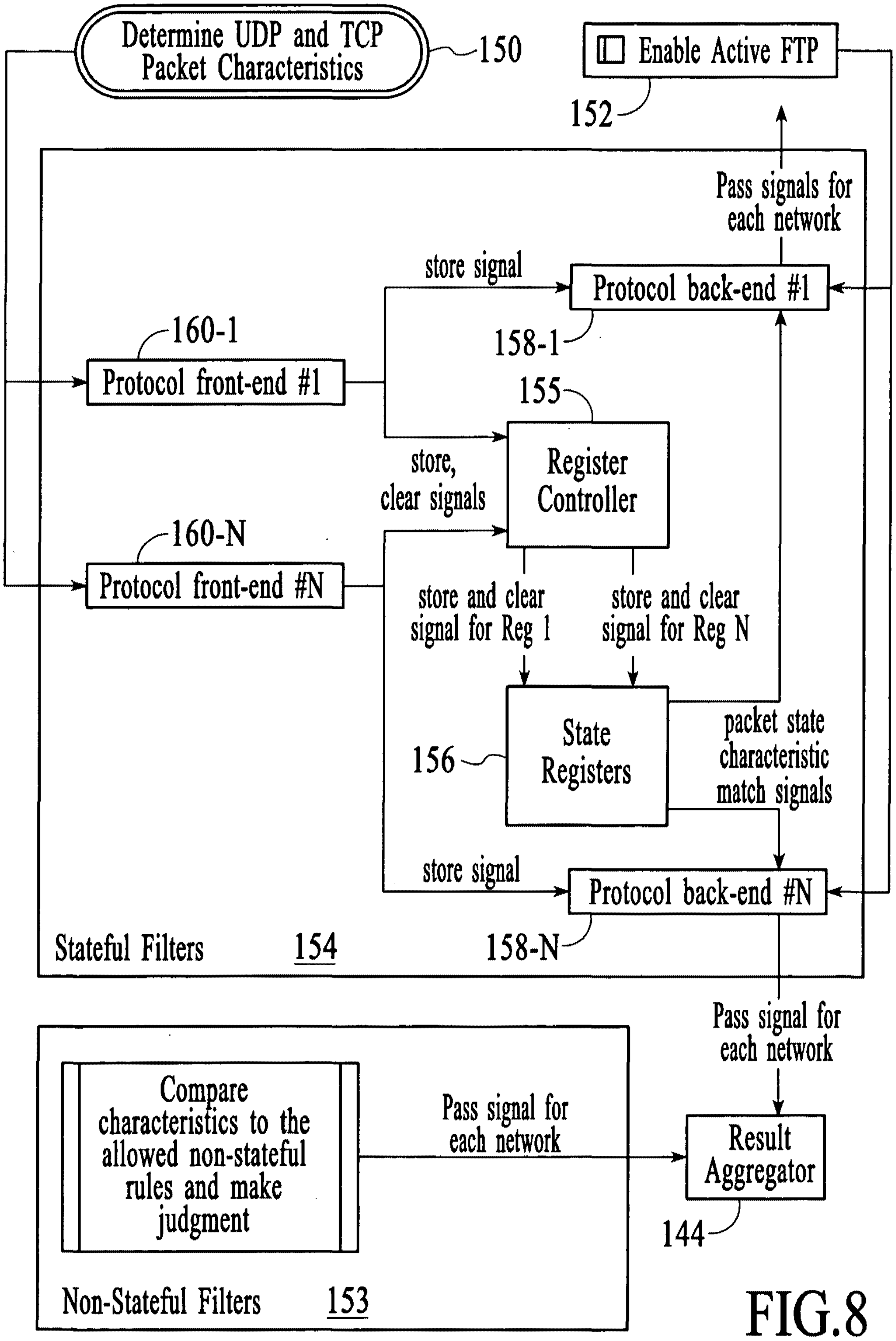

Determine UDP and TCP Packet Characteristics
150
□ Enable Active FTP
152
Pass signals for each network
store signal
Protocol back-end #1
158-1
160-1
Protocol front-end #1
155
Register Controller
store, clear signals
160-N
Protocol front-end #N
store and clear signal for Reg 1
store and clear signal for Reg N
State Registers
156
packet state characteristic match signals
store signal
Protocol back-end #N
158-N
Stateful Filters 154
Pass signal for each network
Compare characteristics to the allowed non-stateful rules and make judgment
Pass signal for each network
Result Aggregator
144
Non-Stateful Filters 153

FIG.8

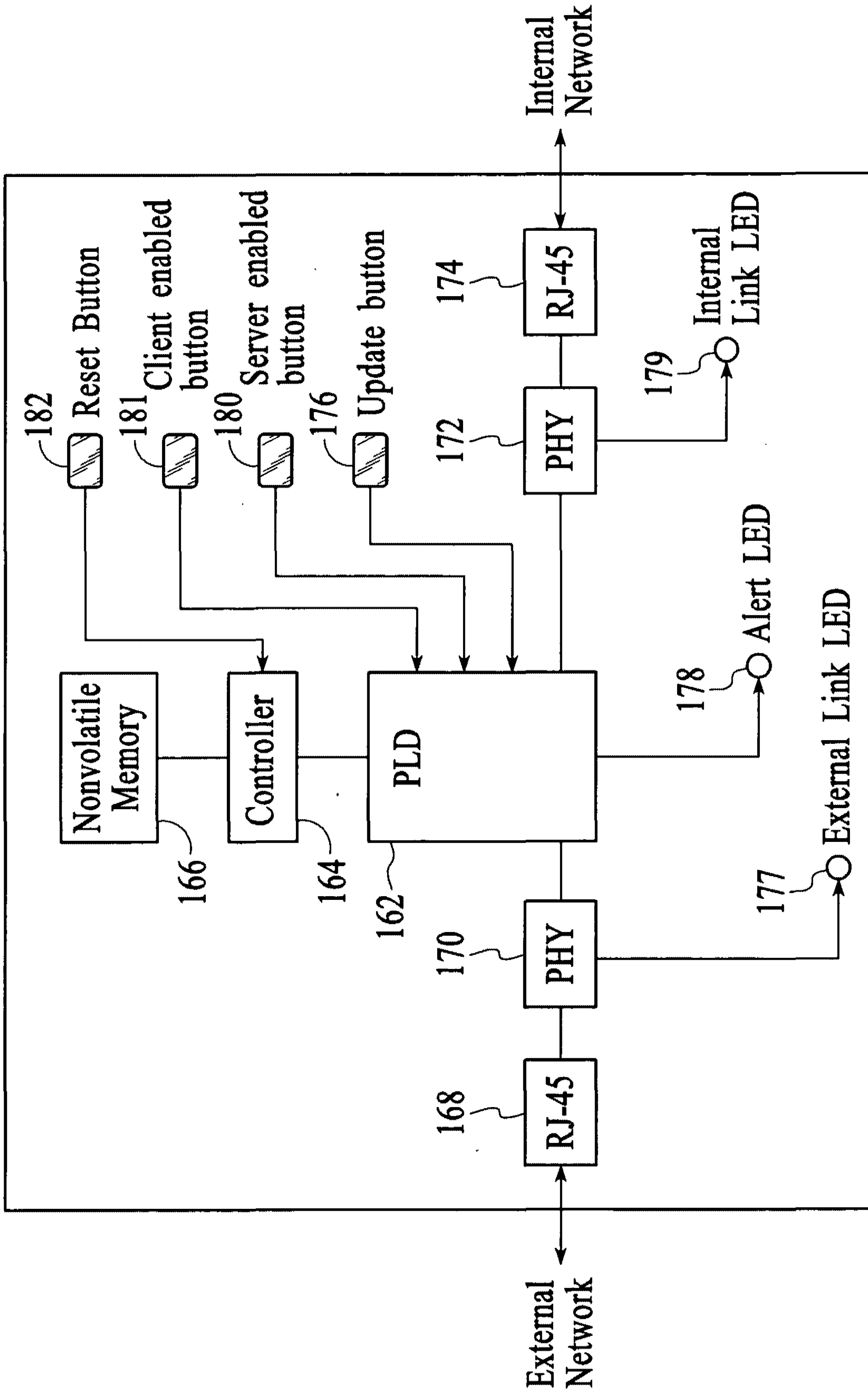

Internal Network
Reset Button
182
Client enabled button
181
Server enabled button
180
Update button
176
174
RJ-45
172
PHY
179
Internal Link LED
Nonvolatile Memory
Controller
PLD
Alert LED
178
External Link LED
177
166
164
162
170
PHY
168
RJ-45
External Network

FIG.9

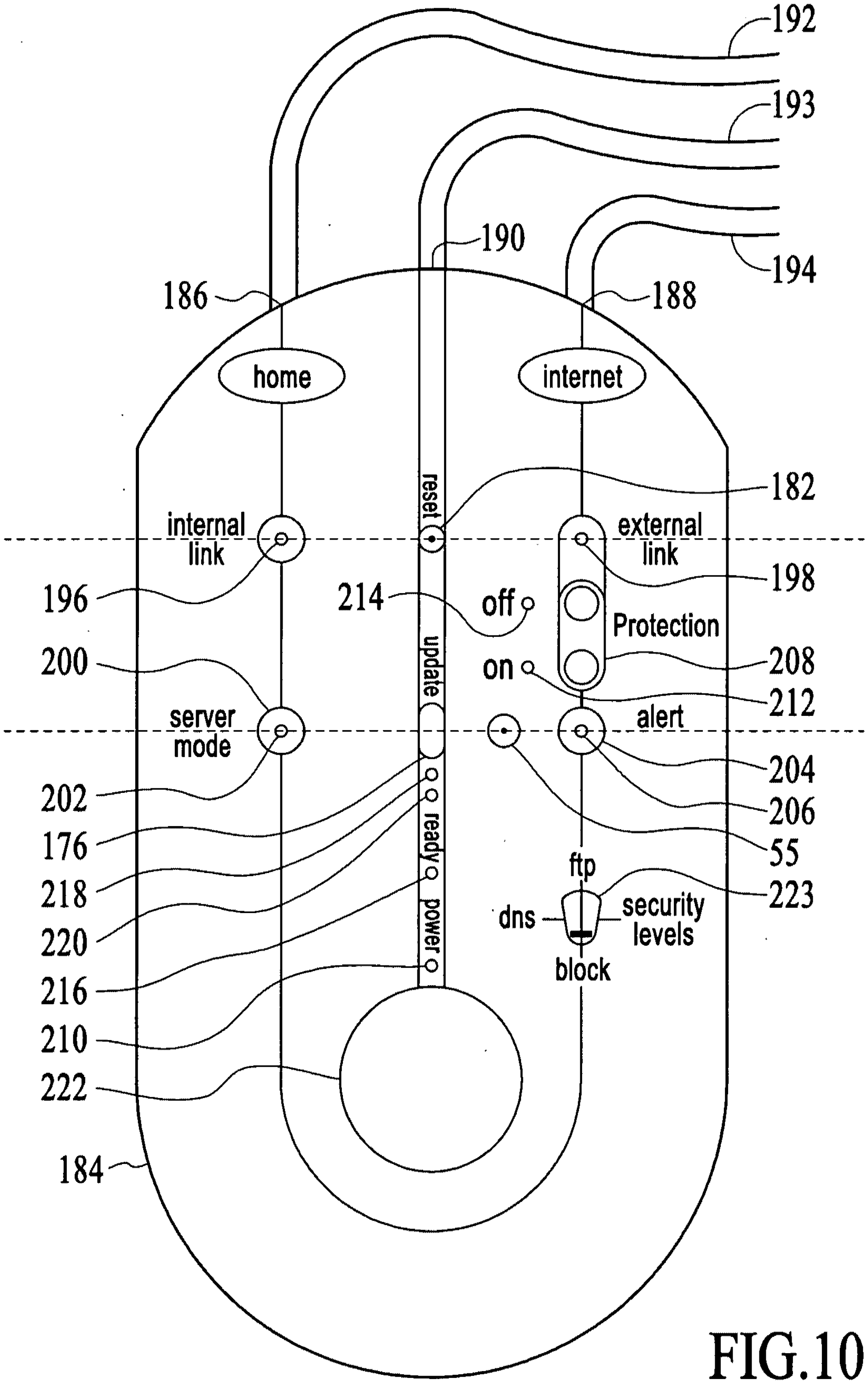

192
193
194
190
186
188
home
internet
182
reset
internal
link
external
link
196
214
off
Protection
198
208
200
on
update
212
server
mode
alert
204
202
206
176
ready
55
218
ftp
223
220
dns
security
levels
216
power
block
210
222
184

FIG.10

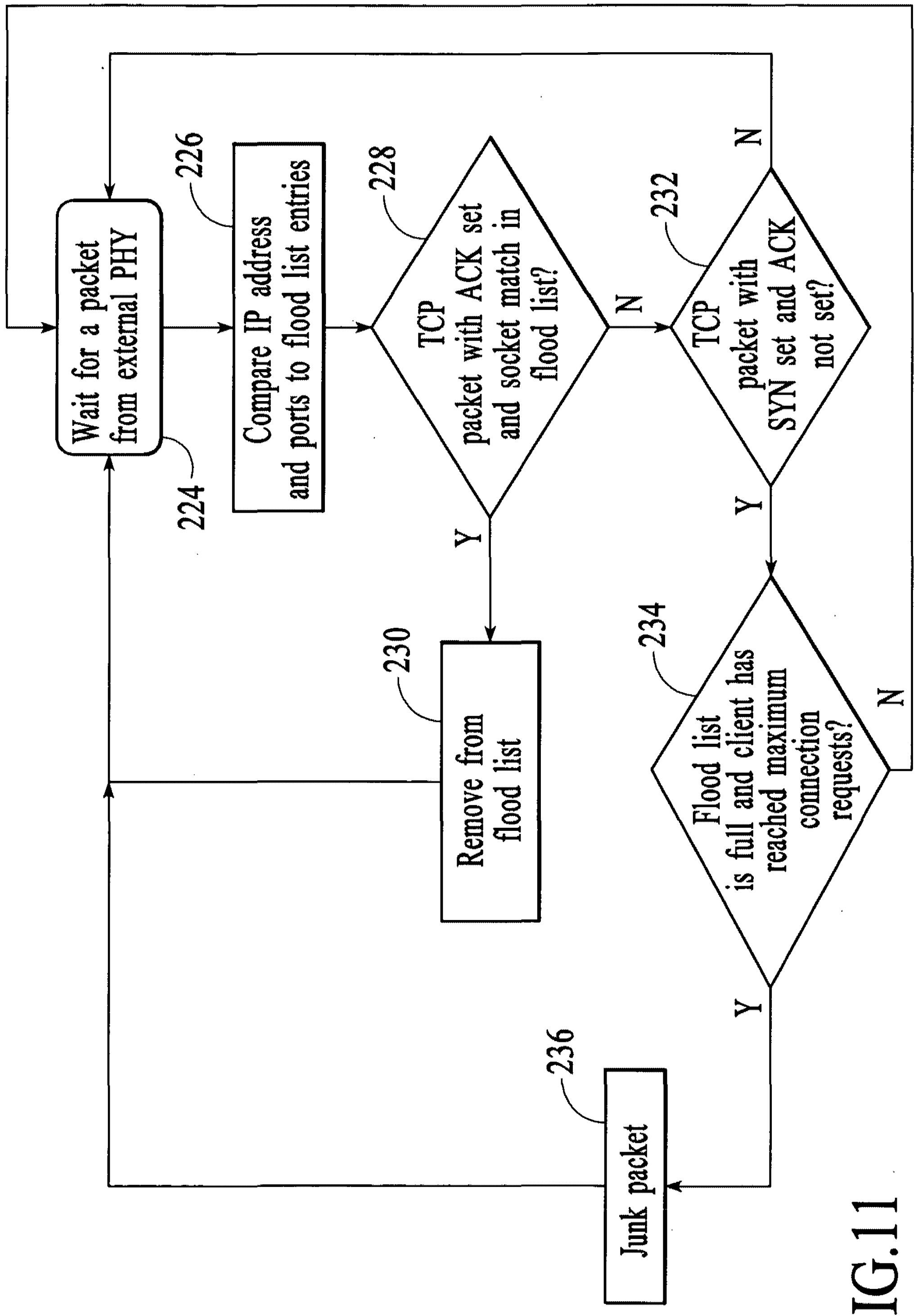
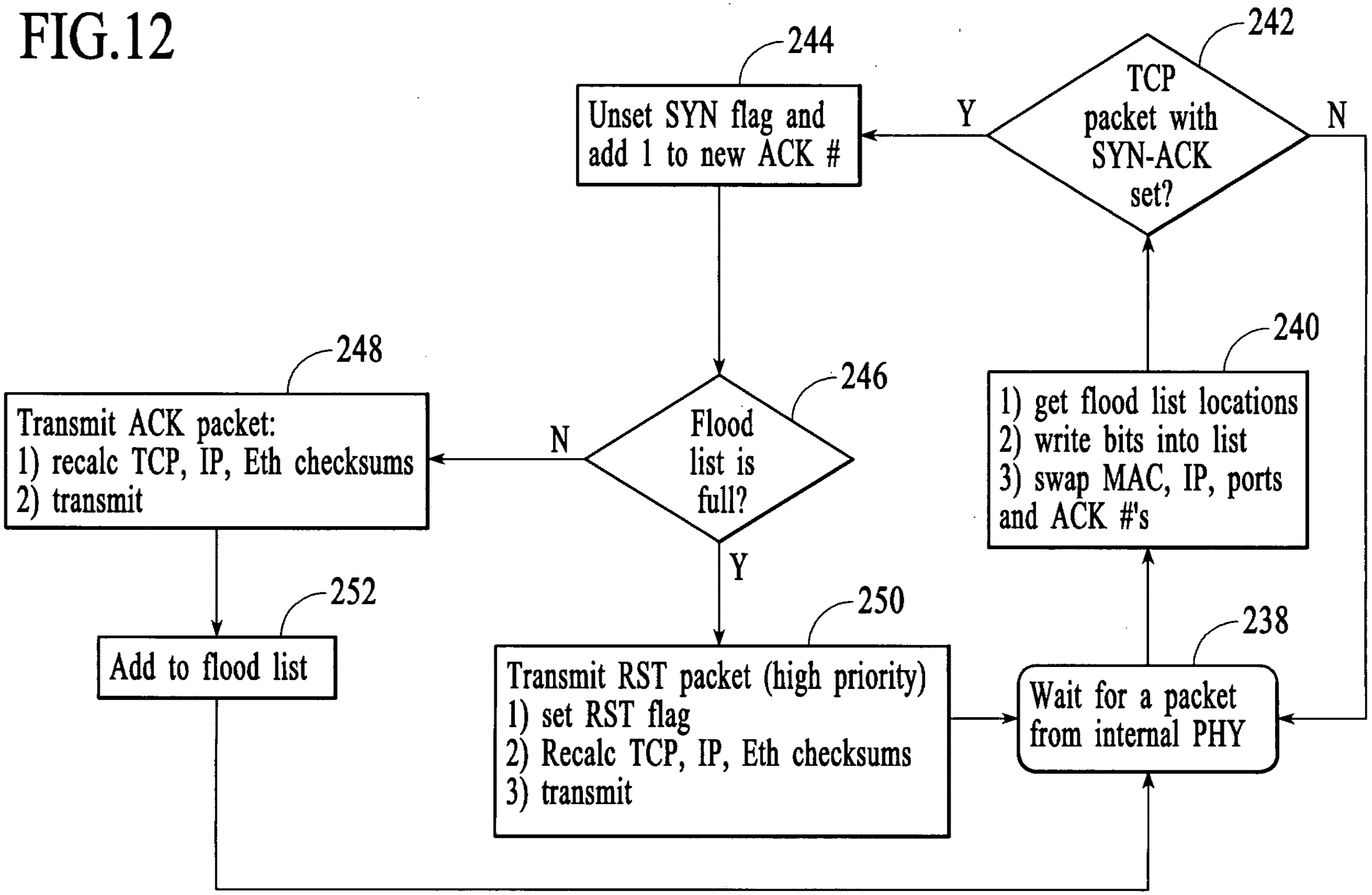

Wait for a packet from external PHY
224
Compare IP address and ports to flood list entries
226
TCP packet with ACK set and socket match in flood list?
228
Y
N
Remove from flood list
230
TCP packet with SYN set and ACK not set?
232
Y
N
Flood list is full and client has reached maximum connection requests?
234
Y
N
Junk packet
236

FIG.11

FIG.12


242
TCP packet with SYN-ACK set?
Y
N
244
Unset SYN flag and add 1 to new ACK #
240
1) get flood list locations
2) write bits into list
3) swap MAC, IP, ports and ACK #'s
246
Flood list is full?
N
Y
248
Transmit ACK packet:
1) recalc TCP, IP, Eth checksums
2) transmit
252
Add to flood list
250
Transmit RST packet (high priority)
1) set RST flag
2) Recalc TCP, IP, Eth checksums
3) transmit
238
Wait for a packet from internal PHY

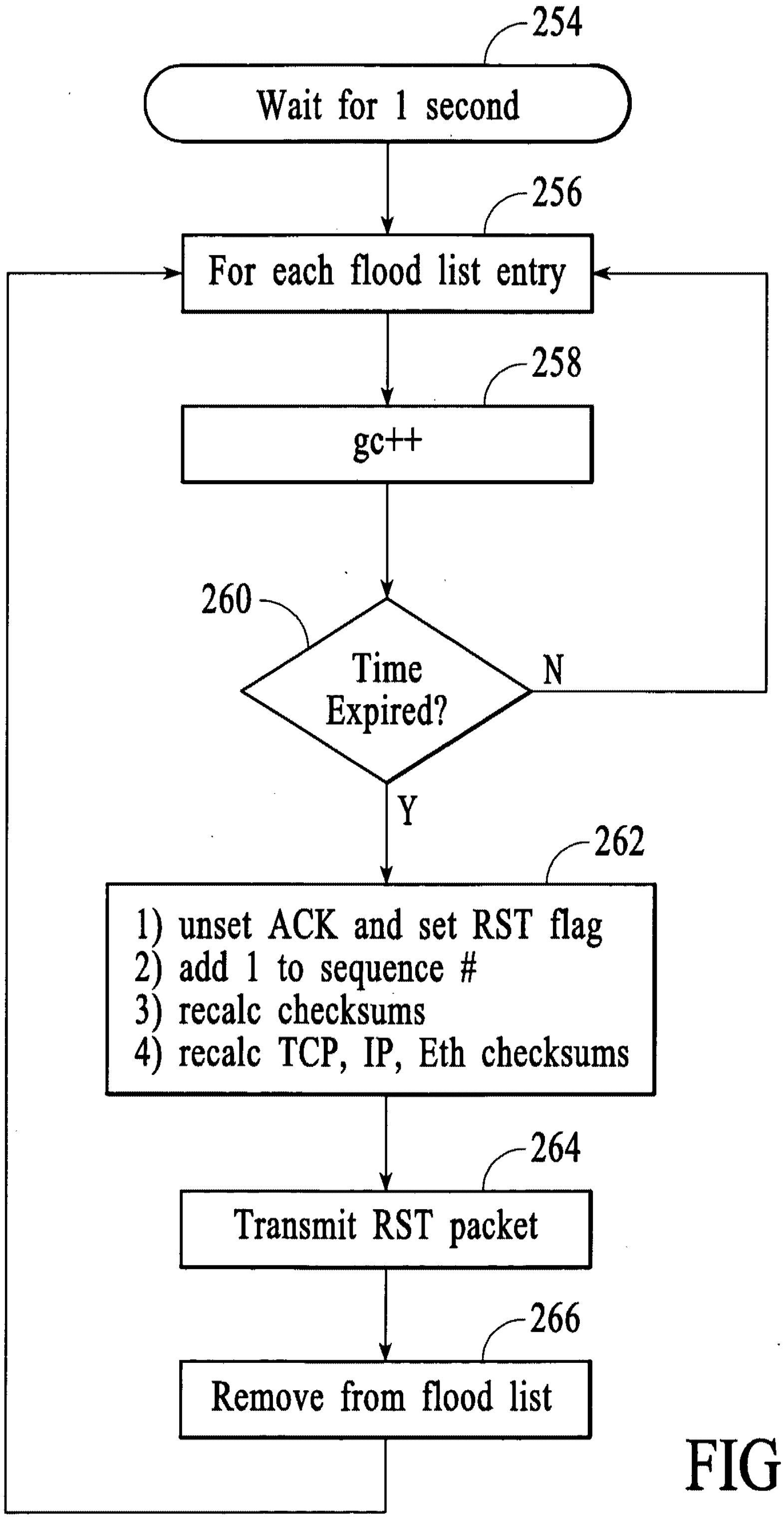

254
Wait for 1 second
256
For each flood list entry
258
gc++
260
Time Expired?
N
Y
262
1) unset ACK and set RST flag
2) add 1 to sequence #
3) recalc checksums
4) recalc TCP, IP, Eth checksums
264
Transmit RST packet
266
Remove from flood list

FIG.13

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	)	
	)	
Krumel	)	
	)	Art Unit: 2134
Serial No.: 09/611,775	)	
	)	
Filed: July 7, 2000	)	Examiner: Simitoski
	)	
For: Real Time Firewall/Data Protection Systems and Methods	)	
	)	
	)	

INFORMATION DISCLOSURE STATEMENT

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

1. Pursuant to 37 C.F.R. 1.97 and 1.98, and in compliance with 37 C.F.R. 1.56, the Office's attention is directed to the patents, publications and other information listed on the attached PTO-1449. A copy of each listed document is enclosed except for: (a) pending applications or (b) those previously cited or submitted to the Office in the following application(s) upon which this application relies for an earlier filing date under 35 U.S.C. 120:

Serial No.: ______________

Filing Date: ______________

Regarding the document(s), publication(s) or other information listed on the attached PTO-1449, Applicant(s) believe(s) the same may qualify as "prior" art to this application and should be treated accordingly, although Applicant(s) reserve(s) the right to contest the prior art status of any document, publication or information cited herein.

2. Regarding each listed document that is not in the English language, an English-language translation accompanies this Statement as indicated on the attached PTO-1449 or a concise explanation of the relevance of the document is set forth in the following documents(s):

(a) ___ Copy of each English language version of a search report indicating the degree of relevance found by the foreign office of each document being submitted from the search report.

(b) ___ Attachment entitled "Concise Explanation of Relevance of Non-English Language Documents."

3. Pursuant to 37 C.F.R. 1.97(b) this Statement is being filed (one must be checked):

(a) ___ Within 3 months of the filing date or date of entry into the National Stage.

(b) ___ Before the mailing date of a first Office Action on the merits. If this Statement is not filed before the mailing date of a first Office Action on the merits, the required certification is given below or, in the absence thereof, the Office is authorized to charge the required fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251 for consideration of this Statement.

(c) ___ After the period set forth in 37 C.F.R. 1.97(b) but before the mailing date of either a final action or a notice of allowance.

(1) ___ The required certification is given below, <u>or</u>

(2) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(p) for consideration of this Statement, or

(3) ___ Charge the fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251

(d) X After the mailing date of either a final action or a notice of allowance, but before payment of the issue fee. Petition hereby is made for consideration of this Statement and the required certification is indicated below.

(1) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(i)(1), or

(2) X Charge the fee set forth in 37 C.F.R. 1.17(i)(1) to Deposit Account No. 50-0251.

4. Certification (if applicable)

(a) ___ The undersigned hereby certifies that each item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than 3 months prior to the filing of this Statement.

(b) X The undersigned hereby certifies that no item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the undersigned's knowledge after making reasonable inquiry, was known to any individual designated in 37 C.F.R. 1.56(c) more than 3 months prior to the filing of this Statement.

5. The Commissioner is hereby authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

December 2, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
(408) 868-1516

I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated above.

Sheet 1 of 1

Form PTO-1449
(REV. 7-92)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use several sheets if necessary)

Attorney's Docket Number: 802-001

Serial No.: 09/611,775

Applicant(s): Krumel

Filing Date: 7/7/00

Group Art Unit: 2134

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	6	7	0	0	8	9	1	03/02/04	Wong	370	401	

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

● PRINTER RUSH ●
(PTO ASSISTANCE)

Application :	09611775	**Examiner :**	Simitoski	**GAU :**	2134
From:	J Black	**Location:**	[✓ IDC] FMF FDC	**Date:**	11/15/05

Tracking #: epm09611775 Week Date: 10/3/05

DOC CODE	DOC DATE	MISCELLANEOUS
□ 1449		□ Continuing Data
□ IDS		□ Foreign Priority
☒ CLM	6/29/05	□ Document Legibility
□ IIFW		□ Fees
□ SRFW		□ Other
□ DRW		
□ OATH		
□ 312		
□ SPEC		

[RUSH] **MESSAGE:**

Claim 16 depends on itself.

Please resolve.

[XRUSH] **RESPONSE:**

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

7590 09/27/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2134	

DATE MAILED: 09/27/2005

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

TITLE OF INVENTION: REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$0	$700	12/27/2005

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax (571) 273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 09/27/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

TITLE OF INVENTION: REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$0	$700	12/27/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
SIMITOSKI, MICHAEL J	2134	726-013000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ______________________

2 ______________________

3 ______________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual □ Corporation or other private group entity □ Government

4a. The following fee(s) are enclosed:

□ Issue Fee

□ Publication Fee (No small entity discount permitted)

□ Advance Order - # of Copies ______________

4b. Payment of Fee(s):

□ A check in the amount of the fee(s) is enclosed.

□ Payment by credit card. Form PTO-2038 is attached.

□ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number ______________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

□ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ______________________ Date ______________________

Typed or printed name ______________________ Registration No. ______________________

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 09/27/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2134	

DATE MAILED: 09/27/2005

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 619 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 619 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571) 272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

Notice of Allowability	**Application No.** 09/611,775	**Applicant(s)** KRUMEL, ANDREW K.
	Examiner Michael J. Simitoski	**Art Unit** 2134

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to RCE of 7/28/2005.

2. ☒ The allowed claim(s) is/are 1-66.

3. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a) □ All b) □ Some* c) □ None of the:

   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. □ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

   (a) □ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

   1) □ hereto or 2) □ to Paper No./Mail Date _____.

   (b) □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. □ Notice of References Cited (PTO-892)
2. □ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. □ Notice of Informal Patent Application (PTO-152)
6. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. □ Examiner's Statement of Reasons for Allowance
9. □ Other _____.

EXAMINER'S AMENDMENT

1. The IDS and response of 7/28/2005 was received and considered.

2. Claims 1-66 are allowed.

3. An examiner's informal amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

The application has been amended as follows:

In claim 44: Please replace "sever mode" (line 4 of the claim) to "server mode".

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Simitoski whose telephone number is (571) 272-3841. The examiner can normally be reached on Monday - Thursday, 6:45 a.m. - 4:15 p.m.. The examiner can also be reached on alternate Fridays from 6:45 a.m. – 3:15 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached at (571) 272-3838.

Any response to this action should be mailed to:
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Or faxed to:
(571) 273-8300
(for formal communications intended for entry)
Or:
(571) 273-3841 (Examiner's fax, for informal or draft communications, please label "PROPOSED" or "DRAFT")

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MJS
September 16, 2005

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Sheet 1 of 2

Form PTO-1449
(REV. 7-92)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use several sheets if necessary)

Attorney's Docket Number 802-001

Serial No. 09/611,775

Applicant(s): Krumel

Filing Date: 7/7/00

Group Art Unit: 2134

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5	3	4	3	4	7	1	08-1994	Cassagnol	370	401	
	5	4	2	6	3	7	8	6/20/95	Ong	326	39	
	5	4	2	6	3	7	9	06-1995	Trimberger	326	39	
	5	5	9	0	0	6	0	12-1996	Granville	702	155	
	5	7	4	5	2	2	9	04-1998	June	356	73	
	5	7	9	4	0	3	3	8/11/98	Aldebert et al.	395	653	
	5	9	0	3	5	6	6	05-1999	Flammer	370	406	
	5	9	7	4	5	4	7	10-1999	Klimenko	713	2	
	6	0	2	0	7	5	8	02-2000	Patel	326	40	
	6	0	7	6	1	6	8	06-2000	Fiveash	713	201	
	6	0	4	9	2	2	2	4/11/00	Lawmann	326	38	
	6	0	5	2	7	8	5	04-2000	Lin	709	225	
	6	0	7	8	7	3	6	6/20/00	Guccione	395	500.17	
	6	0	9	2	1	2	3	07-2000	Steffan	710	8	
	6	1	5	1	6	2	5	11-2000	Swales	709	218	
	6	1	7	5	8	3	9	01-2001	Takao	715	500	
	6	1	8	2	2	2	5	01-2001	Hagiuda	713	201	
	6	2	1	5	7	6	9	04-2001	Ghani	370	230	
	6	3	1	0	6	9	2	10-2001	Fan	358	1.14	
	6	3	2	6	8	0	6	12-2001	Fallside	326	38	
	6	3	3	3	7	9	0	12-2001	Kageyama	358	1.15	
	6	3	4	3	3	2	0	01-2002	Fairchild	709	224	
	6	3	6	3	5	1	9	02-2002	Levi	716	16	
	6	3	7	4	3	1	8	04-2002	Hayes	710	107	
	6	3	8	9	5	4	4	05-2002	Katagiri	713	300	

MS	6	4	1	4	4	7	6	07-2002	Yagi	324	127	
	6	4	3	0	7	1	1	08-2002	Sekizawa	714	47	
	6	5	4	9	9	4	7	04-2003	Suzuki	709	229	
	6	6	2	8	6	5	3	09-2003	Salim	370	389	
	6	6	4	0	3	3	4	10-2003	Rasmussen	717	171	
	6	7	3	4	9	8	5	05-2004	Ochiai	358	1.15	
	6	7	7	1	6	4	6	08-2004	Sarkissian	370	392	
↓	6	7	7	9	0	0	4	08-2004	Zintel	709	227	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

MS		"Jini Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/jini1_1.pdf, pp. 1-20.
		"Jini Device Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/devicearch1_1.pdf, pp. 1-14.
↓		Sollins, K., "The TFTP Protocol (Revision 2.0)", MIT, July 1992. Available from Internet: http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1350.html, pp. 1-10.

EXAMINER

DATE CONSIDERED 9/16/05

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Sheet 1 of 1

Form PTO-1449 (REV. 7-92) U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office	Attorney's Docket Number 802-001	Serial No. 09/611,775	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary)	Applicant(s): Krumel		
	Filing Date: July 7, 2000	Group Art Unit	

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5	7	4	0	3	7	5	4/14/98	Dunne et al.	395	200.68	
	5	8	3	5	7	2	6	11/10/98	Shwed et al.	395	200.59	
	5	8	8	4	0	2	5	3/16/99	Baehr et al.	395	187.01	
	5	9	6	8	1	7	6	10/19/99	Nessett et al.	713	201	
	6	0	0	3	1	3	3	12/14/99	Moughanni et al.	713	200	
	6	0	0	9	4	7	5	12/28/99	Shrader	709	249	

FOREIGN PATENT DOCUMENTS

	DOCUMENT NUMBER							DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
												YES	NO
	WO	96/	3	4	4	7	9	10/31/96	PCT				
	WO	99/	4	8	3	0	3	9/23/99	PCT				
	WO	00/	0	2	1	1	4	1/13/00	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Issue Classification	**Application/Control No.** 09/611,775	**Applicant(s)/Patent under Reexamination** KRUMEL, ANDREW K.
	Examiner Michael J. Simitoski	**Art Unit** 2134

ISSUE CLASSIFICATION

ORIGINAL					CROSS REFERENCE(S)							
CLASS				SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
726				13	713	154						
INTERNATIONAL CLASSIFICATION					709	229						
H	0	4	L	9/00								
G	0	6	F	15/16								
				/								
				/								
				/								

Michael J. Simitoski 9/16/2005 (Assistant Examiner) (Date)	GREGORY MORSE SUPERVISORY PATENT EXAMINER TECHNOLOGY CENTER 2100 (Primary Examiner) (Date)	Total Claims Allowed: 66	
Brenda Harrison (Legal Instruments Examiner) (Date)		O.G. Print Claim(s) 31	O.G. Print Fig. Fig. 8

☒ Claims renumbered in the same order as presented by applicant □ CPA □ T.D. □ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		31		61		91		121		151		181
	2		32		62		92		122		152		182
	3		33		63		93		123		153		183
	4		34		64		94		124		154		184
	5		35		65		95		125		155		185
	6		36		66		96		126		156		186
	7		37		67		97		127		157		187
	8		38		68		98		128		158		188
	9		39		69		99		129		159		189
	10		40		70		100		130		160		190
	11		41		71		101		131		161		191
	12		42		72		102		132		162		192
	13		43		73		103		133		163		193
	14		44		74		104		134		164		194
	15		45		75		105		135		165		195
	16		46		76		106		136		166		196
	17		47		77		107		137		167		197
	18		48		78		108		138		168		198
	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

Search Notes	**Application/Control No.** 09/611,775	**Applicant(s)/Patent under Reexamination** KRUMEL, ANDREW K.
	Examiner Michael J. Simitoski	**Art Unit** 2134

SEARCHED			
Class	Subclass	Date	Examiner
726	13,11	9/16/2005	MJS
713	154	9/16/2005	MJS

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
726	**13,11**	9/16/2005	**MJS**
713	154	9/16/2005	MJS

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
709/229,249,225 370/356,389,392,401,395.21,395.32 See attached EAST search notes for details.	9/16/2005	MJS
Consulted Andrew Caldwell regarding allowance.	9/16/2005	MJS

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L10	4950	(709/229,249,225).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2005/09/16 06:38
L13	10376	(370/356,389,392,401,395.21,395.32). CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; IBM_TDB	OR	OFF	2005/09/16 06:39
L23	ABS 180	726/11.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 07:30
L24	ABS + IMG 185	726/13.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 07:31
L25	52	(selectiv$4 near2 invalidat$3)".clm"	US-PGPUB	OR	ON	2005/09/16 07:32
L26	KWIC 52	(selectiv$4 near2 invalidat$3)".clm" and "packet.clm"	US-PGPUB	OR	ON	2005/09/16 07:32
L28	8	(selectiv$4 near2 invalidat$3).clm.	US-PGPUB	OR	ON	2005/09/16 07:33
L29	395	(selectiv$4 near2 alter$5).clm.	US-PGPUB	OR	ON	2005/09/16 07:33
L30	10	(selectiv$4 near2 alter$5).clm. and packet.clm.	US-PGPUB	OR	ON	2005/09/16 07:35
L31	61	(invalid and packet and (portion cell)). clm.	US-PGPUB	OR	ON	2005/09/16 07:35
L32	KWIC 51	(invalid and packet and (portion cell)). clm. and network	US-PGPUB	OR	ON	2005/09/16 07:36
L33	15	(invalid and packet and (portion cell)). clm. and network and (filter$3 firewall)	US-PGPUB	OR	ON	2005/09/16 07:42
L34	4760	(time with ((end last) near2 (packet portion cell)) and (receiv$3 arriv$3))	US-PGPUB	OR	ON	2005/09/16 07:43
L35	723	(time with ((end last) near2 (packet portion cell)) with (receiv$3 arriv$3))	US-PGPUB	OR	ON	2005/09/16 07:43
L36	333	(time with ((end last) near2 (portion cell)) with (receiv$3 arriv$3))	US-PGPUB	OR	ON	2005/09/16 07:43
L40	Full 6	krumel.in.	US-PGPUB	OR	ON	2005/09/16 07:46
L41	261	110 and @ad<"20000707" and @pd>"20040428"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:02
L42	47	110 and @ad<"20000707" and @pd>"20040428" and firewall	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:03
L48	KWIC 540	113 and @ad<"20000707" and @pd>"20040428"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:04
L49	37	113 and @ad<"20000707" and @pd>"20040428" and firewall	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:04
L50	4	(148 141) and (alter near2 (packet cell portion))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:05
L51	ABS + KWIC 28	(148 and 141)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:05
L52	4	(148 141) and (alter near2 packet)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:05

L53	ABS + KWIC 8	(148 141) and (alter$3 near2 (packet cell portion))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:05
L54	KWIC 83	(148 141) and ((alter$5 invalid$5) with (packet cell portion))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:05
L55	944	@ad<"20000707" and ((invalidat$3) with (packet cell portion))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:06
L56	12	@ad<"20000707" and ((invalidat$3) with ((end portion) near2 (packet cell)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:06
L57	ABS + KWIC 13	@ad<"20000707" and (time with ((end last) near2 (portion cell)) with (receiv$3 arriv$3))	US-PGPUB	OR	ON	2005/09/16 08:06
L58	0	@ad<"20000707" and (time with ((end last) near2 (portion cell)) with (receiv$3 arriv$3)) and (firewall (packet adj filter$3))	US-PGPUB	OR	ON	2005/09/16 08:06
L59	0	@ad<"20000707" and (time with ((end last) near2 (portion packet cell)) with (receiv$3 arriv$3)) and (firewall (packet adj filter$3))	US-PGPUB	OR	ON	2005/09/16 08:06

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L24	Kwic 48	((((@ad<"20000707" and 370/356,389,392, 395.21,395.32,401.ccls.) and ("709"/$. ccls. and "713"/$.ccls.)) (@ad<"20000707" and 709/229,249,225. ccls.) (@ad<"20000707" and 370/356,389, 392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time) and rule) not ((("5740375") or ("5835726") or ("5884025") or ("5968176") or ("6003133") or ("6009475")).PN.)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:09
L23	41	((((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<"20000707") not bowman.in.) and (switch button)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L22	Kwic 88	(((@ad<"20000707" and 370/356,389,392, 395.21,395.32,401.ccls.) and ("709"/$. ccls. and "713"/$.ccls.)) (@ad<"20000707" and 709/229,249,225. ccls.) (@ad<"20000707" and 370/356,389, 392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L21	172	(((@ad<"20000707" and 370/356,389,392, 395.21,395.32,401.ccls.) and ("709"/$. ccls. and "713"/$.ccls.)) (@ad<"20000707" and 709/229,249,225. ccls.) (@ad<"20000707" and 370/356,389, 392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L20	Kwic 18	(@ad<"20000707" and 370/356,389,392,395. 21,395.32,401.ccls.) and ("709"/$.ccls. and "713"/$.ccls.)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L19	5783	@ad<"20000707" and 370/356,389,392,395. 21,395.32,401.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L18	2206	@ad<"20000707" and 709/229,249,225.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L17	0	@ad<"20000707" and 713/201.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L16	Kwic 88	@ad<"20000707" and firewall and ((determin$5 decid$3 decision) near3 (threshold (time adj limit$3) (too adj long)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L15	88	@ad<"20000707" and firewall and ((determin$5 decid$3 decision) near3 (threshold (time adj limit$3) (too adj long)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L14	ABS + Kwic 3	@ad<"20000707" and firewall and (((determin$5 decid$3 decision) near3 ((time adj limit$3) (too adj long))))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L13	12	@ad<"20000707" and firewall and ((rules) same ((determin$5 decid$3 decision) near3 (threshold (time adj limit$3) (too adj long))))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L12	184	@ad<"20000707" and firewall and ((determin$5 decid$3 decision) with (threshold (time adj limit$3) (too adj long)))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:08
L10	Kwic 56	(((configur$5 manag$3) near (router firewall hub)) same tcp) and @ad<"20000707"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:07

L8	12	@ad<"20000707" and firewall and ((rules) same ((determin$5 decid$3 decision) near3 (threshold (time adj limit$3) (too adj long))))	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:07

L7	ABS + KWIC 27	@ad<"20000707" and atm and (burst adj2 (size length)) same (packet adj2 (size length))	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:07

L6	4	@ad<"20000707" and ((real adj time) near (firewall (packet adj filter)))	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:07

L4	KWIC 48	(((@ad<"20000707" and 370/356,389,392,395.21,395.32,401.ccls.) and ("709"/$.ccls. and "713"/$.ccls.)) (@ad<"20000707" and 709/229,249,225.ccls.) (@ad<"20000707" and 370/356,389,392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time) and rule	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:07

L3	ABS + KWIC 18	(@ad<"20000707" and 370/356,389,392,395.21,395.32,401.ccls.) and ("709"/$.ccls. and "713"/$.ccls.)	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:07

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L26	Full 4	726/13.ccls. and ((invalid$3 valid) with packet).clm.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/09/16 08:17

<table>
<tr><th>Ref #</th><th>Hits</th><th>Search Query</th><th>DBs</th><th>Default Operator</th><th>Plurals</th><th>Time Stamp</th></tr>
<tr><td>L25</td><td>33
Full (applicants IDS)</td><td>"5343471"|"5426378"|"5426379"|"5590060"|"5745229"|"5794033"|"5903566"|"5974547"|"6020458"|"6049222"|"6052785"|"6076168"|"6078736"|"6092123"|"6151625"|"6175839"|"6182225"|"6215769"|"6310692"|"6326806"|"6333790"|"6343320"|"6363519"|"6374318"|"6389544"|"6414476"|"6430711"|"6549947"|"6628653"|"6640334"|"6734985"|"6771646"|"6779004").PN.</td><td>US-PGPUB; USPAT; EPO; JPO; IBM_TDB</td><td>OR</td><td>ON</td><td>2005/09/16 08:14</td></tr>
</table>

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp

L27	ABI
	+
	JMG
	117
		713/154.ccls.	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/09/16 08:24

PTO/SB/30 (04-05)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Request for Continued Examination (RCE) Transmittal

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	09/611,775
Filing Date	07/07/2000
First Named Inventor	KRUMEL
Art Unit	2134
Examiner Name	SIMITOSKI, MICHAEL J.
Attorney Docket Number	802-001

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. Submission required under 37 CFR 1.114 Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

   a. ☒ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

      i. □ Consider the arguments in the Appeal Brief or Reply Brief previously filed on ____________

      ii. ☒ Other AMENDMENT & IDS

   b. □ Enclosed

      i. □ Amendment/Reply
      ii. □ Affidavit(s)/ Declaration(s)
      iii. □ Information Disclosure Statement (IDS)
      iv. □ Other ____________

2. Miscellaneous

   a. □ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of ______ months. (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

   b. □ Other ____________

3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.

   a. ☒ The Director is hereby authorized to charge the following fees, any underpayment of fees, or credit any overpayments, to Deposit Account No. 50-0251 [✓ I have enclosed a duplicate copy of this sheet.]

      i. ☒ RCE fee required under 37 CFR 1.17(e)
      ii. □ Extension of time fee (37 CFR 1.136 and 1.17)
      iii. □ Other ____________

   b. □ Check in the amount of $ ____________ enclosed

   c. □ Payment by credit card (Form PTO-2038 enclosed)

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Signature		Date	7/28/05
Name (Print/Type)	ALAN R. LOUDERMILK	Registration No.	32,788

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the U.S. Patent and Trademark Office on the date shown below.

Signature			
Name (Print/Type)	ALAN R. LOUDERMILK	Date	7/28/05

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IFW

AF

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Enter with RCE filed 7-28-05
B.#. 8-12-05

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL REJECTION

Do not enter. 7/12/05

Sir or Madam:

In response to the office action mailed April 28, 2005, please re-examine the above-identified application in view of the following amendment and remarks.

UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 07/15/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2134	

DATE MAILED: 07/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Advisory Action Before the Filing of an Appeal Brief

Application No.	*09/611,775*	Applicant(s)	KRUMEL, ANDREW K.
Examiner	Michael J. Simitoski	Art Unit	2134

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 29 June 2005 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

a) □ The period for reply expires ______months from the mailing date of the final rejection.

b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. □ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☒ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because

(a) ☒ They raise new issues that would require further consideration and/or search (see NOTE below);

(b) □ They raise the issue of new matter (see NOTE below);

(c) ☒ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

(d) □ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: See Continuation Sheet. (See 37 CFR 1.116 and 41.33(a)).

4. □ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).

5. □ Applicant's reply has overcome the following rejection(s): _____.

6. □ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

7. □ For purposes of appeal, the proposed amendment(s): a) □ will not be entered, or b) □ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.

The status of the claim(s) is (or will be) as follows:
Claim(s) allowed: _____.
Claim(s) objected to: _____.
Claim(s) rejected: _____.
Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. □ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).

9. □ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).

10. □ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. □ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.

12. □ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s). _____

13. ☒ Other: See Continuation Sheet.

Continuation of 3. NOTE: The claims dependent on claims 1 and 31 require further consideration in light of the newly amended independent claims.

Continuation of 13. Other: The IDS of 6/29/2005 is was not considered because it does not meet the certification requirements as set forth in 37 CFR §1.97.

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

not considered

Sheet 1 of 2

Form PTO-1449 (REV. 7-92) U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** (Use several sheets if necessary)	Attorney's Docket Number 802-001	Serial No. 09/611,775
	Applicant(s): Krumel	
	Filing Date: 7/7/00	Group Art Unit: 2134

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5	3	4	3	4	7	1	08-1994	Cassagnol	370	401	
	5	4	2	6	3	7	8	6/20/95	Ong	326	39	
	5	4	2	6	3	7	9	06-1995	Trimberger	326	39	
	5	5	9	0	0	6	0	12-1996	Granville	702	155	
	5	7	4	5	2	2	9	04-1998	June	356	73	
	5	7	9	4	0	3	3	8/11/98	Aldebert et al.	395	653	
	5	9	0	3	5	6	6	05-1999	Flammer	370	406	
	5	9	7	4	5	4	7	10-1999	Klimenko	713	2	
	6	0	2	0	7	5	8	02-2000	Patel	326	40	
	6	0	7	6	1	6	8	06-2000	Fiveash	713	201	
	6	0	4	9	2	2	2	4/11/00	Lawmann	326	38	
	6	0	5	2	7	8	5	04-2000	Lin	709	225	
	6	0	7	8	7	3	6	6/20/00	Guccione	395	500.17	
	6	0	9	2	1	2	3	07-2000	Steffan	710	8	
	6	1	5	1	6	2	5	11-2000	Swales	709	218	
	6	1	7	5	8	3	9	01-2001	Takao	715	500	
	6	1	8	2	2	2	5	01-2001	Hagiuda	713	201	
	6	2	1	5	7	6	9	04-2001	Ghani	370	230	
	6	3	1	0	6	9	2	10-2001	Fan	358	1.14	
	6	3	2	6	8	0	6	12-2001	Fallside	326	38	
	6	3	3	3	7	9	0	12-2001	Kageyama	358	1.15	
	6	3	4	3	3	2	0	01-2002	Fairchild	709	224	
	6	3	6	3	5	1	9	02-2002	Levi	716	16	
	6	3	7	4	3	1	8	04-2002	Hayes	710	107	
	6	3	8	9	5	4	4	05-2002	Katagiri	713	300	

Sheet 2 of 2

	6	4	1	4	4	7	6	07-2002	Yagi	324	127	
	6	4	3	0	7	1	1	08-2002	Sekizawa	714	47	
	6	5	4	9	9	4	7	04-2003	Suzuki	709	229	
	6	6	2	8	6	5	3	09-2003	Salim	370	389	
	6	6	4	0	3	3	4	10-2003	Rasmussen	717	171	
	6	7	3	4	9	8	5	05-2004	Ochiai	358	1.15	
	6	7	7	1	6	4	6	08-2004	Sarkissian	370	392	
	6	7	7	9	0	0	4	08-2004	Zintel	709	227	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

"Jini Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/jini1_1.pdf, pp. 1-20.

"Jini Device Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/devicearch1_1.pdf, pp. 1-14.

Sollins, K., "The TFTP Protocol (Revision 2.0)", MIT, July 1992. Available from Internet: http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1350.html, pp. 1-10.

EXAMINER

DATE CONSIDERED 7/12/07

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

IFW

AF

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL REJECTION

Do not enter. 7/12/05
gm

Sir or Madam:

In response to the office action mailed April 28, 2005, please re-examine the above-identified application in view of the following amendment and remarks.

IFW

AF

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel	)	
	)	
Serial No.: 09/611,775	)	
	)	
Filed: July 7, 2000	)	Examiner: Simitoski, Michael J.
	)	
For: Real Time Firewall/Data Protection Systems and Methods	)	Group Art Unit: 2134
	)	
	)	
	)	

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER FINAL REJECTION

Sir or Madam:

In response to the office action mailed April 28, 2005, please re-examine the above-identified application in view of the following amendment and remarks.

IN THE CLAIMS:

1. (currently amended) A method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets, the method comprising the steps of:

receiving a packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet, wherein the packet is selectively altered to be invalid if a determination has not been made as to whether the packet is valid or invalid by the time the end portion of the packet is received.

2. (original) The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. (original) The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. (original) The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. (original) The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. (original) The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. (original) The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. (original) The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. (original) The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. (original) The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. (original) The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. (original) The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. (original) The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. (original) The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. (original) The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. (original) The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. (original) The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. (original) The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. (original) The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. (original) The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. (original) The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. (original) The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. (original) The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. (original) The method of claim 25, wherein the at least one light source comprises an LED.

30. (original) The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. (currently amended) A system for filtering packets of data between at least an external network and an internal network, wherein data is transmitted and received in the form of a plurality of packets, comprising:

a first interface circuit for coupling data packets to and from the external network;

a second interface circuit for coupling data packets to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria, wherein the packet is selectively altered to be invalid if a determination has not been made as to whether the packet is valid or invalid by the time the end portion of the packet is received .

32. (original) The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. (original) The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. (original) The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. (original) The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. (original) The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. (original) The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the packet has

been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. (original) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. (original) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. (original) The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. (original) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. (original) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. (original) The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. (original) The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. (original) The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. (original) The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. (original) The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. (original) The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. (original) The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. (original) The system of claim 55, wherein the at least one light source comprises an LED.

60. (original) The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. (original) The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. (original) The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. (original) The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. (original) The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. (original) The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. (original) The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

REMARKS

Claims 1-66 are in the application. Claims 1-38 and 40-66 were rejected. Claim 39 was objected to but otherwise indicated as allowable. With respect to claim 39, the Examiner noted that the prior art does not teach or suggest invalidating a packet if the decision/result is not received by the time the end portion/last cell is received.

While Applicant submits that the prior art is distinguishable in various respects, in an effort to expedite prosecution Applicant has amended independent claims 1 and 31 to incorporate the allowable subject matter as noted by the Examiner. Thus, with the independent claims amended to incorporate allowable subject matter, all claims should now be in condition for allowance, and such is respectfully requested.

Applicant also wishes to note that there four applications filed by Applicant based on the same product development efforts. These are:

Ser. No.	Status	Filing Date	Examiner/Art Unit
09/611,775	Pending	Jul. 7, 2000	Simitoski/2134
09/745,599	Pending	Dec. 21, 2000	Gold/2157
09/746,519	Pending	Dec. 21, 2000	Levitan/2662
09/746,107	Pending	Dec. 21, 2000	Luu/2141

Applicant has reviewed these applications and herewith is submitting an IDS that cross-cites the art from the other applications. The form 1449 includes all art cited in the four applications. For the convenience of the Examiner, on the form 1449 attached to the IDS all references previously considered in this application have been crossed-out (the 1449 reflects all prior art cited in the four applications).

Reconsideration and allowance is requested.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251.

No new matter has been added.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

June 27, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

I hereby certify that this is being sent to the USPTO via
Fed Ex on the date indicated above.

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	)	
	)	
Krumel	)	
	)	Art Unit: 2134
Serial No.: 09/611,775	)	
	)	
Filed: July 7, 2000	)	Examiner: Simitoski
	)	
For: Real Time Firewall/Data Protection Systems and Methods	)	
	)	
	)	

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

1. Pursuant to 37 C.F.R. 1.97 and 1.98, and in compliance with 37 C.F.R. 1.56, the Office's attention is directed to the patents, publications and other information listed on the attached PTO-1449. A copy of each listed document is enclosed except for: (a) pending applications or (b) those previously cited or submitted to the Office in the following application(s) upon which this application relies for an earlier filing date under 35 U.S.C. 120:

Serial No.: ______________

Filing Date: ______________

Regarding the document(s), publication(s) or other information listed on the attached PTO-1449, Applicant(s) believe(s) the same may qualify as "prior" art to this application and should be treated accordingly, although Applicant(s) reserve(s) the right to contest the prior art status of any document, publication or information cited herein.

2. Regarding each listed document that is not in the English language, an English-language translation accompanies this Statement as indicated on the attached PTO-1449 or a concise explanation of the relevance of the document is set forth in the following documents(s):

(a) ___ Copy of each English language version of a search report indicating the degree of relevance found by the foreign office of each document being submitted from the search report.

(b) ___ Attachment entitled "Concise Explanation of Relevance of Non-English Language Documents."

3. Pursuant to 37 C.F.R. 1.97(b) this Statement is being filed (one must be checked):

(a) ___ Within 3 months of the filing date or date of entry into the National Stage.

(b) ___ Before the mailing date of a first Office Action on the merits. If this Statement is not filed before the mailing date of a first Office Action on the merits, the required certification is given below or, in the absence thereof, the Office is authorized to charge the required fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251 for consideration of this Statement.

(c) ___ After the period set forth in 37 C.F.R. 1.97(b) but before the mailing date of either a final action or a notice of allowance.

(1) ___ The required certification is given below, <u>or</u>

(2) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(p) for consideration of this Statement, or

(3) ___ Charge the fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251

(d) X After the mailing date of either a final action or a notice of allowance, but before payment of the issue fee. Petition hereby is made for consideration of this Statement and the required certification is indicated below.

(1) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(i)(1), or

(2) X Charge the fee set forth in 37 C.F.R. 1.17(i)(1) to Deposit Account No. 50-0251.

4. Certification (if applicable)

(a) ___ The undersigned hereby certifies that each item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than 3 months prior to the filing of this Statement.

(b) __ The undersigned hereby certifies that no item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the undersigned's knowledge after making reasonable inquiry, was known to any individual designated in 37 C.F.R. 1.56(c) more than 3 months prior to the filing of this Statement.

5. The Commissioner is hereby authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

27 ARL
June ~~26~~, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
(408) 868-1516

I hereby certify that the foregoing is being sent by FedEx on the date indicated above.

Sheet 1 of 2

Form PTO-1449 U.S. DEPARTMENT OF COMMERCE (REV. 7-92) Patent and Trademark Office **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** (Use several sheets if necessary)	Attorney's Docket Number 802-001	Serial No. 09/611,775
	Applicant(s): Krumel	
	Filing Date: 7/7/00	Group Art Unit: 2134

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
	5	3	4	3	4	7	1	08-1994	Cassagnol	370	401	
	5	4	2	6	3	7	8	6/20/95	Ong	326	39	
	5	4	2	6	3	7	9	06-1995	Trimberger	326	39	
	5	5	9	0	0	6	0	12-1996	Granville	702	155	
	5	7	4	5	2	2	9	04-1998	June	356	73	
	5	7	9	4	0	3	3	8/11/98	Aldebert et al.	395	653	
	5	9	0	3	5	6	6	05-1999	Flammer	370	406	
	5	9	7	4	5	4	7	10-1999	Klimenko	713	2	
	6	0	2	0	7	5	8	02-2000	Patel	326	40	
	6	0	7	6	1	6	8	06-2000	Fiveash	713	201	
	6	0	4	9	2	2	2	4/11/00	Lawmann	326	38	
	6	0	5	2	7	8	5	04-2000	Lin	709	225	
	6	0	7	8	7	3	6	6/20/00	Guccione	395	500.17	
	6	0	9	2	1	2	3	07-2000	Steffan	710	8	
	6	1	5	1	6	2	5	11-2000	Swales	709	218	
	6	1	7	5	8	3	9	01-2001	Takao	715	500	
	6	1	8	2	2	2	5	01-2001	Hagiuda	713	201	
	6	2	1	5	7	6	9	04-2001	Ghani	370	230	
	6	3	1	0	6	9	2	10-2001	Fan	358	1.14	
	6	3	2	6	8	0	6	12-2001	Fallside	326	38	
	6	3	3	3	7	9	0	12-2001	Kageyama	358	1.15	
	6	3	4	3	3	2	0	01-2002	Fairchild	709	224	
	6	3	6	3	5	1	9	02-2002	Levi	716	16	
	6	3	7	4	3	1	8	04-2002	Hayes	710	107	
	6	3	8	9	5	4	4	05-2002	Katagiri	713	300	

	6	4	1	4	4	7	6	07-2002	Yagi	324	127	
	6	4	3	0	7	1	1	08-2002	Sekizawa	714	47	
	6	5	4	9	9	4	7	04-2003	Suzuki	709	229	
	6	6	2	8	6	5	3	09-2003	Salim	370	389	
	6	6	4	0	3	3	4	10-2003	Rasmussen	717	171	
	6	7	3	4	9	8	5	05-2004	Ochiai	358	1.15	
	6	7	7	1	6	4	6	08-2004	Sarkissian	370	392	
	6	7	7	9	0	0	4	08-2004	Zintel	709	227	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

		"Jini Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/jini1_1.pdf, pp. 1-20.
		"Jini Device Architecture Specifications." Version 1.1, Sun Microsystems, Inc., October 2000. Available from Internet: http://www.sun.com/jini/specs/devicearch1_1.pdf, pp. 1-14.
		Sollins, K., "The TFTP Protocol (Revision 2.0)", MIT, July 1992. Available from Internet: http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1350.html, pp. 1-10.

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

PTO/SB/06 (08-03)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number 09/611 725

6-29-05

CLAIMS AS ~~FILED – PART I~~ Amended

	(Column 1)	(Column 2)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE	FEE		RATE	FEE
~~BASIC FEE (37 CFR 1.16(a))~~				~~$____~~	OR		~~$____~~
TOTAL CLAIMS (37 CFR 1.16(c))	66 minus ~~20~~ 66 =	* 0	X $____ =	0	OR	X $____ =	
INDEPENDENT CLAIMS (37 CFR 1.16(b))	2 minus 3 =	* 0	X $____ =	0	OR	X $____ =	
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(d))		0	+ $____ =	0	OR	+ $____ =	
			TOTAL	0	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

CLAIMS AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
	Total (37 CFR 1.16(c))	*	Minus	**	=	X $____ =		OR	X $____ =	
	Independent (37 CFR 1.16(b))	*	Minus	***	=	X $____ =		OR	X $____ =	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $____ =		OR	+ $____ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1)		(Column 2)	(Column 3)					
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
	Total (37 CFR 1.16(c))	*	Minus	**	=	X $____ =		OR	X $____ =	
	Independent (37 CFR 1.16(b))	*	Minus	***	=	X $____ =		OR	X $____ =	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $____ =		OR	+ $____ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1)		(Column 2)	(Column 3)					
AMENDMENT C		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
	Total (37 CFR 1.16(c))	*	Minus	**	=	X $____ =		OR	X $____ =	
	Independent (37 CFR 1.16(b))	*	Minus	***	=	X $____ =		OR	X $____ =	
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(d))					+ $____ =		OR	+ $____ =	
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 04/28/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 04/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	**Application No.** 09/611,775	**Applicant(s)** KRUMEL, ANDREW K.	
	Examiner Michael J. Simitoski	**Art Unit** 2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 March 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-66* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38 and 40-66* is/are rejected.

7)☒ Claim(s) *39* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. The response of 3/7/2005 was received and considered.

2. Claims 1-66 are pending.

Response to Arguments

3. Applicant's arguments filed 3/7/2005 have been fully considered but they are not persuasive.

Applicant's response (p. 10, ¶4 – p. 11, ¶1) asserts that Xu teaches away from the presently claimed invention because ATM uses a unit of data transmission called a cell. Further, Applicant's response (p. 11, ¶2) asserts that Xu requires one or a plurality of ATM cells/packets to be received and processed and finds only disclosure addressing the need to received one or more entire ATM cells/packets and therefore Xu is "directly opposing" the claimed invention. However, the Examiner disagrees with Applicant's assertion (p. 11, ¶1) that "The ATM cell in Xu, to the extent that a proper correspondence may be drawn, corresponds to a packet in the present claims". The Xu reference teaches transferring packets, the packets being sent in units of a cell. Further, despite the fact that ATM transmits "cell" as one unit of transmission, ATM is still a "packet-based" network, wherein data is transmitted and received in the form of a plurality of packets" because ATM cells carry packets. As such, Xu discloses allowing all cells of a packet except the last one (end portion of the packet) to be passed, where the last portion of the packet (last cell) is selectively altered/randomly generated to be invalid if it was determined that the packet should be an invalid/unsafe packet (p. 277, ¶3).

Applicant's response (p. 11, ¶3) asserts that the claimed invention uses the packet as the unit of data transmission and that the packet is analyzed to determine whether the end portion should be modified. However, as described above, Xu transmits and receives packets and filtering decisions are made based on that – the result of which is the modification of the end portion of the packet (last ATM cell).

Claim Objections

4. Claim 16 is objected to because of the following informalities: The claim depends upon "claim 16". *For the purposes of this office action, claim 16 is understood to depend upon claim 15.* Appropriate correction is required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

6. Claims 1-4, 11-16, 31-38, 40 & 41 are rejected under 35 U.S.C. 102(a) as being anticipated by "Design of A High-Performance ATM Firewall" by Xu.

Regarding claim 1, Xu teaches receiving a packet from the external computing system/WAN over the network (p. 272 §2.1), the packet having at least a first portion/header and an end portion/last cell, and transmitting/passing the packet to the internal computing system/LAN (p. 277 ¶2-4), in parallel with the step of receiving and transmitting the packet, determining characteristics/class of the packet from the first portion/header (p. 272 §2.1, p. 277 ¶3), in parallel with the step of receiving and transmitting the packet, performing a plurality of checks/TCP/IP rules on the packet (p. 272 ¶1, p. 275 ¶1), wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks (p. 280 ¶1-3 & p. 287 ¶1), in parallel with the step of receiving an transmitting the packet, determining if the packet should be a valid/safe packet or an invalid/unsafe packet based on the plurality of checks/rules (pp. 275-278 §2.2.3), and after receiving the end portion/last cell of the packet, selectively altering/passing or generating randomly the end portion of the packet based on whether the packet has been determined to be a valid/safe packet or an invalid/unsafe packet, wherein the packet is selectively altered/generated randomly to be invalid/unsafe if it was determined that the packet should be an invalid/unsafe packet (p. 277 ¶2).

Regarding claim 2, Xu discloses the packet being analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system/LAN (p. 277 ¶2-3).

Regarding claim 3, Xu discloses examining the packet before the last cell has arrived (p. 277 ¶2-3)

Regarding claim 4, Xu discloses determining a packet invalid/unsafe if it is determined that the packet is harmful/dangerous (p. 272 §2.1 & p. 278 ¶2).

Regarding claim 11, Xu discloses the plurality of checks/rules being performed with a programmable logic device/ATM firewall with cache, wherein logic within the programmable logic device/ATM firewall with cache is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet (p. 276 ¶2-3).

Regarding claim 12, Xu discloses a physical interface/input module receiving the packet from the network (p. 284 §4.2) wherein the packet is coupled to the programmable logic device/ATM firewall with cache, wherein the packet is coupled from the programmable logic device to a second physical interface/output module (p. 286 §4.3) for transmission to the internal computing system/LAN (p. 282 Fig. 2 & p. 283 §4.1 & Fig. 3).

Regarding claim 13, Xu discloses the programmable logic device/ATM firewall with cache performing a plurality of checks while the packet is being coupled from the first physical interface/input module to the second physical interface/output module (pp. 284-286 & p. 277 ¶2-4).

Regarding claims 14 & 15, Xu discloses filtering based on port numbers (p. 275 ¶1).

Regarding claim 16, Xu discloses filtering based on IP addresses (source and destination) (p. 275 ¶1).

Regarding claim 31, Xu discloses a first interface circuit/input module for coupling data packets to and from an external network/WAN (p. 282 Fig. 2 & p. 284 §4.2), a second interface circuit/output module (p. 286 §4.3 & p. 283 Fig. 3) for coupling data packets to and from an internal network/LAN (p. 282 Fig. 2 & p. 283 §4.1), a programmable logic device/ATM firewall with cache coupled between the first interface circuit/input module and the second interface circuit/output module (p. 282 Fig. 2 & p. 283 Fig. 3), wherein as a packet is being received and transmitted between the first and second interface circuits (p. 282 §2.1), the packet is simultaneously subjected to a plurality of filtering criteria/TCP/IP rules (p. 272 ¶1 & p. 275-278 §2.2.3) by the programmable logic device/ATM firewall with cache, wherein an end portion/last cell of the packet is selectively altered/passed or generated randomly by the programmable logic device based on the filtering criteria/rules (p. 277 ¶2).

Regarding claim 32, Xu discloses the filtering criteria determining whether the packet is to be a valid/safe packet or an invalid/unsafe packet, wherein the packet is selectively altered/generated randomly to be invalid/unsafe if it was determined that the packet should be an invalid/unsafe packet (p. 277 ¶2).

Regarding claim 33, Xu discloses determining characteristics/class (p. 272 §2.1, p. 277 ¶3), of a packet and a filter portion/call-screening service that subjects the packet to a plurality of checks/TCP/IP rules on the packet (p. 272 ¶1, p. 273 §2.2.1 & p. 275 ¶1), while the packet is being received and transmitted between the first and second interface circuits (p. 277 ¶2-3).

Regarding claim 34, Xu discloses a stateful filter portion/packet-filter (p. 272 §2.1, p. 273 §2.2.1, p. 285 ¶2 & Fig. 5) and a non-stateful filter portion/traffic-monitor (p. 272 §2.1, p. 273 §2.2.1 & p. 282 Fig. 2).

Regarding claim 35 & 36, Xu discloses the stateful filter portion/packet-filter subjecting the packet to one or more stateful filtering criterion/decision on current packet (p. 285 ¶2) while the non-stateful filter portion/rules (p. 275 ¶1) subjecting the packet to one or more non-stateful filtering criterion (p. 273 §2.2.1, p. 280 ¶1 & p. 285 ¶2).

Regarding claim 37, Xu discloses a result aggregator logic/output module that receives one ore more signals/decision from the stateful filter portion and the non-stateful filter portion (p. 292 ¶1), wherein based on the received signals/decision the result aggregator logic/OM controls whether the packet is selectively altered to be invalid/dropped (p. 277 ¶2 & p. 292 ¶1).

Regarding claim 38, Xu discloses the result aggregator logic/OM receiving a completion signal/decision that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria (p. 292 ¶3).

Regarding claim 40, Xu discloses the packet being subjected to the plurality of filtering criteria/rules (p. 273 §2.2.1) in parallel with the packet being received and transmitted between the first and second interface circuits/modules (p. 280 ¶1-3 & p. 287 ¶1), wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received (p. 277 ¶2-4).

Regarding claim 41, Xu discloses the packet being subjected to the plurality of filtering criteria in real time (p. 277 ¶2-3) with the packet being received and transmitted between the first and second interface circuits/modules (p. 283 Fig. 3).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 30, 44 & 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu.

Regarding claim 44, Xu lacks basing a user-controlled switch's state (effectively enabling/disabling a predetermined portion of the filtering criteria/rules) on whether a computer coupled to the internal network is controlled to operate in a client mode or a server mode. However, official notice is hereby taken that it is known in the network firewall art/network security art that a client/workstation requires different traffic needs (open ports, bandwidth, limitations on number of connections) than does a server. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to base a user-controlled switch's state on whether a computer coupled to the internal network is operating as a client or server. One of ordinary skill in the art would have been motivated to perform such a modification, as it was known in the art to do so.

Regarding claims 30 & 60, Xu lacks a speaker to provide feedback. However, official notice is hereby taken that it was known in the art, as the time the invention was made, to provide a speaker, such as a PC main board speaker, to provide audio feedback (for example on errors). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a speaker in Xu's system to provide feedback. One of ordinary skill in the art would have been motivated to perform such a modification as it was known in the art to do so.

9. Claims 5-8, 10, 17-19, 23-27, 29, 42, 43, 45, 46, 47-49, 53-57, 59, 61-63 & 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claims 1 & 31 above, in view of "PacketShaper 4000 Getting Started Version 4.0" by Packeteer.

Regarding claims 5-8, 10, 42, 43, 45, 61-63 & 66, Xu discloses a firewall system and lacks detailed physical description of the device(s), and hence lacks a physical switch affecting the operation of the firewall. However, Packeteer teaches that it is known to include a power switch to enable/disable function of a device, such as an on/off switch (p. 7). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include an on/off toggle switch, thereby affecting the checks based on the state of the switch, affecting the configuration of the checking circuit (on/off), enabling/disabling the checks (on/off). The plurality of checks would selectively perform based on the state an on/off switch.

An on/off switch would also control the configuration (on/off). One of ordinary skill in the art would have been motivated to perform such a modification, as it was well known in the art to do so, as taught by Packeteer (p. 7).

Regarding claims 23, 24, 46, 53 & 54, Xu discloses a firewall system, as modified above, but lacks detailed physical description of the device(s), and hence lacks a reset switch. However, Packeteer teaches that it is known to include a power switch/reset switch to enable/disable/reset function of a device, such as an on/off switch (p. 7). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a physical reset switch/power switch to reset the device described by Xu. One of ordinary skill in the art would have been motivated to perform such a modification, as it was well known in the art to do so, as taught by Packeteer (p. 7).

Regarding claims 17-19, 25, 26, 29, 47-49, 55, 56 & 59, Xu discloses a system, as modified above, but lacks visual feedback that the system is operational, the system is subject to filtering criteria, a light source indicative of the operating status having a first color or second color depending on the status and lacks an LED. However, Packeteer teaches that it is known in the art to provide a "status LED", being green or amber in color depending on whether shaping (filtering) is on/operational (p. 41) on a hardware packet-shaper/packet-filter (p. 1). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a status LED in Xu's system. One of ordinary skill in the art would have been motivated to perform such a modification to convey status information, as was known in the art, as taught by Packeteer (pp. 1 & 41).

Regarding claims 27 & 57, Xu discloses a system, as modified above, but lacks a light source that is selectively controlled to blink depending on the operating status. However, Packeteer teaches that it is known to include "network LEDs" to that flicker/blink when transmission or receiving activity occurs (p. 41) in a hardware packet-shaper/packet-filter (p. 1). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include network LEDs in Xu's system. One of ordinary skill in the art would have been motivated to perform such a modification to convey activity information, as was known in the art, as taught by Packeteer (pp. 1 & 41).

10. Claims 20-22 & 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu in view of Packeteer, as applied to claims 18 & 47 above, in further view of "BlackICE Pro User's Guide Version 2.0" by Network Ice Corporation (NIC). Xu discloses a system, as modified above, but lacks audio or visual feedback when the system has rejected one or more packets, when it is suspected to be under attack, or the severity of the attack. However, NIC teaches that to make users aware of attacks and spot trends and patterns of attacks, it is useful to provide a list of possible attacks on the system (p. 3 Fig. 3) and indicating the severity (p. 21). Further, when a critical or serious event occur, they can cause the blocking of addresses and ports/rejection of packets, and indicate this to the user (p. 21 & p. 37). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use visual indicators to indicate when the system has rejected packets and when the system is under attack and to indicate the severity of an attack. One of ordinary skill in the art would have been

motivated to perform such a modification to make users aware of attacks and to spot trends, as taught by NIC (pp. 1, 3, 21 & 37).

11. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claim 7 above, in view of U.S. Patent 6,052,788 to Wesinger, Jr. et al. (Wesinger). Xu discloses a system, as modified above to include a user-controlled switch such as a power switch, but lacks the circuit being configured or reconfigured based on commands from the internal computing system/LAN. However, Wesinger that configuration of firewalls may be easily accomplished by running a "configurator" which provides a Web-based front-end for editing configuration files, preferably from a secured client (col. 9 lines 31-46). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to change the firewall configuration based on commands from the internal computing system/LAN/secure client (through a Web-browser interface). One of ordinary skill in the art would have been motivated to perform such a modification to easily accomplish firewall configuration, as taught by Wesinger (col. 9 lines 31-46).

12. Claims 28 & 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu in view of Packeteer, as applied to claims27 & 57 above, in further view of "BlackICE Pro User's Guide Version 2.0" by Network Ice Corporation (NIC) in further view of U.S. Patent 6,133,844 to Ahne et al. (Ahne). Xu discloses a system, as modified above, but lacks a light blinking at a rate indicative of a severity level of an attack. Packeteer teaches blinking LEDs indicating traffic activity (pp. 1 & 41). NIC teaches indicating a severity level of an attack to a user (pp. 1, 3, 21

& 37). Ahne teaches that on a printing device, an LED's blink rate, *inter alia*, can be altered and the LEDs can be used to convey the operating status of the device (col. 7 lines 22-52 & col. 8 lines 20-37). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the blink rate of a light, as taught by Ahne, on Xu's firewall system, as suggested by Packeteer, to indicate the severity level of an attack, as taught by NIC. One of ordinary skill in the art would have been motivated to perform such a modification to convey operating status to a user, as taught by Ahne (col. 7 lines 22-52 & col. 8 lines 20-37).

13. Claims 64 & 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claim 61 above, in view of U.S. Patent 5,905,859 to Holloway et al. (Holloway). Xu discloses user specified criteria/specifying or updating rules via firewall management service (p. 281 §2.2.6), but lacks details about the specific hardware involved and therefore, lacks the configuration data transferred from configuration software via a cable attachment. However, Holloway teaches that it is common in the art of managing network devices to supply an RS232 serial port connection to change configuration parameters from a local console (col. 7 lines 11-32). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to transfer configuration parameters via a cable attachment/RS232. One of ordinary skill in the art would have been motivated to perform such a modification to enable a local console to change configuration parameters, as is known in the art to do, as taught by Holloway (col. 7 lines 11-32).

Allowable Subject Matter

14. Claim 39 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

15. The following is a statement of reasons for the indication of allowable subject matter:

Regarding claim 39, the prior art relied upon fails to teach or suggest invalidating a packet if the decision/result is not received by the time the end portion/last cell is received.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. The '662 patent reference is cited for teaching a firewall modifying the checksum in the data portion of an IEEE 1394 packet to invalidate the packet at the receiving end, when a security device decides the packet is to be blocked.

b. The Newton and Derfler, Jr. references are cited for teaching ATM;

c. The "ATM", "ATM Efficiency" web references and '695, '316, '797, '816 & '992 patent references are cited for teaching the burst size (set of ATM cells) equal to one IP packet, effectively transferring on burst (or frame) per IP packet.

17. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Simitoski whose telephone number is (571) 272-3841. The examiner can normally be reached on Monday - Thursday, 6:45 a.m. - 4:15 p.m.. The examiner can also be reached on alternate Fridays from 6:45 a.m. – 3:15 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached at (571) 272-3838.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks
Washington, DC 20231

Or faxed to:

(703)746-7239 (for formal communications intended for entry)

Or:

(571)273-3841 (Examiner's fax, for informal or draft communications, please label "PROPOSED" or "DRAFT")

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MJS
April 22, 2005

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Notice of References Cited

Application/Control No. 09/611,775	Applicant(s)/Patent Under Reexamination KRUMEL, ANDREW K.		
Examiner Michael J. Simitoski	Art Unit 2134	Page 1 of 1	

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,530,695 A	06-1996	Dighe et al.	370/232
	B	US-5,657,316 A	08-1997	Nakagaki et al.	370/394
	C	US-6,011,797 A	01-2000	Sugawara, Tsugio	370/395.51
	D	US-6,134,662 A	10-2000	Levy et al.	713/200
	E	US-6,608,816 B1	08-2003	Nichols, Kathleen M.	370/235
	F	US-6,791,992 B1	09-2004	Yun et al.	370/415
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	AARNet. "ATM", <http://www.aarnet.edu.au/engineering/networkdesign/mtu/atm.html>.
	V	Derfler, Jr., Frank J. et al. How Networks Work, September 2000, pp. 162-167.
	W	Newton, Harry. Newton's TELECOM Dictionary", 2003 CMP Books, pp. 78-79.
	X	Unknown. "ATM Efficiency", <http://homepages.uel.ac.uk/u0227461/Website/efficiency.htm>.

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	**Application/Control No.** 09/611,775	**Applicant(s)/Patent under Reexamination** KRUMEL, ANDREW K.
	Examiner Michael J. Simitoski	**Art Unit** 2134

SEARCHED

Class	Subclass	Date	Examiner
713	201 (text)	4/21/2005	MJS
709	229, 249	4/21/2005	MJS
709	225	4/21/2005	MJS
370	356, 389	4/22/2005	MJS
370	392, 401	4/22/2005	MJS
370	395.21	4/22/2005	MJS
370	395.32	4/22/2005	MJS
	above=txt		

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
Updated previous class search and EAST search since last action.	4/22/2005	MJS
See new EAST and NPL search notes.	4/22/2005	MJS

Index of Claims	**Application/Control No.** 09/611,775	**Applicant(s)/Patent under Reexamination** KRUMEL, ANDREW K.
	Examiner Michael J. Simitoski	**Art Unit** 2134

√	**Rejected**	–	**(Through numeral) Cancelled**	N	**Non-Elected**	A	**Appeal**
=	**Allowed**	÷	**Restricted**	I	**Interference**	O	**Objected**

Claim		Date								
Final	Original	4/22/05								
	1	√								
	2	√								
	3	√								
	4	√								
	5	√								
	6	√								
	7	√								
	8	√								
	9	√								
	10	√								
	11	√								
	12	√								
	13	√								
	14	√								
	15	√								
	16	√								
	17	√								
	18	√								
	19	√								
	20	√								
	21	√								
	22	√								
	23	√								
	24	√								
	25	√								
	26	√								
	27	√								
	28	√								
	29	√								
	30	√								
	31	√								
	32	√								
	33	√								
	34	√								
	35	√								
	36	√								
	37	√								
	38	√								
	39	O								
	40	√								
	41	√								
	42	√								
	43	√								
	44	√								
	45	√								
	46	√								
	47	√								
	48	√								
	49	√								
	50	√								
	51	√								
	52	√								
	53	√								
	54	√								
	55	√								
	56	√								
	57	√								
	58	√								
	59	√								
	60	√								
	61	√								
	62	√								
	63	√								
	64	√								
	65	√								
	66	√								
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

U.S. Patent and Trademark Office

Part of Paper No. 04222005

Non-patent literature search information - 4/22/05

09611775
Michael J. Simitoski
Michael.Simitoski@uspto.gov
(571) 272-3841

Google

IP over atm burst size length frame cell cells packet
"single burst" atm ip packet cells
atm burst ip packet
atm cells frames cell frame

ACM

Other

Search tool	Search Terms

Inventor Search performed: (the following, if any, found particularly relevant)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L18	26	@ad<"20000707" and atm and (burst adj2 (size length)) same (packet adj2 (size length))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:32
L17	139	@ad<"20000707" and atm and (burst adj2 (size length)) same packet	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:32
L14	5	@ad<"20000707" and (($4ip adj packet) with burst) and atm	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:32
L16 ABS + kwic	0	@ad<"20000707" and (($4ip adj packet) same ((one single) adj burst)) and atm	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:22
L15	0	@ad<"20000707" and (($4ip adj packet) with ((one single) adj burst)) and atm	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:22
L13	22	@ad<"20000707" and (($4ip adj packet) same burst) and atm	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:22
L12	8	@ad<"20000707" and (ethernet adj frame) same burst and atm	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:21
L11	19	@ad<"20000707" and (ethernet adj frame) same burst	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:19
L10	9	@ad<"20000707" and atm and (packet with burst with cells!) and (ethernet adj frame)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:18
L9	ABS 65	@ad<"20000707" and atm and (packet with burst with cells!)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:16
L8	125	@ad<"20000707" and atm and (packet with burst) and (burst with cells!)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:13
L7	4	@ad<"20000707" and (packet adj filter$3) and (((chang$3 substitut$4 alter$5 modif$7) near2 packet near2 (body payload data)) same (checksum crc))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 08:12
L6	5	@ad<"20000707" and firewall$3 and (((chang$3 substitut$4 alter$5 modif$7) near2 packet near2 (body payload data)) same (checksum crc))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 07:21
L5	24	@ad<"20000707" and firewall$3 and (((chang$3 substitut$4 alter$5 modif$7) near2 packet) same (checksum crc))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 07:19
L4 ABS + kwic	21	@ad<"20000707" and firewall$3 and (((chang$3 substitut$4 alter$5 modif$7) near2 packet) same (checksum))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 07:19
L3	13	@ad<"20000707" and firewall$3 and (((chang$3 substitut$4 alter$5 modif$7) near2 packet) same (unsafe invalid intru$6))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 07:19
L2	1	@ad<"20000707" and firewall$3 and ((chang$3 substitut$4 alter$5 modif$7) near2 (end) near2 packet)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/04/22 07:13

L1	200	@ad<"20000707" and firewall$3 and ((chang$3 substitut$4 alter$5 modif$7) near2 packet)	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/04/22 07:12

S15	1136	@ad<"20000707" and 713/201.ccls.	US-PGPUB;
			USPAT;
			EPO; JPO;
			IBM_TDB
				OR	ON	2005/04/22 07:10

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this resubmitted amendment (claims portion only) is being sent via facsimile to 703-872-9306 on the date indicated below.

RESUBMITTED AMENDMENT (CLAIMS PORTION ONLY)

Sir or Madam:

In response to the Notice of Non-Compliant Amendment mailed February 28,2005, please re-examine the above-identified application in view of the following resubmitted amendment (claims portion only). This resubmitted amendment corrects the inadvertently incorrect claim status identifiers.

IN THE CLAIMS:

1. (currently amended) A method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets, the method comprising the steps of:

receiving a ~~communication~~ packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

2. (original) The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. (original) The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. (original) The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. (original) The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. (original) The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. (original) The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. (original) The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. (original) The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. (original) The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. (original) The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. (original) The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. (original) The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. (original) The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. (original) The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. (original) The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. (original) The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. (original) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. (original) The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. (original) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. (original) The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. (original) The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively

controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. (original) The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. (original) The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. (original) The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. (original) The method of claim 25, wherein the at least one light source comprises an LED.

30. (original) The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. (currently amended) A system for filtering packets of data between at least an external network and an internal network, wherein data is transmitted and received in the form of a plurality of packets, comprising:

a first interface circuit for coupling data packets to and from the external network;

a second interface circuit for coupling data packets to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria.

32. (original) The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is

selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. (original) The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. (original) The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. (original) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. (original) The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. (original) The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. (original) The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the packet has been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. (original) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to

selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. (original) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. (original) The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. (original) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. (original) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. (original) The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. (original) The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. (original) The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. (original) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. (original) The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. (original) The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. (original) The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. (original) The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. (original) The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. (original) The system of claim 55, wherein the at least one light source comprises an LED.

60. (original) The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. (original) The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. (original) The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. (original) The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. (original) The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. (original) The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. (original) The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

March 7, 2005
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

PATENT APPLICATION FEE DETERMINATION RECORD
Effective December 29, 1999

Application or Docket Number
9/611,275

CLAIMS AS FILED - PART I

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

	(Column 1)		(Column 2)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED		NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE					345.00	OR		690.00
TOTAL CLAIMS	66	minus 20=	*46	X$ 9=	414	OR	X$18=	
INDEPENDENT CLAIMS	2	minus 3 =	*	X39=		OR	X78=	
MULTIPLE DEPENDENT CLAIM PRESENT				+130=		OR	+260=	
				TOTAL	759	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 66	Minus	** 66	= —	X$ 9=		OR	X$18=	
Independent	* 2	Minus	*** 3	= —	X39=		OR	X78=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 66	Minus	** 66	= —	X$ 9=		OR	X$18=	
Independent	* 2	Minus	*** 3	= —	X39=		OR	X78=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=		OR	+260=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

3-7-05

AMENDMENT C

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY			OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
T tal	* 66	Minus	** 66	= 0	X$ 9=	0	OR	X$18=	
Independent	* 2	Minus	*** 3	= 0	X39=	0	OR	X78=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+130=	0	OR	+260=	
					TOTAL ADDIT. FEE	0	OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875
(Rev. 12/99)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 02/28/2005

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 02/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Notice of Non-Compliant Amendment (37 CFR 1.121)

Application No. 09/611,775	**Applicant(s)** KRUMEL, ANDREW K.
Examiner Michael J Simitoski	**Art Unit** 2134

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

The amendment document filed on _31 October 2004_ is considered non-compliant because it has failed to meet the requirements of 37 CFR 1.121. In order for the amendment document to be compliant, correction of the following item(s) is required.

THE FOLLOWING MARKED (X) ITEM(S) CAUSE THE AMENDMENT DOCUMENT TO BE NON-COMPLIANT:

□ 1. Amendments to the specification:
- □ A. Amended paragraph(s) do not include markings.
- □ B. New paragraph(s) should not be underlined.
- □ C. Other _____.

□ 2. Abstract:
- □ A. Not presented on a separate sheet. 37 CFR 1.72.
- □ B. Other _____.

□ 3. Amendments to the drawings:
- □ A. The drawings are not properly identified in the top margin as "Replacement Sheet," "New Sheet," or "Annotated Sheet" as required by 37 CFR 1.121(d).
- □ B. The practice of submitting proposed drawing correction has been eliminated. Replacement drawings showing amended figures, without markings, in compliance with 37 CFR 1.84 are required.
- □ C. Other _____.

☒ 4. Amendments to the claims:
- □ A. A complete listing of all of the claims is not present.
- □ B. The listing of claims does not include the text of all pending claims (including withdrawn claims)
- ☒ C. Each claim has not been provided with the proper status identifier, and as such, the individual status of each claim cannot be identified. Note: the status of every claim must be indicated after its claim number by using one of the following status identifiers: (Original), (Currently amended), (Canceled), (Previously presented), (New), (Not entered), (Withdrawn) and (Withdrawn-currently amended).
- □ D. The claims of this amendment paper have not been presented in ascending numerical order.
- □ E. Other: _____.

For further explanation of the amendment format required by 37 CFR 1.121, see MPEP § 714 and the USPTO website at http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/officeflyer.pdf .

TIME PERIODS FOR FILING A REPLY TO THIS NOTICE:

1. Applicant is given **no new time period** if the non-compliant amendment is an after-final amendment or an amendment filed after allowance. If applicant wishes to resubmit the non-compliant after-final amendment with corrections, the **entire corrected amendment** must be resubmitted within the time period set forth in the final Office action.

2. Applicant is given **one month**, or thirty (30) days, whichever is longer, from the mail date of this notice to supply the **corrected section** of the non-compliant amendment in compliance with 37 CFR 1.121, if the non-compliant amendment is one of the following: a preliminary amendment, a non-final amendment (including a submission for a request for continued examination (RCE) under 37 CFR 1.114), a supplemental amendment filed within a suspension period under 37 CFR 1.103(a) or (c), and an amendment filed in response to a *Quayle* action.

Extensions of time are available under 37 CFR 1.136(a) only if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action.

Failure to timely respond to this notice will result in:
Abandonment of the application if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action; or
Non-entry of the amendment if the non-compliant amendment is a preliminary amendment or supplemental amendment.

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this amendment is being sent via facsimile to 703-872-9318 on the date indicated below.

RESUBMITTED AMENDMENT

Sir or Madam:

In response to the Notice of Non-Compliant Amendment mailed October 18, 2004, please re-examine the above-identified application in view of the following resubmitted amendment and remarks. This resubmitted amendment corrects the inadvertently incorrect claim status identifiers.

IN THE CLAIMS:

1. (currently amended) A method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets, the method comprising the steps of:

receiving a ~~communication~~ packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

2. (originally presented) The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. (originally presented) The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. (originally presented) The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. (originally presented) The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. (originally presented) The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. (originally presented) The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. (originally presented) The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. (originally presented) The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. (originally presented) The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. (originally presented) The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. (originally presented) The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. (originally presented) The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. (originally presented) The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. (originally presented) The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. (originally presented) The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. (originally presented) The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. (originally presented) The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system

performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. (originally presented) The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. (originally presented) The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. (originally presented) The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. (originally presented) The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. (originally presented) The method of claim 25, wherein the at least one light source comprises an LED.

30. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. (currently amended) A system for filtering packets of data between at least an external network and an internal network, wherein data is transmitted and received in the form of a plurality of packets, comprising:

a first interface circuit for coupling data packets to and from the external network;

a second interface circuit for coupling data packets to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria.

32. (originally presented) The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. (originally presented) The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. (originally presented) The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. (originally presented) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. (originally presented) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. (originally presented) The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. (originally presented) The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. (originally presented) The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the

packet has been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. (originally presented) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. (originally presented) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. (originally presented) The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. (originally presented) The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. (originally presented) The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. (originally presented) The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. (originally presented) The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. (originally presented) The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. (originally presented) The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. (originally presented) The system of claim 55, wherein the at least one light source comprises an LED.

60. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. (originally presented) The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. (originally presented) The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. (originally presented) The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. (originally presented) The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. (originally presented) The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. (originally presented) The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

REMARKS

Claims 1-66 were in the application. Claims 1-38 and 40-66 were rejected primarily in view of Xu, either alone or combined with a number of other references. Claim 39 was objected to but indicated as allowable over the art of record.

While Applicant respectfully traverses the rejections in view of Xu (whether alone or in combination with other references), Applicant has chosen to clarify the claims to emphasize certain fundamental distinctions over the Xu reference. As all rejections were premised on an analysis of the Xu reference, Applicant submits that, for at least the reasons set forth below, Xu is readily distinguishable from the invention defined by the presently pending claims, and all claims should be allowable.

The invention defined by the presently pending claims, as amplified by the amendments to the independent claims herein, is directed to a method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets. Thus, the unit of data transmission in essence is the packet. In accordance with the claimed invention, packets having at least a first portion and an end portion are received and transmitted, while in parallel with such steps characteristics of a packet are determined from the first portion, a plurality of checks are performed on the packet, wherein at least certain of the plurality of checks are performed in parallel with other of the plurality of checks, and it is determined if the packet should be a valid packet or an invalid packet based on the plurality of checks. In accordance with the presently claimed invention, after receiving the end portion of the packet, the end portion of the packet is selectively altered based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet. Thus, as a packet is received and transmitted, it in parallel is analyzed to determine whether it should be selectively altered so as to be invalidated.

Xu, respectfully, teaches directly away from the presently claimed invention. Xu is directed to an ATM firewall design. As Xu explains, and as is well known in the art, the unit of data transmission in an ATM network is the ATM cell. The ATM cell of Xu, to the extent that a proper correspondence may be drawn, corresponds to a packet in the present claims. As such, it is clear that the invention claimed herein is neither disclosed in nor suggested by Xu.

The filtering techniques of Xu in general require one or a plurality of ATM cells/packets to be received and processed in order for filtering-type decisions to be made. Indeed, Applicant has reviewed Xu and finds only disclosure addressing the need to receive one or more entire ATM cells/packets before the decision is made whether to invalidate the transmission. This must be the case because Xu contemplates filtering IP packets, and in general IP packets typically will have a size that greatly exceeds the fixed size of an ATM cell/packet. See, for example, the discussion in Xu at pages 275-277 regarding "packet filtering service." Xu states that a recent survey showed that the average packet size in a WAN is around 348, which will occupy 8 ATM cells/packets if AAL5 is used. Including the possibility of interleaving, the arrival time between the first ATM cell/packet and the last ATM cell/packet will be 22 ATM cell times. Thus, it is clear that Xu is addressing a filtering scheme that is directly opposed to what is addressed in the present claims.

As independent claims 1 and 31 make clear, in accordance with the presently claimed invention the unit of data transmission is the packet, and during the process of receiving and transmitting a packet, the packet is analyzed and a determination is made as to whether an end portion of the packet should be selectively modified in order to invalidate the packet. Thus, unlike Xu which necessarily contemplates receiving one or a plurality of entire ATM cells/packets in order to make filtering decisions, in accordance with the presently claimed invention the process of receiving and transmitting the packet is commenced, while in parallel the filtering decisions are made so that a decision may be made prior to transmission of the end portion of the packet. The system of Xu does not operate in this manner, and in fact Xu teaches away from operation in this manner.

Accordingly, Applicant submits that Xu is readily distinguishable from the claimed invention, whether considered alone or in combination with the other references. Reconsideration and allowance is requested.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251.

No new matter has been added.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

October 31, 2004
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 29, 1999

Application or Docket Number

9/611,225

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
FOR	NUMBER FILED	NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	66 minus 20=	* 46
INDEPENDENT CLAIMS	2 minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT		

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
	345.00	OR		690.00
X$ 9=	414	OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL	759	OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 66	Minus	** 66	= —
Independent	* 2	Minus	*** 3	= —
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 66	Minus	** 66	= —
Independent	* 2	Minus	*** 3	= —
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
T tal	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 10/18/2004

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 10/18/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

Notice of Non-Compliant Amendment (37 CFR 1.121)

The amendment document filed on 6-28-04 is considered non-compliant because it has failed to meet the requirements of 37 CFR 1.121. In order for the amendment document to be compliant, correction of the following item(s) is required. **Only the corrected section of the non-compliant amendment document must be resubmitted (in its entirety), e.g., the entire "Amendments to the claims" section of applicant's amendment document must be re-submitted.** 37 CFR 1.121(h).

THE FOLLOWING CHECKED (X) ITEM(S) CAUSE THE AMENDMENT DOCUMENT TO BE NON-COMPLIANT:

□ 1. Amendments to the specification:
- □ A. Amended paragraph(s) do not include markings.
- □ B. New paragraph(s) should not be underlined.
- □ C. Other__________

□ 2. Abstract:
- □ A. Not presented on a separate sheet. 37 CFR 1.72.
- □ B. Other__________

□ 3. Amendments to the drawings: __________

☑ 4. Amendments to the claims:
- □ A. A complete listing of <u>all</u> of the claims is not present.
- □ B. The listing of claims does not include the text of all pending claims (including withdrawn claims)
- ☑ C. Each claim has not been provided with the proper status identifier, and as such, the individual status of each claim cannot be identified. Note: the status of every claim must be indicated after its claim number by using one of the following 7 status identifiers: (Original), (Currently amended), (Canceled), (Withdrawn), (Previously presented), (New) and (Not entered).
- □ D. The claims of this amendment paper have not been presented in ascending numerical order.
- ☑ E. Other: Status identifiers are incorrect.

*For further explanation of the amendment format required by 37 CFR 1.121, see MPEP Sec. 714 and the USPTO website at http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/officeflver.pdf.

If the non-compliant amendment is a **PRELIMINARY AMENDMENT**, applicant is given ONE MONTH from the mail date of this letter to supply the corrected section which complies with 37 CFR 1.121. Failure to comply with 37 CFR 1.121 will result in non-entry of the preliminary amendment and examination on the merits will commence without consideration of the proposed changes in the preliminary amendment(s). This notice is not an action under 35 U.S.C. 132, and **this ONE MONTH time limit is not extendable.**

*If the non-compliant amendment is a reply to a **NON-FINAL OFFICE ACTION (including a submission for an RCE)**, and since the amendment appears to be a *bona fide* attempt to be a reply (37 CFR 1.135(c)), applicant is given a TIME PERIOD of ONE MONTH from the mailing of this notice within which to re-submit the corrected section which complies with 37 CFR 1.121 in order to avoid abandonment. **EXTENSIONS OF THIS TIME PERIOD ARE AVAILABLE UNDER 37 CFR 1.136(a).**

If the amendment is a reply to a **FINAL REJECTION,** this form may be an attachment to an Advisory Action. **<u>The period for response to a final rejection continues to run from the date set in the final rejection</u>**, and is not affected by the non-compliant status of the amendment.

LaTrone Martino 571-272-0538
Legal Instruments Examiner (LIE) Telephone No.

Rev. 6/04

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)
)

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this amendment is being sent via facsimile to 703-872-9318 on the date indicated below.

AMENDMENT

Sir or Madam:

In response to the office action mailed February 27, 2004, please re-examine the above-identified application in view of the following amendment and remarks. A petition for extension of time accompanies this amendment and is hereby requested.

IN THE CLAIMS:

1. (presently amended) A method for communicating data between an external computing system and an internal computing system over a packet-based network, wherein data is transmitted and received in the form of a plurality of packets, the method comprising the steps of:

receiving a ~~communication~~ packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

2. (originally presented) The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. (originally presented) The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. (originally presented) The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. (originally presented) The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. (originally presented) The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. (originally presented) The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. (originally presented) The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. (originally presented) The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. (originally presented) The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. (originally presented) The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. (originally presented) The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. (originally presented) The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. (originally presented) The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. (originally presented) The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. (originally presented) The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. (originally presented) The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. (originally presented) The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. (originally presented) The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. (originally presented) The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. (originally presented) The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. (originally presented) The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. (originally presented) The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. (originally presented) The method of claim 25, wherein the at least one light source comprises an LED.

30. (originally presented) The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. (originally presented) A system for filtering packets of data between at least an external network and an internal network, wherein data is transmitted and received in the form of a plurality of packets, comprising:

a first interface circuit for coupling data packets to and from the external network;

a second interface circuit for coupling data packets to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria.

32. (originally presented) The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. (originally presented) The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. (originally presented) The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. (originally presented) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. (originally presented) The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. (originally presented) The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. (originally presented) The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. (originally presented) The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the packet has been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. (originally presented) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. (originally presented) The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. (originally presented) The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. (originally presented) The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. (originally presented) The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback

devices selectively provide visual or audio feedback of the operation or status of the system.

48. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. (originally presented) The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. (originally presented) The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. (originally presented) The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. (originally presented) The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. (originally presented) The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. (originally presented) The system of claim 55, wherein the at least one light source comprises an LED.

60. (originally presented) The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. (originally presented) The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. (originally presented) The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. (originally presented) The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. (originally presented) The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. (originally presented) The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. (originally presented) The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

REMARKS

Claims 1-66 were in the application. Claims 1-38 and 40-66 were rejected primarily in view of Xu, either alone or combined with a number of other references. Claim 39 was objected to but indicated as allowable over the art of record.

While Applicant respectfully traverses the rejections in view of Xu (whether alone or in combination with other references), Applicant has chosen to clarify the claims to emphasize certain fundamental distinctions over the Xu reference. As all rejections were premised on an analysis of the Xu reference, Applicant submits that, for at least the reasons set forth below, Xu is readily distinguishable from the invention defined by the presently pending claims, and all claims should be allowable.

The invention defined by the presently pending claims, as amplified by the amendments to the independent claims herein, is directed to a method for communicating data between an external computing system and an internal computing system over a packet-based network, <u>wherein data is transmitted and received in the form of a plurality of packets</u>. Thus, the unit of data transmission in essence is the packet. In accordance with the claimed invention, packets having at least a first portion and an end portion are received and transmitted, while in parallel with such steps characteristics of a packet are determined from the first portion, a plurality of checks are performed on the packet, wherein at least certain of the plurality of checks are performed in parallel with other of the plurality of checks, and it is determined if the packet should be a valid packet or an invalid packet based on the plurality of checks. In accordance with the presently claimed invention, after receiving the end portion of the packet, the end portion of the packet is selectively altered based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet. Thus, as a packet is received and transmitted, it in parallel is analyzed to determine whether it should be selectively altered so as to be invalidated.

Xu, respectfully, teaches directly away from the presently claimed invention. Xu is directed to an ATM firewall design. As Xu explains, and as is well known in the art, the unit of data transmission in an ATM network is the ATM cell. The ATM cell of Xu, to the extent that a proper correspondence may be drawn, corresponds to a packet in the present claims. As such, it is clear that the invention claimed herein is neither disclosed in nor suggested by Xu.

The filtering techniques of Xu in general require one or a plurality of ATM cells/packets to be received and processed in order for filtering-type decisions to be made. Indeed, Applicant has reviewed Xu and finds only disclosure addressing the need to receive one or more entire ATM cells/packets before the decision is made whether to invalidate the transmission. This must be the case because Xu contemplates filtering IP packets, and in general IP packets typically will have a size that greatly exceeds the fixed size of an ATM cell/packet. See, for example, the discussion in Xu at pages 275-277 regarding "packet filtering service." Xu states that a recent survey showed that the average packet size in a WAN is around 348, which will occupy 8 ATM cells/packets if AAL5 is used. Including the possibility of interleaving, the arrival time between the first ATM cell/packet and the last ATM cell/packet will be 22 ATM cell times. Thus, it is clear that Xu is addressing a filtering scheme that is directly opposed to what is addressed in the present claims.

As independent claims 1 and 31 make clear, in accordance with the presently claimed invention the unit of data transmission is the packet, and during the process of receiving and transmitting a packet, the packet is analyzed and a determination is made as to whether an end portion of the packet should be selectively modified in order to invalidate the packet. Thus, unlike Xu which necessarily contemplates receiving one or a plurality of entire ATM cells/packets in order to make filtering decisions, in accordance with the presently claimed invention the process of receiving and transmitting the packet is commenced, while in parallel the filtering decisions are made so that a decision may be made prior to transmission of the end portion of the packet. The system of Xu does not operate in this manner, and in fact Xu teaches away from operation in this manner.

Accordingly, Applicant submits that Xu is readily distinguishable from the claimed invention, whether considered alone or in combination with the other references. Reconsideration and allowance is requested.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251.

No new matter has been added.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

June 28, 2004
Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel	)	
	)	
Serial No.: 09/611,775	)	
	)	
Filed: July 7, 2000	)	Examiner: Simitoski, Michael J.
	)	
For: Real Time Firewall/Data Protection Systems and Methods	)	Group Art Unit: 2134
	)	

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this amendment is being sent via facsimile to 703-872-9318 on the date indicated below.

PETITION AND FEE FOR EXTENSION OF TIME UNDER 37 C.F.R. 1.136(a)

Dear Commissioner:

Applicant hereby petitions for a one-month extension of time to respond to the Office Action mailed dated February 27, 2004. Please charge Deposit Account No. 50-0251 in the amount of $55.00 for the extension fee. An amendment responsive to the outstanding Office Action accompanies this petition.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

June 28, 2004
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number
09/611775

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS		
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	minus 20=	*
INDEPENDENT CLAIMS	minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II 6-28-04

6-28-04

AMENDMENT A		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	* 66	Minus	** 66	= —
	Independent	* 2	Minus	*** 3	= —
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=	—	OR	X$18=	
X43=	—	OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE	—	OR	TOTAL ADDIT. FEE	

AMENDMENT B		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Krumel)
)
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Simitoski, Michael J.
)
For: Real Time Firewall/Data Protection) Group Art Unit: 2134
Systems and Methods)
)

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this amendment is being sent via facsimile to 703-872-9318 on the date indicated below.

PETITION AND FEE FOR EXTENSION OF TIME UNDER 37 C.F.R. 1.136(a)

Dear Commissioner:

Applicant hereby petitions for a one-month extension of time to respond to the Office Action mailed dated February 27, 2004. Please charge Deposit Account No. 50-0251 in the amount of $55.00 for the extension fee. An amendment responsive to the outstanding Office Action accompanies this petition.

Please charge any additional fees due, or credit any overpayment, to Deposit Account No. 50-0251.

Respectfully submitted,

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

June 28, 2004
P.O. Box 3607
Los Altos, CA 94024-0607
408-868-1516

FEE ONLY

8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/611,775	07/07/2000	Andrew K. Krumel	802-001	6989

7590 02/27/2004

Loudermilk & Associates
P.O. Box 3607
Los Altos, CA 94024-0607

EXAMINER	
SIMITOSKI, MICHAEL J	

ART UNIT	PAPER NUMBER
2134	9

DATE MAILED: 02/27/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

8

8

Office Action Summary	**Application No.** 09/611,775	**Applicant(s)** KRUMEL, ANDREW K.	
	Examiner Michael J Simitoski	**Art Unit** 2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 July 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-66* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38 and 40-66* is/are rejected.

7)☒ Claim(s) *16 and 39* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 July 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 2.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. The IDS of 10/17/2000 (paper #2) has been received and considered.

2. Claims 1-66 are pending.

Claim Objections

3. Claim 16 is objected to because of the following informalities: The claim depends upon "claim 16". *For the purposes of this office action, claim 16 is understood to depend upon claim 15.* Appropriate correction is required.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

5. Claims 1-4, 11-16, 31-38, 40 & 41 are rejected under 35 U.S.C. 102(a) as being anticipated by "Design of A High-Performance ATM Firewall" by Xu.

Regarding claim 1, Xu teaches receiving a communication packet from the external computing system/WAN over the network (page 272 §2.1), the packet having at least a first portion/header and an end portion/last cell, and transmitting/passing the packet to the internal computing system/LAN (page 277 ¶2-4), in parallel with the step of receiving and transmitting the packet, determining characteristics/class of the packet from the first portion/header (page 272 §2.1, page 277 ¶3), in parallel with the step of receiving and transmitting the packet, performing a plurality of checks/TCP/IP rules on the packet (page 272 ¶1, page 275 ¶1), wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks (page 280 ¶1-3 & page 287 ¶1), in parallel with the step of receiving an transmitting the packet, determining if the packet should be a valid/safe packet or an invalid/unsafe packet based on the plurality of checks/rules (pages 275-278 §2.2.3), and after receiving the end portion/last cell of the packet, selectively altering/passing or generating randomly the end portion of the packet based on whether the packet has been determined to be a valid/safe packet or an invalid/unsafe packet, wherein the packet is selectively altered/generated randomly to be invalid/unsafe if it was determined that the packet should be an invalid/unsafe packet (page 277 ¶2).

Regarding claim 2, Xu discloses the packet being analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system/LAN (page 277 ¶2-3).

Regarding claim 3, Xu discloses examining the packet before the last cell has arrived (page 277 ¶2-3)

Regarding claim 4, Xu discloses determining a packet invalid/unsafe if it is determined that the packet is harmful/dangerous (page 272 §2.1 & page 278 ¶2).

Regarding claim 11, Xu discloses the plurality of checks/rules being performed with a programmable logic device/ATM firewall with cache, wherein logic within the programmable logic device/ATM firewall with cache is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet (page 276 ¶2-3).

Regarding claim 12, Xu discloses a physical interface/input module receiving the packet from the network (page 284 §4.2) wherein the packet is coupled to the programmable logic device/ATM firewall with cache, wherein the packet is coupled from the programmable logic device to a second physical interface/output module (page 286 §4.3) for transmission to the internal computing system/LAN (page 282 Fig. 2 & page 283 §4.1 & Fig. 3).

Regarding claim 13, Xu discloses the programmable logic device/ATM firewall with cache performing a plurality of checks while the packet is being coupled from the first physical interface/input module to the second physical interface/output module (pages 284-286 & page 277 ¶2-4).

Regarding claims 14 & 15, Xu discloses filtering based on port numbers (page 275 ¶1).

Regarding claim 16, Xu discloses filtering based on IP addresses (source and destination) (page 275 ¶1).

Regarding claim 31, Xu discloses a first interface circuit/input module for coupling data to and from an external network/WAN (page 282 Fig. 2 & page 284 §4.2), a second interface circuit/output module (page 286 §4.3 & page 283 Fig. 3) for coupling data to and from an internal network/LAN (page 282 Fig. 2 & page 283 §4.1), a programmable logic device/ATM firewall with cache coupled between the first interface circuit/input module and the second interface circuit/output module (page 282 Fig. 2 & page 283 Fig. 3), wherein as a packet is being received and transmitted between the first and second interface circuits (page 282 §2.1), the packet is simultaneously subjected to a plurality of filtering criteria/TCP/IP rules (page 272 ¶1 & page 275-278 §2.2.3) by the programmable logic device/ATM firewall with cache, wherein an end portion/last cell of the packet is selectively altered/passed or generated randomly by the programmable logic device based on the filtering criteria/rules (page 277 ¶2).

Regarding claim 32, Xu discloses the filtering criteria determining whether the packet is to be a valid/safe packet or an invalid/unsafe packet, wherein the packet is selectively altered/generated randomly to be invalid/unsafe if it was determined that the packet should be an invalid/unsafe packet (page 277 ¶2).

Regarding claim 33, Xu discloses determining characteristics/class (page 272 §2.1, page 277 ¶3), of a packet and a filter portion/call-screening service that subjects the packet to a plurality of checks/TCP/IP rules on the packet (page 272 ¶1, page 273 §2.2.1 & page 275 ¶1), while the packet is being received and transmitted between the first and second interface circuits (page 277 ¶2-3).

Regarding claim 34, Xu discloses a stateful filter portion/packet-filter (page 272 §2.1, page 273 §2.2.1, page 285 ¶2 & Fig. 5) and a non-stateful filter portion/traffic-monitor (page 272 §2.1, page 273 §2.2.1 & page 282 Fig. 2).

Regarding claim 35 & 36, Xu discloses the stateful filter portion/packet-filter subjecting the packet to one or more stateful filtering criterion/decision on current packet (page 285 ¶2) while the non-stateful filter portion/rules (page 275 ¶1) subjecting the packet to one or more non-stateful filtering criterion (page 273 §2.2.1, page 280 ¶1 & page 285 ¶2).

Regarding claim 37, Xu discloses a result aggregator logic/output module that receives one ore more signals/decision from the stateful filter portion and the non-stateful filter portion (page 292 ¶1), wherein based on the received signals/decision the result aggregator logic/OM controls whether the packet is selectively altered to be invalid/dropped (page 277 ¶2 & page 292 ¶1).

Regarding claim 38, Xu discloses the result aggregator logic/OM receiving a completion signal/decision that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria (page 292 ¶3).

Regarding claim 40, Xu discloses the packet being subjected to the plurality of filtering criteria/rules (page 273 §2.2.1) in parallel with the packet being received and transmitted between the first and second interface circuits/modules (page 280 ¶1-3 & page 287 ¶1), wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received (page 277 ¶2-4).

Regarding claim 41, Xu discloses the packet being subjected to the plurality of filtering criteria in real time (page 277 ¶2-3) with the packet being received and transmitted between the first and second interface circuits/modules (page 283 Fig. 3).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 30, 44 & 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu.

Regarding claim 44, Xu lacks basing a user-controlled switch's state (effectively enabling/disabling a predetermined portion of the filtering criteria/rules) on whether a computer coupled to the internal network is controlled to operate in a client mode or a server mode. However, official notice is hereby taken that it is known in the network firewall art/network security art that a client/workstation requires different traffic needs (open ports, bandwidth, limitations on number of connections) than does a server. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to base a user-controlled switch's state on whether a computer coupled to the internal network is operating as a client or server. One of ordinary skill in the art would have been motivated to perform such a modification, as it was known in the art to do so.

Regarding claims 30 & 60, Xu lacks a speaker to provide feedback. However, official notice is hereby taken that it was known in the art, as the time the invention was made, to provide a speaker, such as a PC main board speaker, to provide audio feedback (for example on errors). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a speaker in Xu's system to provide feedback. One of ordinary skill in the art would have been motivated to perform such a modification as it was known in the art to do so.

8. Claims 5-8, 10, 17-19, 23-27, 29, 42, 43, 45, 46, 47-49, 53-57, 59, 61-63 & 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claims 1 & 31 above, in view of "PacketShaper 4000 Getting Started Version 4.0" by Packeteer.

Regarding claims 5-8, 10, 42, 43, 45, 61-63 & 66, Xu discloses a firewall system and lacks detailed physical description of the device(s), and hence lacks a physical switch affecting the operation of the firewall. However, Packeteer teaches that it is known to include a power switch to enable/disable function of a device, such as an on/off switch (page 7). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include an on/off toggle switch, thereby affecting the checks based on the state of the switch, affecting the configuration of the checking circuit (on/off), enabling/disabling the checks (on/off). The plurality of checks would selectively perform based on the state an on/off switch. An on/off switch would also control the configuration (on/off). One of ordinary skill in the art would have been motivated to perform such a modification, as it was well known in the art to do so, as taught by Packeteer (page 7).

Regarding claims 23, 24, 46, 53 & 54, Xu discloses a firewall system, as modified above, but lacks detailed physical description of the device(s), and hence lacks a reset switch. However, Packeteer teaches that it is known to include a power switch/reset switch to enable/disable/reset function of a device, such as an on/off switch (page 7). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a physical reset switch/power switch to reset the device described by Xu. One of ordinary skill in the art would have been motivated to perform such a modification, as it was well known in the art to do so, as taught by Packeteer (page 7).

Regarding claims 17-19, 25, 26, 29, 47-49, 55, 56 & 59, Xu discloses a system, as modified above, but lacks visual feedback that the system is operational, the system is subject to filtering criteria, a light source indicative of the operating status having a first color or second color depending on the status and lacks an LED. However, Packeteer teaches that it is known in the art to provide a "status LED", being green or amber in color depending on whether shaping (filtering) is on/operational (page 41) on a hardware packet-shaper/packet-filter (page 1). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a status LED in Xu's system. One of ordinary skill in the art would have been motivated to perform such a modification to convey status information, as was known in the art, as taught by Packeteer (pages 1 & 41).

Regarding claims 27 & 57, Xu discloses a system, as modified above, but lacks a light source that is selectively controlled to blink depending on the operating status. However, Packeteer teaches that it is known to include "network LEDs" to that flicker/blink when transmission or receiving activity occurs (page 41) in a hardware packet-shaper/packet-filter (page 1). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include network LEDs in Xu's system. One of ordinary skill in the art would have been motivated to perform such a modification to convey activity information, as was known in the art, as taught by Packeteer (pages 1 & 41).

9. Claims 20-22 & 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu in view of Packeteer, as applied to claims 18 & 47 above, in further view of "BlackICE Pro User's Guide Version 2.0" by Network Ice Corporation (NIC). Xu discloses a system, as modified above, but lacks audio or visual feedback when the system has rejected one or more packets, when it is suspected to be under attack, or the severity of the attack. However, NIC teaches that to make users aware of attacks and spot trends and patterns of attacks, it is useful to provide a list of possible attacks on the system (page 3 Fig. 3) and indicating the severity (page 21). Further, when a critical or serious event occur, they can cause the blocking of addresses and ports/rejection of packets, and indicate this to the user (page 21 & page 37). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use visual indicators to indicate when the system has rejected packets and when the system is

under attack and to indicate the severity of an attack. One of ordinary skill in the art would have been motivated to perform such a modification to make users aware of attacks and to spot trends, as taught by NIC (pages 1, 3, 21 & 37).

10. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claim 7 above, in view of U.S. Patent 6,052,788 to Wesinger, Jr. et al. (Wesinger). Xu discloses a system, as modified above to include a user-controlled switch such as a power switch, but lacks the circuit being configured or reconfigured based on commands from the internal computing system/LAN. However, Wesinger that configuration of firewalls may be easily accomplished by running a "configurator" which provides a Web-based front-end for editing configuration files, preferably from a secured client (col. 9 lines 31-46). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to change the firewall configuration based on commands from the internal computing system/LAN/secure client (through a Web-browser interface). One of ordinary skill in the art would have been motivated to perform such a modification to easily accomplish firewall configuration, as taught by Wesinger (col. 9 lines 31-46).

11. Claims 28 & 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu in view of Packeteer, as applied to claims27 & 57 above, in further view of "BlackICE Pro User's Guide Version 2.0" by Network Ice Corporation (NIC) in further view of U.S. Patent 6,133,844 to Ahne et al. (Ahne). Xu discloses a system, as modified above, but lacks a light blinking at a rate indicative of a severity level of an attack. Packeteer teaches blinking LEDs indicating traffic

activity (pages 1 & 41). NIC teaches indicating a severity level of an attack to a user (pages 1, 3, 21 & 37). Ahne teaches that on a printing device, an LED's blink rate, *inter alia*, can be altered and the LEDs can be used to convey the operating status of the device (col. 7 lines 22-52 & col. 8 lines 20-37). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the blink rate of a light, as taught by Ahne, on Xu's firewall system, as suggested by Packeteer, to indicate the severity level of an attack, as taught by NIC. One of ordinary skill in the art would have been motivated to perform such a modification to convey operating status to a user, as taught by Ahne (col. 7 lines 22-52 & col. 8 lines 20-37).

12. Claims 64 & 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Xu, as applied to claim 61 above, in view of U.S. Patent 5,905,859 to Holloway et al. (Holloway). Xu discloses user specified criteria/specifying or updating rules via firewall management service (page 281 §2.2.6), but lacks details about the specific hardware involved and therefore, lacks the configuration data transferred from configuration software via a cable attachment. However, Holloway teaches that it is common in the art of managing network devices to supply an RS232 serial port connection to change configuration parameters from a local console (col. 7 lines 11-32). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to transfer configuration parameters via a cable attachment/RS232. One of ordinary skill in the art would have been motivated to perform such a modification to enable a local console to change configuration parameters, as is known in the art to do, as taught by Holloway (col. 7 lines 11-32).

Allowable Subject Matter

13. Claim 39 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

14. The following is a statement of reasons for the indication of allowable subject matter:

Regarding claim 39, the prior art relied upon fails to teach or suggest invalidating a packet if the decision/result is not received by the time the end portion/last cell is received.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. IBM Technical Disclosure Bulletins NN8606320 (1986), NN950431 (1995), NA81123528 (1981), NN9704141 (1997), NN9512419 (1995), NN9502341 (1995), NN9308183 (1993), NN8606254 (1986), NN83102393 (1983) and 3com SuperStack 3 Firewall data sheet were cited for relevance in the various applications of LEDs acting as indicators, through color, blink rate, etc.

b. "Design of a High-Performance ATM Firewall", 1998 ACM was cited as and older, less refined version of the primary reference to Xu relied upon.

c. "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching" was cited for teaching intrusion alarms for network security.

d. "Norton Personal Firewall 2000 User's Guide" was cited for relevance in software firewall methods of displaying operating information to a user.

e. "A High Speed Firewall Architecture for ATM/OC-3c" was cited for teaching bit-parallelism in firewall rule/policy matching.

f. U.S. Patent 6,092,108 was cited for relevance in packet fragmentation in packet filtering environments.

g. U.S. Patent 6,335,935 was cited for teaching the dropping of packets when queues are full.

h. U.S. Patent 6,691,168 was cited for teaching parallel rule processing to speed up network filtering.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Simitoski whose telephone number is (703) 305-8191. The examiner can normally be reached on Monday - Thursday, 6:45 a.m. - 4:15 p.m.. The examiner can also be reached on alternate Fridays from 6:45 a.m. – 3:15 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached on (703) 308-4789.

Any response to this action should be mailed to:
Commissioner of Patents and Trademarks
Washington, DC 20231

Or faxed to:
(703) 746-7239 (for formal communications intended for entry)

Or:
(703) 746-7240 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA 22202, Fourth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9000.

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

MJS
February 20, 2004

Notice of References Cited	Application/Control No. 09/611,775	Applicant(s)/Patent Under Reexamination KRUMEL, ANDREW K.
	Examiner Michael J Simitoski	Art Unit 2134 — Page 1 of 3

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,905,859 A	05-1999	Holloway et al.	713/201
	B	US-6,052,788 A	04-2000	Wesinger et al.	713/201
	C	US-6,092,108 A	07-2000	DiPlacido et al.	709/224
	D	US-6,133,844 A	10-2000	Ahne et al.	340/815.45
	E	US-6,335,935 B2	01-2002	Kadambi et al.	370/396
	F	US-6,691,168 B1	02-2004	Bal et al.	709/238
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	3com. "SuperStack 3 Firewall", 2000 3com.
	V	Hughes, James. "A High Speed Firewall Architecture for ATM/OC-3c", February 1996.
	W	IBM Technical Disclosure Bulletins NN8606320 (1986), NN950431 (1995), NA81123528 (1981), NN9704141 (1997), NN9512419 (1995), NN9502341 (1995), NN9308183 (1993), NN8606254 (1986), NN83102393 (1983).
	X	Lakshman, T.V. "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching", 1998 ACM, pp. 203-214.

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Notice of References Cited	Application/Control No. 09/611,775	Applicant(s)/Patent Under Reexamination KRUMEL, ANDREW K.	
	Examiner Michael J Simitoski	Art Unit 2134	Page 2 of 3

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Network ICE Corporation. "Black ICE Pro User's Guide Version 2.0", June 2000 (archive.org).
	V	Packeteer, Inc. "PacketShaper 4000 Getting Started Version 4.0", March 1999.
	W	Symantec, Inc. "Norton Personal Firewall 2000 User's Guide Version 2.0", June 2000 (archive.org).
	X	Xu, Jun and Mukesh Singhal. "Design of a High-Performance ATM Firewall", 1999 ACM.

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Notice of References Cited	Application/Control No. 09/611,775	Applicant(s)/Patent Under Reexamination KRUMEL, ANDREW K.	
	Examiner Michael J Simitoski	Art Unit 2134	Page 3 of 3

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Xu, Jun and Mukesh Singhal. "Design of a High-Performance ATM Firewall", 1998 ACM, pp. 93-102.
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Sheet 1 of 1

Form PTO-1449 (REV. 7-92) U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office	Attorney's Docket Number 802-001	Serial No. 09/611,775	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary)	Applicant(s): Krumel		
	Filing Date: July 7, 2000	Group Art Unit	

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	DOCUMENT NUMBER							DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
MGS	5	7	4	0	3	7	5	4/14/98	Dunne et al.	395	200.68	
MGS	5	8	3	5	7	2	6	11/10/98	Shwed et al.	395	200.59	
MGS	5	8	8	4	0	2	5	3/16/99	Baehr et al.	395	187.01	
MGS	5	9	6	8	1	7	6	10/19/99	Nessett et al.	713	201	
MGS	6	0	0	3	1	3	3	12/14/99	Moughanni et al.	713	200	
MGS	6	0	0	9	4	7	5	12/28/99	Shrader	709	249	

FOREIGN PATENT DOCUMENTS

	DOCUMENT NUMBER							DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
MGS	WO	96/	3	4	4	7	9	10/31/96	PCT			YES	NO
MGS	WO	99/	4	8	3	0	3	9/23/99	PCT				
MGS.	WO	00/	0	2	1	1	4	1/13/00	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

L Number	Hits	Search Text	DB	Time stamp
1	52	@ad<20000705 and ((hub router firewall (packet adj filter$3)) same (alarm alert intrud$3 intrusion attack) same (audio speaker beep$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 09:01
2	8	@ad<20000705 and ((hub router firewall (packet adj filter$3)) same (alarm alert intrud$3 intrusion attack) same (speaker beep$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 09:02
3	330	@ad<20000705 and ((hub router firewall (packet adj filter$3)) same (speaker beep$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 09:03
4	120	@ad<20000705 and ((router firewall (packet adj filter$3)) same (speaker beep$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 09:03
-	0	@ad<20000707 and (firewall and ("allow all"))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:18
-	448	@ad<20000707 and (firewall and (bypass$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:15
-	12	@ad<20000707 and (firewall and (bypass$3) near (toggle switch))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:17
-	2	@ad<20000707 and ((packet adj filter$3) and (bypass$3) near (toggle switch))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:17
-	0	@ad<20000707 and ((packet adj filter$3) and (bypass$3) near (toggle))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:18
-	0	@ad<20000707 and ((packet adj filter$3) and ("allow all"))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:18
-	1	@ad<20000707 and ((packet adj filter$3) and (uninhibit$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:18
-	46	@ad<20000707 and (((packet adj filter$3) firewall) same (disable))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:22
-	0	@ad<20000707 and (((packet adj filter$3) firewall) same ((block pass) near all))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/02/19 10:46
-	26	("LED" light indicat$3) near (alert alarm attack intrusion intruder) near (severity)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:41
-	14	(("LED" light indicat$3) near (alert alarm attack intrusion intruder) near (severity)) and @ad<20000705	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 16:14

-	214	@ad<20000705 and ((alert alarm attack intrusion intruder) near (severity))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:41
-	112	@ad<20000705 and ((alert attack intrusion intruder) near (severity))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:41
-	87	@ad<20000705 and ((attack intrusion intruder) near (severity))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:41
-	12	@ad<20000705 and (("LED" light indicat$3) same ((attack intrusion intruder) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:46
-	0	@ad<20000705 and ((color) same ((attack intrusion intruder) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:46
-	0	@ad<20000705 and ((color) near ((attack intrusion intruder) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:46
-	0	@ad<20000705 and ((color) near ((flood$3 attack$3 intrusion intrud$3) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:47
-	0	@ad<20000705 and ((color) same((flood$3 attack$3 intrusion intrud$3) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:47
-	0	@ad<20000705 and (("LED") same ((flood$3 attack$3 intrusion intrud$3) near (severity)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 15:47
-	415	@ad<20000705 and (blink$3 near rate)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/19 16:14
-	2	@ad<20000705 and (blink$3 near rate) same (attack intrusion intrud$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 07:12
-	0	@ad<20000705 and (blink$3 near rate) same (malicious)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 07:13
-	7	@ad<20000705 and (blink$3 near rate) same (danger$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 07:13
-	0	@ad<20000705 and (blink$3 near rate) same (collision)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 07:14
-	1	@ad<20000705 and (blink$3 near rate) same (router firewall hub)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 07:16
-	2	@ad<20000705 and ((open near port) same (router firewall (packet adj filter$3)) same (client and server))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/20 08:59

Non-patent literature search information - 2/20/04

09611775
Michael J. Simitoski
Michael.Simitoski@uspto.gov
(703) 305-8191

Google

PLD-based firewall
packetshaper user guide
"zone alarm" 2.1 download
black ice firewall

ACM

+packet +rules +parallel +invalidate filter firewall

IEEE

(led <or> light) <and> (intrusion <or> intruder <or> attack)

Applications/Patents from Inventor Search

none applicable

Google™ Advanced Search Preferences Language Tools Search Tips

firewall LED attack [Google Search]

Web · Images · Groups · Directory · News ·

Searched the web for **firewall LED attack**.

Results **11** - **20** of about **35,600**. Search took **0.14** seconds.

[PDF] Network Secure VPN **Firewall** for Departmental
File Format: PDF/Adobe Acrobat - View as HTML
... 1 External, 1 DMZ) **LED** Indicators - Power ... Win nuke **attack** - Port Scan **attack** - Ping of ... Software Specifications Security Feature: - **Firewall**: Stateful Packet ...
www.dlink.co.nz/products/routers/dfl1000/DFL1000.pdf - Similar pages

ZyWALL1 - VPN **Firewall** for telecommuters and SOHO
... The popularity of the Internet has **led** to one ... The ZyWALL 1 provides powerful **firewall** capabilities, including ... of Service (DoS) prevention and **Attack** Alert. ...
www.murkworks.com/Products/ZyWALL/ZyWALL1 - 20k - Cached - Similar pages

D-Link DFL-500 Network Secure VPN **Firewall** for SOHO
... **LED** indicators, -Power -Status -Interface link and activity. **Firewall**, Stateful Packet Inspection (SPI) to Prevent ... of death", IP spoofing, land **attack**, tear drop ...
www.value.co.th/products/dlink/DFL500.asp - 10k - Cached - Similar pages

SecurityFocus HOME Advisories: Kerio Personal **Firewall** Replay ...
... BID 7179] A replay **attack** is possible ... from the administrator's workstation 'C'. This **led** us to ... to reissue the administration commands to the personal **firewall**. ...
www.securityfocus.com/advisories/5330 - 27k - Cached - Similar pages

Course 2771—Three days—Instructor-**led**
... CSPF) is a three day, instructor-**led**, lab-intensive ... AAA Configuration on the Cisco Secure PIX **Firewall**; ... Filtering; Advance Protocol Handling and **Attack** Guards on ...
www.xincon.com/hhc/lan_wan/Cisco_Pix_Firewalls.html - 4k - Cached - Similar pages

Cisco Secure PIX Firewalls :: $24.50 at InformIT.com
... deploying, and managing PIX **Firewall** protected networks. ... 520, 525, and 535, including **LED** information and ... Proxy, Advanced Protocol Handling, **Attack** Guards, and ...
www.informit.com/isapi/product_id~%7BDD5D9082-5332-44DA-9AC7-E7D9D91035C6%7D/ content/index.asp - 25k - Cached - Similar pages

Ravenholm Computing - Cisco Press Cisco Secure PIX Firewalls
... authorize users and services Understand **attack** guards such ... Based on the official instructor-**led** training course (Cisco Secure PIX **Firewall** Advanced-CSPFA ...
www.ravenholm.fi/tuotteet/11_2_14_9995.htm - 25k - Cached - Similar pages

[PDF] Low-cost appliances challenge pricey security platforms in our ...
File Format: PDF/Adobe Acrobat - View as HTML
... Thanks to GbE capability, it easily **led** the field in our performance tests, including performance when under **attack**. 9 8 6 9 7 6 Ingate **Firewall** 1400 Ingate ...
www.astaro.com/data/news/pdf/gb_46.pdf - Similar pages

Cisco Secure PIX **Firewall** Advanced (CSPFA)
... of threats The three primary methods of **attack** The Security ... the Cisco PIX **Firewall** PIX **Firewall** 501, 506 ... and 535 controls, connectors, and **LED's** Proper location ...
www.slsuk.com/CSPFA.htm - 17k - Cached - Similar pages

viren 5 passwort keeper Cisco hack router **attack** jet trinity warez ...
... passwort keeper Cisco hack router **attack** jet trinity ... cd **led** crack pontis **led** crack pontis cd ... software crackz elektrik software viren **firewall** virenscanner w32 ...
www.deniseward-brown.com/firewall/melissa.more.htm - Similar pages
[More results from www.deniseward-brown.com]

◀ Gooooooooooogle ▶

Result Page: **Previous** 1 **2** 3 4 5 6 7 8 9 10 11 **Next**

firewall LED attack [Google Search] Search within results

Google Home - Advertise with Us - Business Solutions - Services & Tools - Jobs, Press, & Help

Google™ Advanced Search Preferences Language Tools Search Tips

LED attack site:www.cisco.com [Google Search]

Web · Images · Groups · Directory · News ·

Searched pages from **www.cisco.com** for **LED attack**. Results **1** - **10** of about **430**. Search took **0.21** seconds.

[PDF] Cisco and Exodus: Protecting Customers from Cyber **Attack**
File Format: PDF/Adobe Acrobat - View as HTML
... The Exodus Cyber **Attack** Tiger Team ™ (CATT), **led** by Senior Director Charles Neal,
is a team of security experts who are responsible for investigating attacks ...
www.cisco.com/warp/public/779/servpro/ solutions/security/exod_cp.pdf - Similar pages

Data Encryption Service Adapter
... To eliminate the intruder-in-the-middle **attack**, every encryption device in the ... The
ESA contains the ENABLED **LED**, standard on all service adapters, and four ...
www.cisco.com/univercd/cc/td/doc/product/lan/ cat5000/cnfg_nts/rsm/rsm_pa/4801encr.htm - 101k - Cached - Similar pages

The Science of IDS **Attack** Identification-Cisco Intrusion Detection ...
... The Science of Intrusion Detection System **Attack** Identification. ... This has **led** to many
confusing claims by vendors in the IDS market about the best methodologies ...
www.cisco.com/en/US/products/sw/secursw/ps2113/ products_white_paper09186a0080092334.shtml - 45k - Cached - Similar pages

[PDF] The Science of Intrusion Detection System **Attack** Identification
File Format: PDF/Adobe Acrobat - View as HTML
... 1 of 5 White Paper The Science of Intrusion Detection System **Attack** Identification
Introduction A ... This has **led** to many confusing claims by vendors in the IDS ...
www.cisco.com/warp/public/cc/pd/sqsw/ sqidsz/prodlit/idssa_wp.pdf - Similar pages

Overview-Cisco Catalyst 6500 Series Switches - Cisco Systems
... After the IDS module detects an **attack**, it responds by generating an alarm. ... The
IDS module (see Figure 1-2) has a status **LED** and a Shutdown button. ...
www.cisco.com/.../ switches/ps708/products_installation_and_configuration_guide_chapter09186a0080f7876c.html - 38k - Cached - Similar pages

Data Enryption Service Adapter Installation and Configuration ...
... This eliminates the intruder-in-the-middle **attack**. ... ESA LEDs. The ESA contains
the enabled **LED**, standard on all service adapters, and a four status LEDs. ...
www.cisco.com/ en/US/products/hw/modules/ps2957/prod_module_installation_guide09186a00800f2681.html - 101k - Cached - Similar pages

[PDF] Cisco - PSIRT Reference Information
File Format: PDF/Adobe Acrobat - View as HTML
... that this program has caused among Cisco customers has **led** us to ... Strategies for **Attack**

defense, tracking or mitigation Characterizing and Tracing Packet Floods ...
www.cisco.com/warp/public/707/ref.pdf - Similar pages

Security Reference Information-Cisco 10000 Series Routers - Cisco ...
... The unexpected concern that this program has caused among Cisco customers has **led** us to suspect that many ... Strategies for **Attack** defense, tracking or mitigation. ...
www.cisco.com/en/US/products/hw/routers/ps133/ products_tech_note09186a0080143d1b.shtml - 42k - Feb 16, 2004 - Cached - Similar pages

The Twilight Zone-Packet Vol. 13, No. 1, First Quarter 2001 ...
... Ziese, **led** the US air force team that brought down the infamous cracker, DataStream Cowboy. Keywords: Cracker, DataStream Cowboy, hacker, password **attack**, ...
www.cisco.com/en/US/about/ac123/ac114/ac173/ac165/ about_cisco_packet_netizen09186a00801143da.html - 40k - Cached - Similar pages

[PDF] Overview
File Format: PDF/Adobe Acrobat - View as HTML
... After the IDS module detects an **attack**, it responds by generating an alarm ... Panel Description The IDS module (see Figure 1-2 on page 1-3) has a status **LED** and a ...
www.cisco.com/univercd/cc/td/doc/product/lan/ cat6000/cfgnotes/idsm_4_0/smchap1.pdf - Similar pages

Gooooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

LED attack site:www.cisco.com [Google Search] Search within results

Dissatisfied with your search results? Help us improve.

Google Home - Advertise with Us - Business Solutions - Services & Tools - Jobs, Press, & Help

L Number	Hits	Search Text	DB	Time stamp
-	6	(("5740375") or ("5835726") or ("5884025") or ("5968176") or ("6003133") or ("6009475")).PN.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/18 07:19
-	0	("US20020083331").PN.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 12:53
-	1	("20020083331").PN.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 12:54
-	1	("20020080784").PN.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:04
-	12	stateful near (filter$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:04
-	17	(stateful near2 filter$3) and @ad<20000707	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:27
-	0	("PLD" near firewall) and @ad<20000707	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:28
-	0	("PLD" same firewall) and @ad<20000707	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:39
-	2	("6182288" "5960177").pn.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:36
-	0	("61822??").pn. and filter	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:36
-	0	("618????").pn. and filter	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:36
-	1860	(618????).pn. and filter	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:36
-	0	(618228?).pn. and filter	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:36
-	15	(61822??).pn. and filter	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:38
-	1136	@ad<20000707 and 713/201.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:39
-	1854	@ad<20000707 and 709/229,249,225.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:39
-	5113	@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:39

-		20	(@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:39
-		129	(((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:45
-		63	(((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:45
-		32	(((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time) and rule	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 13:50
-		32	((((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)) and (filter$3 near2 packet) and (parallel) and (real adj time) and rule) not ((("5740375") or ("5835726") or ("5884025") or ("5968176") or ("6003133") or ("6009475")).PN.)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:41
-		251	(zero adj (copy$3 copied))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:45
-		1	((zero adj (copy$3 copied))) and @ad<20000707 and (filter$3 near2 packet)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:43
-		38	((zero adj (copy$3 copied))) and @ad<20000707 and (filter$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:43
-		38	(((zero adj (copy$3 copied))) and @ad<20000707 and (filter$3)) and (zero adj (copy$3 copied))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:45
-		8	(((zero adj (copy$3 copied))) and @ad<20000707 and (filter$3)) and ((zero adj (copy$3 copied)) same buffer$)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/12 14:46
-		738	(toggle near switch) same (enable disable)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 10:32
-		7	(toggle near switch) same (enable disable) and (firewall (packet adj filter$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 10:49
-		248	(firewall) and (programmable adj logic)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 10:50

-		72	@ad<20000707 and (firewall) and (programmable adj logic)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 10:53
-		1	@ad<20000707 and ((firewall) (packet adj filter$3)).ti. and (programmable adj logic)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 11:07
-		6844	((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 11:12
-		19	(((@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.) and (709/$.ccls. and 713/$.ccls.)) (@ad<20000707 and 709/229,249,225.ccls.) (@ad<20000707 and 370/356,389,392,395.21,395.32,401.ccls.)) and (program$7 near (logic device)) same (check rule filter)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 12:39
-		138	(firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:03
-	title scanned	79	((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<20000707	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:04
-		48	(((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<20000707) not bowman.in.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:05
-		0	((((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<20000707) not bowman.in.) and (toggle)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:05
-		0	((((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<20000707) not bowman.in.) and (toggl$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:05
-		32	((((firewall (packet adj filter$3)) and ((enabl$3 disabl$3) near filter$3)) and @ad<20000707) not bowman.in.) and (switch button)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:22
-		50	toggl$3 near (filter$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 13:23
-		6	toggl$3 same ((disabl$3 enabl$3) near (filter$3))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 16:01
-		1030	(("LED" light) near (blink$3 glow$3)) same (router firewall hub switch)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 16:02
-		10	(("LED" light) near (blink$3 glow$3)) same (router firewall hub switch) same (attack$3 intru$5)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 16:03
-		7	(("LED" light) near (blink$3 glow$3)) same (router firewall hub switch) same (collision)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/17 16:04
-		1	((light "LED" indicator) near2 (alarm attack)) same (blink$3)	IBM_TDB	2004/02/18 07:20
-		25	(light "LED" indicator) near2 (blink$3)	IBM_TDB	2004/02/18 07:21
-		0	((light "LED" indicator) near2 (blink$3)) and @ad<20000707	IBM_TDB	2004/02/18 07:21

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+packet +rules +parallel +invalidate +real-time filter firewall SEARCH

US Patent & Trademark Office

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published since January 1947 and Published before July 2000
Terms used packet rules parallel invalidate real time filter firewall

Found 54 of 96,905

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 54

Result page: **1** 2 3 next

Relevance scale

1 UNICORN: misuse detection for UNICOS
Gary G. Christoph, Kathleen A. Jackson, Michael C. Neuman, Christine L. B. Siciliano, Dennis D. Simmonds, Cathy A. Stallings, Joseph L. Thompson
December 1995 **Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: html(43.88 KB) Additional Information: full citation, references, citings, index terms

2 Application access control at network level
Refik Molva, Erich Rütsche
November 1994 **Proceedings of the 2nd ACM Conference on Computer and communications security**
Full text available: pdf(958.92 KB) Additional Information: full citation, abstract, references, index terms

This paper describes an access control mechanism that enforces at the network level an access control decision that is taken at the application level. The mechanism is based on the pre-computation of encrypted counters called tickets. An access enforcement device verifies the existence of a valid ticket in each packet that is subject to access control and kills unauthorized packets. Tickets are not computed as a function of the user data. Due to the timing constraints of shared media LANs t ...

3 Secure and mobile networking
Vipul Gupta, Gabriel Montenegro
December 1998 **Mobile Networks and Applications**, Volume 3 Issue 4
Full text available: pdf(223.39 KB) Additional Information: full citation, abstract, references, citings, index terms

The IETF Mobile IP protocol is a significant step towards enabling nomadic Internet users. It allows a mobile node to maintain and use the same IP address even as it changes its point of attachment to the Internet. Mobility implies higher security risks than static operation. Portable devices may be stolen or their traffic may, at times, pass through links with questionable security characteristics. Most commercial organizations use some combination of source-filtering routers, sophisticate ...

4 Query evaluation techniques for large databases
Goetz Graefe
June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2
Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

5 Mining in a data-flow environment: experience in network intrusion detection
Wenke Lee, Salvatore J. Stolfo, Kui W. Mok
August 1999 **Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining**
Full text available: Additional Information:

pdf(1.26 MB) full citation, references, citings, index terms

6 Parallel and distributed incremental attribute evaluation algorithms for multiuser software development environments
Gail E. Kaiser, Simon M. Kaplan
January 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 2 Issue 1
Full text available: pdf(3.09 MB) Additional Information: full citation, abstract, references, citings, index terms

The problem of change propagation in multiuser software development environments distributed across a local-area network is addressed. The program is modeled as an attributed parse tree segmented among multiple user processes and changes are modeled as subtree replacements requested asynchronously by individual users. Change propagation is then implemented using decentralized incremental evaluation of an attribute grammar that defines the static semantic properties of the p ...

Keywords: attribute grammar, change propagation, distributed, incremental algorithm, parallel, reliability

7 A software synthesis tool for distributed embedded system design
D.-I. Kang, R. Gerber, L. Golubchik, J. K. Hollingsworth, M. Saksena
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 workshop on Languages, compilers, and tools for embedded systems**, Volume 34 Issue 7
Full text available: pdf(1.64 MB) Additional Information: full citation, abstract, references, index terms

We present a design tool for automated synthesis of embedded systems on distributed COTS-based platforms. Our synthesis tool consists of (1) a graphical user interface for input of software layouts, which maps tasks to resources and (2) a constraints solving engine, which allocates local resources to tasks, all with the goal of meeting specified performance criteria. Our tool differs from previous work in that it allows (a) use of stochastic (rather than worst-case) models of resource usage and ...

8 Session summaries from the 17th symposium on operating systems principle (SOSP'99)
Jay Lepreau, Eric Eide
April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2
Full text available: pdf(3.15 MB) Additional Information: full citation, index terms

9 Applying an information gathering architecture to Netfind: a white pages tool for a changing and growing Internet
Michael F. Schwartz, Calton Pu
October 1994 **IEEE/ACM Transactions on Networking (TON)**, Volume 2 Issue 5
Full text available: pdf(1.71 MB) Additional Information: full citation, references, citings, index terms, review

10 Correct memory operation of cache-based multiprocessors
C. Scheurich, M. Dubois
June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**
Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper shows that cache coherence protocols can implement indivisible synchronization primitives reliably and can also enforce sequential consistency. Sequential consistency provides a commonly accepted model of behavior of multiprocessors. We derive a simple set of conditions needed to enforce sequential consistency in multiprocessors. These conditions are easily applied to prove the correctness of existing cache coherence protocols that rely on one or multiple broadcast buses to enfor ...

11 A structural view of the Cedar programming environment
Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4
Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

12

The process group approach to reliable distributed computing
Kenneth P. Birman
December 1993 **Communications of the ACM**, Volume 36 Issue 12
Full text available: pdf(6.00 MB) Additional Information: full citation, references, citings, index terms

Keywords: fault-tolerant process groups, message ordering, multicast communication

13 Using name-based mappings to increase hit rates
David G. Thaler, Chinya V. Ravishankar
February 1998 **IEEE/ACM Transactions on Networking (TON)**, Volume 6 Issue 1
Full text available: pdf(408.96 KB) Additional Information: full citation, references, citings, index terms

Keywords: World Wide Web, caching, client-server systems, computer networks, distributed agreement, multicast routing, proxies

14 Implementation of Argus
B. Liskov, D. Curtis, P. Johnson, R. Scheifer
November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles**, Volume 21 Issue 5
Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

Argus is a programming language and system developed to support the construction and execution of distributed programs. This paper describes the implementation of Argus, with particular emphasis on the way we implement atomic actions, because this is where Argus differs most from other implemented systems. The paper also discusses the performance of Argus. The cost of actions is quite reasonable, indicating that action systems like Argus are practical.

15 A survey of data mining and knowledge discovery software tools
Michael Goebel, Le Gruenwald
June 1999 **ACM SIGKDD Explorations Newsletter**, Volume 1 Issue 1
Full text available: pdf(1.28 MB) Additional Information: full citation, abstract, references

Knowledge discovery in databases is a rapidly growing field, whose development is driven by strong research interests as well as urgent practical, social, and economical needs. While the last few years knowledge discovery tools have been used mainly in research environments, sophisticated software products are now rapidly emerging. In this paper, we provide an overview of common knowledge discovery tasks and approaches to solve these tasks. We propose a feature classification scheme that can be ...

Keywords: data mining, knowledge discovery in databases, surveys

16 Receiver-driven bandwidth adaptation for light-weight sessions
Elan Amir, Steven McCanne, Randy Katz
November 1997 **Proceedings of the fifth ACM international conference on Multimedia**
Full text available: pdf(1.95 MB) Additional Information: full citation, references, citings, index terms

17 Automatic generation of scheduling and communication code in real-time parallel programs
André Bakkers, Johan Sunter, Evert Ploeg
November 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 workshop on Languages, compilers, & tools for real-time systems**, Volume 30 Issue 11
Full text available: pdf(1.45 MB) Additional Information: full citation, abstract, references, index terms

Inter-process communication and scheduling are notorious problem areas in the design of real-time systems. Using CASE tools, the system design phase will in general result in a system description in the form of parallel processes. Manual allocation of these processes to processors may result in error prone and/or slow communication code. Scheduling of the processes, necessary to meet timing constraints, is also a tedious task that takes many iterations. The described design tools result in code ...

18 The Starfire SMP interconnect
Alan Charlesworth, Nicholas Aneshansley, Mark Haakmeester, Dan Drogichen, Gary Gilbert, Ricki Williams, Andrew Phelps
November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(273.52 KB) Additional Information: full citation, abstract, references, citings

The Starfire interconnect extends the envelope of Unix symmetric multiprocessor (SMP) systems in several dimensions. **Interconnect:** an active centerplane with four address routers and a 16x16 data crossbar provides 64 UltraSPARC processors with uniform memory access at a bandwidth of 10,667 MBps. **Flexibility:** Starfire can be dynamically reconfigured into multiple hardware-protected operating system domains. **Robustness:** Failing boards can be hot swapped without interrupting sy ...

Keywords: SMP, UMA, bandwidth, domains, interconnect, latency, partitions

19 Session 1: Applications: Convenient abstractions in stormcast applications
Dag Johansen, Gunnar Hartvigsen
September 1994 **Proceedings of the 6th workshop on ACM SIGOPS European workshop: Matching operating systems to application needs**
Full text available: pdf(686.79 KB) Additional Information: full citation, abstract, references

In this paper we present experience with meteorology applications and appropriate distributed computing abstractions. We focus on the need for co-existence and integration of multiple paradigms in large scale distributed applications, rather than enforcing a favourite paradigm whenever possible.

20 Personal distributed computing: the Alto and Ethernet software
Butler Lampson
January 1986 **Proceedings of the ACM Conference on The history of personal workstations**
Full text available: pdf(3.00 MB) Additional Information: full citation, abstract, references, citings, index terms

The personal distributed computing system based on the Alto and the Ethernet was a major effort to make computers help people to think and communicate. The paper describes the complex and diverse collection of software that was built to pursue this goal, ranging from operating systems, programming environments, and communications software to printing and file servers, user interfaces, and applications such as editors, illustrators, and mail systems.

Results 1 - 20 of 54 Result page: **1** 2 3 next

2785
2131
#3

PTO/SB/122 (10-01)
Approved for use through 10/31/2002. OMB 0651-0035
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

CHANGE OF CORRESPONDENCE ADDRESS

Application

Address to
Assistant Commissioner for Patents
Washington, D.C. 20231

Application Number	09/611,775
Filing Date	July 7, 2000
First Named Inventor	Krumel
Art Unit	2785
Examiner Name	
Attorney Docket Number	802-001

Please change the Correspondence Address for the above-identified application to:

□ Customer Number ______ → *Place Customer Number Bar Code Label here*

Type Customer Number here

OR

☒ Firm *or* Individual Name	Loudermilk & Associates				
Address	P. O. Box 3607				
Address					
City	Los Altos	State	CA	ZIP	94024-0607
Country	U.S.A.				
Telephone	408-342-1866 (unchanged)	Fax	408-342-1868 (unchanged)		

This form cannot be used to change the data associated with a Customer Number. To change the data associated with an existing Customer Number use "Request for Customer Number Data Change" (PTO/SB/124).

I am the :

□ Applicant/Inventor.

229, 247, 246

□ Assignee of record of the entire interest.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

☒ Attorney or Agent of record.

□ Registered practitioner named in the application transmittal letter in an application without an executed oath or declaration. See 37 CFR 1.33(a)(1). Registration Number ____________

Typed or Printed Name	Alan R. Loudermilk, Reg. No. 32,788
Signature	Alan R. Loudermilk
Date	April 18, 2002

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

□ *Total of ____________ forms are submitted.

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

9 #
2/04/01

Attorney Docket No.: 802-001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:)
)
Krumel)
) Art Unit: Unassigned
Serial No.: 09/611,775)
)
Filed: July 7, 2000) Examiner: Unassigned
)
For: Real Time Firewall/Data Protection Systems and Methods)
)

INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

1. Pursuant to 37 C.F.R. 1.97 and 1.98, and in compliance with 37 C.F.R. 1.56, the Office's attention is directed to the patents, publications and other information listed on the attached PTO-1449. A copy of each listed document is enclosed except for: (a) pending applications or (b) those previously cited or submitted to the Office in the following application(s) upon which this application relies for an earlier filing date under 35 U.S.C. 120:

Serial No.: ______________

Filing Date: _____________

Regarding the document(s), publication(s) or other information listed on the attached PTO-1449, Applicant(s) believe(s) the same may qualify as "prior" art to this application and should be treated accordingly, although Applicant(s) reserve(s) the right to contest the prior art status of any document, publication or information cited herein.

2. Regarding each listed document that is not in the English language, an English-language translation accompanies this Statement as indicated on the attached PTO-1449 or a concise explanation of the relevance of the document is set forth in the following documents(s):

(a) ___ Copy of each English language version of a search report indicating the degree of relevance found by the foreign office of each document being submitted from the search report.

(b) ___ Attachment entitled "Concise Explanation of Relevance of Non-English Language Documents."

3. Pursuant to 37 C.F.R. 1.97(b) this Statement is being filed (one must be checked):

(a) ___ Within 3 months of the filing date or date of entry into the National Stage.

(b) X Before the mailing date of a first Office Action on the merits. If this Statement is not filed before the mailing date of a first Office Action on the merits, the required certification is given below or, in the absence thereof, the Office is authorized to charge the required fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251 for consideration of this Statement.

(c) ___ After the period set forth in 37 C.F.R. 1.97(b) but before the mailing date of either a final action or a notice of allowance.

(1) ___ The required certification is given below, or

(2) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(p) for consideration of this Statement, or

(3) ___ Charge the fee set forth in 37 C.F.R. 1.17(p) to Deposit Account No. 50-0251

(d) ___ After the mailing date of either a final action or a notice of allowance, but before payment of the issue fee. Petition hereby is made for consideration of this Statement and the required certification is indicated below.

(1) ___ Enclosed is a check covering the fee set forth in 37 C.F.R. 1.17(i)(1), or

(2) ___ Charge the fee set forth in 37 C.F.R. 1.17(i)(1) to Deposit Account No. 50-0251.

4. Certification (if applicable)

(a) ___ The undersigned hereby certifies that each item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than 3 months prior to the filing of this Statement.

(b) X The undersigned hereby certifies that no item of information contained in this Statement was cited in a communication from a foreign patent office in a counterpart foreign application or, to the undersigned's knowledge after making reasonable inquiry, was known to any individual designated in 37 C.F.R. 1.56(c) more than 3 months prior to the filing of this Statement.

5. The Commissioner is hereby authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175.

Respectfully submitted,

Alan R Loudermilk /kr

Alan R. Loudermilk
Registration No. 32,788
Attorney for Applicant(s)

October 12, 2000
10950 N. Blaney Ave., Suite B
Cupertino, CA 95014
(408) 342-1866

I hereby certify that the foregoing is being deposited with the U.S. Postal Service, postage prepaid, to the Assistant Commissioner for Patents, Washington, DC 20231 this 12th day of October, 2000.

Karen A Rogers

Karen A. Rogers

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} :
H04L 12/56

A1

(11) International Publication Number: **WO 96/34479**

(43) International Publication Date: 31 October 1996 (31.10.96)

(21) International Application Number: PCT/US95/05444

(22) International Filing Date: 24 April 1995 (24.04.95)

(71) Applicant *(for all designated States except US)*: CISCO SYSTEMS, INCORPORATED [US/US]; 1525 O'Brien Drive, Menlo Park, CA 94025 (US).

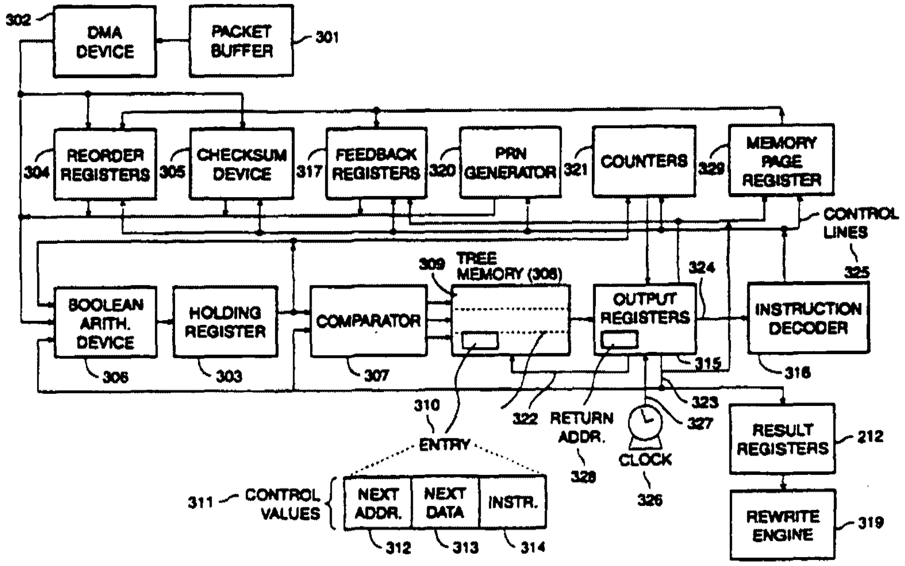
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: WILFORD, Bruce, A. [GB/US]; 935 Eastwood Place, Los Altos, CA 94024 (US). SHERRY, Bruce [US/US]; 15621 North East 164th Street, Woodinville, WA 98072 (US). TSIANG, David [US/US]; 1686 Cak Avenue, Menlo Park, CA 94052 (US). LI, Anthony [US/US]; 1067 Fifth Court, Sunnyvale, CA 94087 (US).

(74) Agents: D'ALESSANDRO, Kenneth et al.; D'Alessandro & Ritchie, P.O. Box 640640, San Jose, CA 95164-640 (US).

(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, MW, SD, SZ, UG), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
With amended claims and statement.

(54) Title: PACKET SWITCHING ENGINE



(57) Abstract

A device for switching packets at high speed. For each packet, the device matches packet data with protocol, to determine how to switch the packet. Matching of data with protocols is highly parallel; the device simultaneously retrieves a data byte, compares a data byte with a protocol byte, tests a comparison result, and executes a processor instruction. A switching engine has a comparator (307) and a decision tree memory (308). The comparator (307) includes three outputs for indicating a comparison result. The tree memory (308) includes three corresponding banks of addressable memory. Each memory location comprises an entry for a next location, an entry for a next protocol byte, and an entry for a processor instruction. A set of protocol byte tests are assembled into the tree memory (308) and a set of routing tables are dynamically generated into the tree memory (308).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia
AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LT	Lithuania
LU	Luxembourg
LV	Larvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

TITLE OF THE INVENTION

Packet Switching Engine

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to packet switching.

2. Description of Related Art

When it is desired to transmit information from one computing device to another, it is known to transmit that information over a network. A network may include a set of computing devices coupled to a communication path, so that each device may communicate with other devices, and a communication protocol and a set of destination addresses, so that each device may recognize communications directed to it. In many networks, each message may be broken into well defined elements, called packets, which may be independently transmitted from a source device to a destination device. Each packet may generally comprise a packet header, with information relating to transmission and routing, and a packet body, with the data to be transmitted.

When it is desired to couple two networks, it is known to provide a switching device which is coupled to both networks, and which may receive packets from one network and retransmit those packets (possibly in another format) to a destination device on the other network. The switching device must generally recognize packets on one network which are addressed to devices on the other, and must generally maintain information about which devices are on which network so it may identify packets that must be copied.

When the two networks that are coupled have different network protocols, the switching device must generally be able to recognize both protocols, and must generally be able to identify the destination of packets encapsulated in each protocol. Because many network protocols are quite different, the switching device may be required to process a substantial part of each packet before it is able to identify the packet's protocol and destination. It would be advantageous for the switching device to do this as quickly as possible.

One method of the prior art is to provide the switching device with an associative memory; the initial part of the packet may then be compared simultaneously with several different expected packet headers. While this method is able to quickly recognize a small section of the packet header, such as that required for bridging, it is subject to the drawback that the extra bytes that must be matched in order for routing would make it very expensive, due to the increased width of the associative memory. Moreover, packets with variable length addresses, such as CLNP, or protocols that have variable length encapsulations, such as IPX, would require all possible combinations to be included in the associative memory; this would also be very expensive because of the increased memory requirement.

Other methods of the prior art do not achieve the simultaneous objectives of being fast, inexpensive, and having general applicability to various types of switching tasks.

Accordingly, it is an object of the invention to provide improved apparatus for packet switching.

SUMMARY OF THE INVENTION

The invention provides specialized apparatus capable of switching packets at high speed. For each packet, the apparatus may match packet data with a set of protocols, to determine how to switch the packet. In a preferred embodiment, matching of data with protocols may be highly parallel, so that the apparatus may simultaneously retrieve a data byte, compare a data byte with a protocol byte, test a comparison result, and execute a processor instruction. Apparatus comprising the invention is capable of processing many more (up to three to four times as many) packets in each instruction cycle as known packet switching devices.

In a preferred embodiment, the invention may comprise a comparator and a decision tree memory, defined herein. The comparator may comprise a plurality of (preferably three) outputs for indicating a comparison result (preferably less-than, equal-to, or greater-than). The decision tree memory may comprise a plurality of banks of addressable memory, each bank being responsive to at least one comparator output. Each memory location may preferably comprise an entry for a next location, an entry for a next data value for a next comparison, and an entry for a processor instruction.

In a preferred embodiment, the invention may further comprise a set of network interface tables, inserted into the decision tree memory, a set of network address tests, assembled or generated into the decision tree memory, and a set of protocol routing tables responsive to network data, dynamically assembled or generated into the decision tree memory.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram of a network comprising a packet switch.

Figure 2 shows a block diagram of a packet switch.

Figure 3 shows a block diagram of a packet switching engine.

Figure 4 shows a flow diagram of operation of a switching processor and switching engine.

Figure 5A shows an example format for a packet, and figure 5B shows an example section of the tree memory, for an example of operation of a switching engine.

Figure 6A shows an example network, and figure 6B shows an example section of the tree memory, for a further example of operation of a switching engine under control of a section of a tree memory, showing a bridging operation.

Figure 7A shows an example format for a packet, and figure 7B shows an example network, for an example of source route bridging. Figure 7C shows first and second example access control lists.

Figure 8 shows a block diagram of data structures used in a tree program generator.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Inventions described herein may be made or used in conjunction with inventions described, in whole or in part, in the following patents, publications, or co-pending applications, hereby incorporated by reference as if fully set forth herein:

U.S. Patent 5,088,032, issued in the name of inventor Leonard Bosack, titled "Method and Apparatus for Routing Communications Among Computer Networks".

COMPUTER NETWORK ENVIRONMENT

Figure 1 shows a block diagram of a network comprising a packet switch.

In a preferred embodiment, the invention may be used in conjunction with a computer network environment such as that shown in figure 1. (Those skilled in the art would recognize, after perusal of this application, that the environment shown in figure 1 is just an example, and that the invention would also work with other environments.) A network environment 101 may comprise a communication network 102 to which is coupled at least one host 103. Each host 103 may comprise a computer or another device which is capable of receiving a message 104 from the network and recognizing if that message 104 is addressed to that host 103. At least one host 103 must also be capable of sending a message 104 onto the network and addressing that message 104 for a destination.

Computer networks are known in the art, so this application does not describe any particular network in detail. Those skilled in the art would recognize, after perusal of this application, that the invention would work with several known networks, such as Ethernet, FDDI, Token Ring, X.25, and other known networks (both LAN and WAN), and that description of particular details of each such network is not generally required for understanding how to make and use the invention.

In a preferred embodiment, the network environment 101 may comprise a plurality of networks 102, which may possibly be the same kind (e.g., each network 102 may comprise an Ethernet), or may possibly be different kinds (e.g., a first network 102 may comprise an Ethernet, while a second network 102 may comprise a Token Ring). A pair of networks 102 may be linked by a switching device 105, sometimes called "bridge", "gateway", "router", or "brouter". As used herein, a "switch" may comprise any of these, and more generally may comprise any switching device 105 capable of receiving packets from a network 102 and retransmitting them (possibly in another form or with another protocol, although in a preferred embodiment the header is changed but the protocol remains the same) on a network 102.

It is explicitly contemplated that a switch 105 may be coupled to the same network 102 twice, such as for retransmission of certain classes of packets to a designated set of recipients. However, in the usual case, a switch 105 may be coupled to two or more networks 102, for retransmission of packets from one network 102 to the other, and possibly vice versa. Where a switch 105 is coupled to more than two networks 102, it is sometimes convenient to treat it as a collection of switches 105 for pairwise coupling those networks 102.

In a preferred embodiment, a source host 103 on a first network 102 may send a message 104 to a destination host 103 on a second network 102, by means of a switch 105. The source host 103 may send the message 104 on the first network 102, addressing the message 104 to the destination host 103. The switch 105 may receive the message 104 and recognize that it should be retransmitted ("switched") to the second network 102. The switch 105 may then retransmit the message 104 on the second network 102; this may involve re-encapsulating data from the message 104 into the protocol format used on the second network 102. The destination host 103 may then receive the (retransmitted) message 104.

In a preferred embodiment, action by the switch 105 in receiving, recognizing, and retransmitting the message 104 may be transparent to the source host 103 and the destination host 103. However, some network protocols may provide for the source host 103 to describe, or even force, an internetwork path for the message 104 to be transmitted to the destination host 103. Moreover, more than one switch 105 may be involved in transmitting the message 104. Thus, transmitting a message 104 from the source host 103 to the destination host 103 may comprise switching by a first switch 105 from the source host's network 102 to an intermediate network 102, and by a second switch 105 from the intermediate network 102 to the destination host's network 102.

In a preferred embodiment, each message 104 may comprise one or more packets 106, each of which may be formatted ("encapsulated") in a header 107 specified by a protocol used on the network 102 on which that packet 106 is transmitted. The header 107 may also comprise information about the packet 106, such as an address of a destination host 103, a packet length, a checksum, or other data considered appropriate by the designers of that protocol, generally in a predetermined order.

The switch 105 may receive every packet 106 transmitted on the first network 102, and recognize which packets 106 to retransmit to the second network 102. The switch 105 may similarly switch from the second network 102 to the first network 102. To recognize which packets 106 to retransmit, the switch 105 may examine the headers 107 and identify a destination address or other routing information. To identify this routing information, the switch 105 may generally examine the packets 106 and identify a header 107, and within the header 107 identify routing information in a location specified by the protocol for that packet 106.

Because packets are commonly switched based on eight bit bytes, the term "word" used herein generally refers to an eight bit byte, unless otherwise specified. However, those skilled in the art would recognize, after perusal of this application, that switching based on other data word sizes is within the scope and spirit of the invention.

PACKET SWITCHING DEVICE

Figure 2 shows a block diagram of a packet switch.

In a preferred embodiment, a switch 105 may comprise a network interface 201, such as Ethernet interface, FDDI interface, or Token Ring interface. The network interface 201 is coupled to the network 102 and performs low-level operations for each packet 106. Such low-level operations may comprise reading a packet 106 into a shared memory 203, and computing a checksum for the packet 106. More than one network 102 will be coupled to the switch 105, but there may be only a single network interface 201 coupled to all of those networks 102.

The switch 105 may also comprise a first internal bus 204, coupled to each network interface 201. In a preferred embodiment, the first internal bus 204 may comprise a "Cisco bus" or "CX bus", both available from Cisco Systems, Inc. of Menlo Park, California as part of one or more of its products. The first internal bus 204 may be coupled to a shared memory 203.

The shared memory 203 may be coupled, by means of a third internal bus 211, to a switching processor 205 and a switching engine 206, described in more detail with reference to figure 3. The switching processor 205 may also be coupled to the switching engine 206 by means of a set of interface registers 210 and a set of result registers 212.

The switching processor 205 and switching engine 206 may also be coupled to a second internal bus 207, which may be coupled to a high-level processor 208 and a high-level memory 209. In a preferred embodiment, the high-level processor 208 may comprise a Motorola "68000" series processor operating at 25 MHz (available from Motorola Corporation of Chicago, Illinois) and the second internal bus 207 may comprise a "Multibus" bus (available from Intel Corporation of Santa Clara, California). In a preferred embodiment, the memory 209 may comprise at least about 16 MB of memory. Although a preferred embodiment generally does not require mass storage for storing packets 106, the high-level processor 208 may comprise mass storage for other purposes, such as storing code upgrades, logging data, utility programs, or other known purposes.

In a preferred embodiment, each network interface 201 may receive packets 106 from the network 102 it is coupled to. The switching processor 205 may identify packets 106 addressed to the switch 105 itself and may forward information from those packets 106 to the high-level processor 208. Information from those packets 106 may comprise routing information from hosts 103 or other switches 105 regarding the state of the network 102, such as traffic on designated network links or quality of communication to designated other networks 102 or hosts 103. The high-level processor 208 may record routing information in a routing table in the high-level memory 209. Routing tables, and recording routing information in routing tables, are known in the art.

In a preferred embodiment, the switching processor 205 may also collect statistical information about packets 106, and forward that information to the high-level processor 208. For example, in a preferred embodiment the switching processor 205 may count the number of packets 106 transmitted on the network 102 and forward that information to the high-level processor 208 upon the latter's request. In a preferred embodiment, the high-level processor 208 may request the data periodically from the switching processor 205, e.g., every ten seconds.

In a preferred embodiment, the switching processor 205 and the switching engine 206 may operate to examine packets 106 and identify protocol patterns in headers 107. The switching processor 205 and switching engine 206 may be capable of quick operation, and may be capable of requesting the high-level processor 208 to switch a packet 106 if that packet 106 requires more complex processing, such as fragmentation. Fragmentation is known in the art.

PACKET SWITCHING ENGINE

Figure 3 shows a block diagram of a packet switching engine.

In a preferred embodiment, a packet 106 to be switched may be held in a packet buffer 301 in the shared memory 203 for review by the switching engine 206. The packet buffer 301 may be coupled to a DMA device 302, which may transfer words from the packet buffer 301 to one or more of the following: a boolean arithmetic device 306, a set of reorder registers 304, or a checksum device 305, in response to a set of control signals 325 from an instruction decoder 316.

In a preferred embodiment, the first input of the boolean arithmetic device 306 may be coupled to the reorder registers 304, the checksum device 305, a set of feedback registers 317, a pseudorandom number generator 320, and the DMA device 302. The second input of the boolean arithmetic device 306 may be coupled to a next data field 313. The boolean arithmetic device 306 may have an output coupled to a holding register 303. The output of the holding register 303 may be coupled to a third input of the boolean arithmetic device 306. The boolean arithmetic device 306 may select two of its three inputs, under control of control lines 325, and perform a boolean operation on them. The boolean operation to be performed may be any one of the boolean operations known in the art, such as but not limited to AND, XOR, and IDENTITY. The IDENTITY function would cause data to pass through the boolean arithmetic device 306 unaltered, allowing direct loading of data into the holding register 303.

The output of the holding register 303 may be coupled to an input of a comparator 307. The comparator 307 may also receive a second input comprising a data value for comparison; it may determine a set of comparison results and present those results at a set of outputs. In a preferred embodiment, the comparator may determine whether its first input is less than, equal to, or greater than the data value for comparison, and the outputs may correspond exactly to whether the less than ("<"), equal to ("="), or greater than (">") comparisons are true. However, in an alternative embodiment, the comparator 307 could generate an address or a part of an address in response to its inputs.

The outputs of the comparator 307 may be coupled to a decision tree memory 308, herein a "tree memory". In a preferred embodiment, the tree memory 308 may comprise a set of three addressable memories 309, each selected by one output of the comparator 307. Thus, one addressable memory 309 may be enabled by the "<" output, one by the "=" output, and one by the ">" output.

The tree memory 308 may also receive a second input comprising an address for indicating a memory location in each addressable memory 309 for the tree memory 308. Thus, the outputs of the comparator 307 and the second input of the tree memory 308 may collectively indicate an entry 310 in the tree memory 308. Each entry 310 may comprise a set of control values 311 for control of the switching engine. In a preferred embodiment, the control values 311 may comprise a next address 312 for the tree memory 308, a next data value 313 for comparison, and an instruction 314. The tree memory 308 may present the control values 311 at an output.

The output of.the tree memory 308 may be coupled to a set of output registers 315. In a preferred embodiment, the set of output registers 315 may comprise at least four sets of registers, that may be configured in a 2-deep or 4-deep pipeline. Pipelined registers are known in the art.

The output registers 315 may in turn be coupled to a set of control lines 322 (16 bits wide in a preferred embodiment), 323 (8 bits wide in a preferred embodiment), and 324 (8 bits wide in a preferred embodiment), that may couple the control values 311 (the next address 312, next data word 313, and instruction 314, respectively) to other circuits. The next address 312 may be coupled to the tree memory 308. The instruction 314 may be coupled to an instruction decoder 316. The next data word 313 may be coupled to the result registers 212 (figure 2), the boolean arithmetic device 306, the comparator 307, and a set of feedback registers 317.

In a preferred embodiment, the output registers 315 may also comprise a return address register 328 indicating a location in the tree memory 308. The return address register 328 may be set by a CALL instruction 314 to the current location before execution of a sub-routine. The return address register 328 may be used by a RETURN instruction 314, or by a forced return operation, described herein, to indicate the location to return to after the sub-routine is terminated or interrupted.

In a preferred embodiment, the output registers 315 may comprise circuits for ensuring that feedback between inputs to the tree memory 308 and output from the tree memory 308 are well defined. Such circuits are known in the art. The output registers 315 may be coupled to a clock circuit 326 and a set of clock control lines 327.

The next address 312 may be coupled to the second input of the tree memory 308, and may comprise an address for indicating a memory location in each addressable memory 309 for the tree memory 308, for a next instruction cycle. Alternatively, when performing a RETURN instruction 314 or a forced return operation, the return address register 328 may be coupled to the tree memory 308 and the ">" output of the comparator 307 is forced to be enabled. In a preferred embodiment, the next address 312 may comprise a 16-bit value.

The next data value 313 may be coupled to the first input of the boolean arithmetic device 306, and may comprise a set of mask bits for a boolean operation, for a next instruction cycle. In a preferred embodiment, the tree memory 308 may direct, by means of an instruction 314, that the next data value 313 may be used as a set of mask bits. However, in a preferred embodiment, data words from the packet 106 may generally be used without masking; i.e., the selected boolean operation is generally IDENTITY. The next data value 313 may also be coupled to the second input of the comparator 307, and may comprise a data value for comparison, for a next instruction cycle. In a preferred embodiment, the next data value 313 may comprise an 8-bit value.

The instruction 314 may be coupled to an instruction decoder 316, which may decode and execute the instruction 314. In a preferred embodiment, the instruction decoder 316 may comprise an ASIC, a PAL, or a similar device, such as the FPGA XC4000 device (available from Xilinx Corporation of San Jose, California). The instruction decoder 316 may output a set of control signals (not shown) for controlling registers and devices. Registers to be controlled may comprise the result registers 212, holding register 303, reorder registers 304, checksum device 305, boolean arithmetic device 306, output registers 315, as well as the feedback registers 317, a pseudorandom number generator 320, and a set of counter registers 321. In a preferred embodiment, the instruction 314 may comprise an 8-bit value.

The tree memory 308 may operate in cooperation with other circuits to comprise a finite state machine that matches packets 106 using a branching decision tree. Each address of the tree memory 308 may represent a state of the finite state machine, at which a data word of the header 107 may be compared with a known data value, one or more actions taken in response to the comparison, and a next state selected in response to the comparison. Additional state for the finite state machine may be defined by the feedback registers 317, as described herein.

In a preferred embodiment, data words from the packet 106 may be held, by means of an instruction 314, in a set of reorder registers 304, and may be coupled to the first input of the boolean arithmetic device 306. Although in a preferred embodiment data words from the packet 106 may be examined sequentially in the order in which they appear in the header 107, they may also be examined out of order. In such case, the tree memory 308 may direct, by means of an instruction 314, that a data word from one of the reorder registers may be used for a next instruction cycle, instead of a data word from the holding register 303.

Data words from the packet 106 may also be accumulated and a checksum held in the checksum device 305. The checksum device 305 may simultaneously compute checksums according to one or more protocol specifications. In a preferred embodiment, the checksum device 305 may simultaneously compute a checksum according to the IP protocol and a checksum according to the CLNP protocol.

In a preferred embodiment, the checksum device 305 may also compare the checksum it computes for each protocol against a known correct checksum. In response to a control line from the instruction decoder 316, the checksum device 305 may set result bits indicating whether the checksum is correct. These result bits may be coupled to the holding register 303, and may be used in the next tree memory operation, instead of data from the

DMA device 302. In a preferred embodiment, bit 7 may be set to indicate that the IP protocol checksum is correct, and bit 6 may be set to indicate that the CLNP protocol checksum is correct. When the tree memory 308 determines that the packet 106 was sent according to the IP particular protocol, for example, it may test the IP checksum bit and ignore the CLNP checksum bit.

Data words may also be generated by the tree memory 308 and held, by means of an instruction 314, in a set of feedback registers 317. The tree memory 308 may direct, by means of an instruction 314, that a data word from the feedback registers 317 may be loaded into the holding register 303, for use in the next tree memory operation, instead of data from the DMA device 302. As shown herein, the feedback registers 317 may be used to store par- [illegible]

[illegible]

[illegible]

mal. Free running counters advance at each instruction cycle and are known in the art. The tree memory 308 may direct, by means of an instruction 314, that a data word (i.e., a pseudorandom number) from the pseudorandom number generator 320 may be loaded into the holding register 303 for use on the next instruction cycle. A set of pseudorandom numbers generated by the pseudorandom number generator 320 may be used in load sharing for certain protocols, such as DECNET.

A set of counters 321 may also be coupled to the second input of the output registers 315. In a preferred embodiment, there may be two counters 321, each of which may be loaded with the contents of the holding register 303. For example, a length value for a variable length header field, such as that found in source route bridging, may be loaded into the holding register 303 and subsequently loaded into a counter 321 by means of an instruction 314. Each counter 321 may be set to increment or decrement (although in a preferred embodiment, counters 321 may only be set to decrement) each time a data word of the packet 106 is read. Each counter 321 may also be set to increment or decrement (although in a preferred embodiment, counters 321 may only be set to decrement) by means of an instruction 314.

Upon either counter 321 reaching zero, the output registers 314 may perform a forced return, by coupling a saved location from the return address register 328 to the address inputs of the tree memory 308. For example, the tree memory 308 may load a first counter 321 with a data value from the holding register 303, set that counter 321 to decrement, and perform a CALL instruction 314. Each succeeding data word read from the packet 106 into the holding register 303 causes the counter 321 to decrement. Upon reaching zero, the counter 321 causes the output registers 315 to perform a forced return, by coupling the address in the return address register 328 to the address inputs of the tree memory 308 and forcing the ">" output of the comparator 307 to be enabled.

Data words may be generated by the tree memory 308 and held, by means of an instruction 314, the result registers 212 (figure 2). The result registers 212 may be used for communication with the switching processor 205, such as to indicate an output network interface 201 for the packet 106 and a protocol type for the packet 106. These results allow the switching processor 205 to determine, for example, if the packet 106 may be directly output, or must be revised before output, to a selected network interface 201.

The result registers 212 may also be coupled to a rewrite engine 319, which may alter the packet 106 in response to data values stored therein, and may generate a signal indicating when it has finished.

OPERATION OF PACKET SWITCHING DEVICE

Figure 4 shows a flow diagram of operation of a switching processor and switching engine.

In a preferred embodiment, operation of the switching processor 205 and the switching engine 206, along with other circuits including the network interfaces 201 and the high-level processor 208, may proceed essentially asynchronously. Asynchronous processes are known in the art, so a detailed description of signaling between such devices is not given except where particular to the invention. Those skilled in the art would recognize, after perusal of this application, that such description is not necessary for understanding how to make or use the invention.

At a step 401, a packet 106 may be received from the network 102. A network interface 201 coupled to the network 102 may move the packet 106 into the shared memory 203 by means of the first internal bus 204. In a preferred embodiment, the shared memory 203 may comprise an input queue; a pointer to the packet 106 may be generated and appended to that queue.

At a step 402, the switching processor 205 may examine the packet 106 in the shared memory 203 by means of the third internal bus 211. In a preferred embodiment, the switching processor may examine the interface memory's input queue, may remove the first element from that queue, and may examine the packet 106 pointed to by that first element.

At a step 403, the switching processor 205 may place a pointer to the packet 106 into the interface registers 210. In a preferred embodiment, the shared memory 203 may comprise one or more buffer areas; the switching processor 205 may move the packet 106 into a buffer area with an area of free memory preceding the header 107, and may generate a packet pointer 410 to point to the first word of the packet 106.

At a step 404, the switching engine 206 may examine the interface registers 210 and retrieve the packet pointer 410 to the packet 106.

At a step 405, the switching engine 206 may operate under control of the tree memory 308. The switching engine 206 may examine the packet 106 and may place a set of results in the result registers 212.

At a step 406, the rewrite engine 319 may alter the packet 106 in response to data values stored therein, and may generate a signal for indicating when it has finished.

At a step 407 (concurrent with step 406), the switching processor 205 may examine the result registers 212. In a preferred embodiment, the switching processor 205 may determine to which network 102 the packet 106 should be routed, and may adjust the packet's header checksum, hop count, packet length, "time to live", and other parameters. The switching processor 205 may also wait for the signal indicating that step 406 is complete before control proceeds to step 408.

At a step 408, the switching processor 205 may append the packet 106 to an output queue in the shared memory 203. In a preferred embodiment, the shared memory 203 may comprise one or more output queues for each network interface 201, and the selection of which output queue onto which the packet 206 is placed may be in response to data in the result registers 212.

At a step 409, a network interface 201 (possibly different from the network interface 201 that received the packet 106) may output the packet 106 to a network 102 (possibly different from the network 102 from which the packet 106 was received). In a preferred embodiment, the network interface 201 may remove the first element from the output queue, and may output the packet 106 pointed to by that first element.

OPERATION OF PACKET SWITCHING ENGINE

Figure 5A shows an example format for a packet, and figure 5B shows an example section of the tree memory, for an example of operation of a switching engine.

In this example, each expected packet type has a protocol format as shown in table 5-1 herein. As shown in the table, more than one format may be valid for certain protocols. The protocol format data may be used to prepare the tree memory 308 with a set of nodes, organized as a directed graph, for classifying the packet 106. However, for simplicity, only a subsection of the tree memory 308 is shown.

The tree memory 308 may be prepared ahead of time with a set of static values for representing the protocol format data. In response to protocol format data, a program may generate a set of values for insertion into the tree memory 308. Alternatively, as the protocol format data does not change rapidly, the protocol format data may be coded directly in a format for insertion into the tree memory.

In a preferred embodiment, the tree memory 308 may be initiated with a predetermined tree memory location A; the result of the last comparison remains undetermined. The tree memory 308 entry 310 for location A may therefore preferably comprise a NOP (no operation), as described herein, with all its branches pointing to a second predetermined location B. As all branches at location A point to location B, the tree memory 308 entry 310 at location B is sure to be executed second with a defined result of the last comparison. Location B is thus where normal execution begins.

In this example, the packet 106 may be received on an Ethernet network 102, where the packet's maximum length is 1526 (decimal) bytes. Thus in this example, the length field (shown in the table as two bytes denoted "len len") is always less than 06 00 hexadecimal. As the minimum fixed value specified by any protocol format is 06 00 (hexadecimal), there should be no packets 106 which could be valid under more than one format.

In this example, each data word of the packet 106 is an eight bit byte, expressed as two hexadecimal digits. Thus for example, 03 represents the bit pattern 0000 0011.

In this example, each location of the tree memory 308 has three values, separated by dots, each of which comprises a next address 312 pointing to the next node, an eight bit byte for the next data value 313 for comparison, and an instruction 314. The next address 312 is represented by an arrow pointing to a next location. The representation of an instruction 314 may include a "+" symbol to indicate that the instruction 314 directs the packet pointer 410 to advance (i.e., the instruction bit for that action is set). Thus for example, [80.+00.--] would represent three values, 80, +00 and --. The first, 80, indicates next comparing with hexadecimal 80; the second, +00, indicates advancing the packet pointer 410 and next comparing with 00; the third, --, indicates a no-operation (i.e., do nothing).

In this example, a no-operation is indicated to show that the type of the packet 106 has been recognized, or determined to be of a type that is not known. In practice, when the type of the packet 106 has been recognized, the "--" would be replaced with the next instruction 314 in a subsection of the tree memory 308 for processing that type of packet 106. When the type of the packet 106 has been determined to be one that is not known, the tree memory 308 would move on to process the next packet 106.

An instruction 314 "+" indicates advancing the packet's pointer and no further operation. In practice, the "+" would similarly be replaced with the next instruction 314 in a subsection of the tree memory 308 for processing that type of packet 106, with the instruction bit set for advancing the packet pointer 410.

In this example, the packet buffer 301 holds a packet 106 transmitted on an Ethernet network 102. After a destination address 501 and a source address 502, the packet may comprise a type field 503, followed by the remainder of the packet 106. The type field 503 may comprise a 16-bit type value, or it may comprise a length.

In a preferred embodiment, the tree memory 308 may comprise subsections for parsing and recognizing the destination address 501 and the source address 502. After parsing and recognizing the destination address 501 and the source address 502, the tree memory 308 may parse and recognize the type field 503. This example shows parsing and recognition of the IP, Apollo, and Appletalk1 type fields 503.

/ / /

Type Field	Encapsulation Data	
IP	08	00
Apollo	80	19
Appletalk1	80	9B

In a first subexample, the packet 106 is an IP packet. After the destination address 501 and the source address 502 the packet 106 has the following data:

08 00 <IP information>

In this first subexample, the 08 00 identifies the packet 106 as an IP packet. The packet pointer 410 will start out pointing at the 08 byte in the packet 106; the tree memory 308 will start out at a top node 1001, which is [80.80.80]. In practice, the comparison result would be defined by an outcome of a comparison step from a previous operation, such as parsing the source address. However, in this example, the comparison result is said to be initially undefined, but is one of less than, equal to, or greater than.

At node 1001, the comparator 307 compares the packet's byte 08 with the data value 80, and the tree memory 308 continues with the next node 1002, which is [08.+19.--]. The comparison result is "<", because 08 < 80.

At node 1002, the comparator 307 compares the packet's byte 08 with the (less than) data value 08, and the tree memory 308 continues with the next node 1003, which is [--.+00.--]. The comparison result is "=", because 08 = 08. The (equal to) selection of the next node 1003 is +00, advancing the packet pointer 410 so it will point to the next byte, i.e., the 00 byte.

At node 1003, the comparator 307 compares the packet's byte 00 with the (equal to) data value 00, and the tree memory 308 continues with the next node 1005, which is [--.+.--]. The comparison result is "=", because 00 = 00. The next node 1007 begins parsing of the IP information.

In a second subexample, the packet 106 is an Appletalk1 packet. After the destination address 501 and the source address 502 the packet 106 has the following data:

80 9B <Appletalk1 information>

In this second subexample, the 80 9B identifies the packet 106 as an Appletalk1 packet. The packet pointer 410 will start out pointing at the 80 byte in the packet 106; the tree memory 308 will start out at a top node 1001, which is [80.80.80]. As noted for a previous example, the comparison result is said to be initially undefined, but is one of less than, equal to, or greater than.

At node 1001, the comparator 307 compares the packet's byte 80 with the data value 80, and the tree memory 308 continues with the next node 1002, which is [08.+19.--]. The comparison result is "=", because 80 = 80. The (equal to) selection of the next node 1002 is +19, advancing the packet pointer 410 so it will point to the next byte, i.e., the 9B byte.

At node 1002, the comparator 307 compares the packet's byte 9B with the (equal to) data value 19, and the tree memory 308 continues with the next node 1004, which is [--.+.9B]. The comparison result is ">", because 9B > 19.

At node 1004, the comparator 307 compares the packet's byte 9B with the (greater than) data value 9B, and the tree memory 308 continues with the next node 1006, which is [--.+.--]. The comparison result is "=", because 9B = 9B. The next node 1007 begins parsing of the Appletalk1 information.

In a third subexample, the packet 106 is an unknown type of packet 106. After the destination address 501 and the source address 502 the packet 106 has the following data:

18 99 <further information>

In this third subexample, the 18 99 does not identify the packet 106 as any known type. The packet pointer 410 will start out pointing at the 18 byte in the packet 106; the tree memory 308 will start out at a top node 1001, which is [80.80.80]. As noted for a previous example, the comparison result is said to be initially undefined, but is one of less than, equal to, or greater than.

At node 1001, the comparator 307 compares the packet's byte 18 with the data value 80, and the tree memory 308 continues with the next node 1002, which is [08.+19.--]. The comparison result is "<", because 18 < 80.

At node 1002, the comparator 307 compares the packet's byte 18 with the (less than) data value 08, and the tree memory 308 continues with the next node 1003, which is [--.+00.--]. The comparison result is ">", because 18 > 08. The next node 1003 discards the packet 106 as being of an unknown type.

Figure 6A shows an example network, and figure 6B shows an example section of the tree memory, for a further example of operation of a switching engine under control of a section of a tree memory, showing a bridging operation.

In this further example, the packet 106 may be addressed from any one of hosts 103 A, B, C, or D, on one of two networks 102, to any other one of those hosts 103. A switch 105 may perform bridging between these two networks 102, and may have a zeroth network interface 201 to a zeroth network 102 and a first network interface 201 to a first network 102. In this example, the switch 105 has already received packets 106 allowing it to determine the location of each of the hosts 103 in the figure. This is sometimes called "learning" an address; learning an address is known in the art.

If the switch 105 is performing both bridging and routing, in addition to matching addresses for bridging, it will match its own address in the destination address field, in case it is being asked to route the packet. Performing bridging and routing in the same switch 105 is known in the art.

A section of tree memory 308 may comprise a decision tree 601, entered at a location 602 BB, at which a new packet 106 is received and processed. In the first decision tree, a reorder register 304 R0 may be set to indicate a network interface 201 from which the packet 106 was received, a feedback register 317 F0 may be set to indicate a phase 0 for matching the destination address for the packet 106, and a result register 318 RR2 may be set to indicate no need to "learn" the source address of the packet 106.

The tree memory 308 proceeds to a decision tree 603, entered at a location 604 AA, at which a destination address or a source address in the packet 106 may be parsed and recognized. The processes of parsing and recognizing destination and source addresses are known in the art. Accordingly, those skilled in the art would recognize, after perusal of this application, how to construct a section of tree memory 308 for conducting such parsing. Four possible results, one for each possible host 103, are shown. Treatment of broadcast, multicast, or other types of packet 106 are left out of this example to keep it simple. Those skilled in the art will recognize, after perusal of this application, that treatment of broadcast, multicast, or other types of packet 106 would be workable, and are within the scope and spirit of the invention.

The tree memory 308 proceeds to a decision tree 605 for hosts 103 A or B (input from the zeroth network interface 201), or to a decision tree 606 for hosts 103 C or D (input from the first network interface 201).

At the decision tree 605, the tree memory 308 may test feedback register 317 F0, and may proceed to a decision tree 607 for a "0" (phase 0, matched the destination address), or to a decision tree 608 for a "1" (phase 1, matched the source address).

At the decision tree 607, the tree memory 308 may test reorder register 304 R0, and may proceed to a decision tree 609 for a "0" (the zeroth network interface 201), or to a decision tree 610 for a "1" (the first network interface 201).

At the decision tree 608, tree memory 308 may proceed to a following decision tree for parsing the protocol type, as described with reference to figure 5B.

At the decision tree 609, the tree memory 308 may set result register 318 RR0 to indicate that the packet 106 should be sent to its destination address. The tree memory 308 may then proceed with a further decision tree 611. At this point, the tree memory 308 has identified the packet 106 as having come from one network 102 and being destined for the other network 102; hence, it should be sent on to its destination. Since the destination is "A" or "B", the packet 106 should be sent on to the zeroth network interface 201.

At the decision tree 610, the tree memory 308 may set result register 318 RR0 to indicate that the packet 106 should be discarded. The tree memory 308 may then proceed with a further decision tree 620. At this point, the tree memory 308 has identified the packet 106 as having come from one network 102 and being destined for the same network 102; hence, it has already reached its destination via that network 102, and may proceed to a following decision tree for parsing the protocol type, as described with reference to figure 5B.

At the decision tree 611, the tree memory 308 may set the result register 318 RR1 to indicate that the packet 106 should be output on the zeroth network interface 201. The tree memory 308 may then proceed with a further decision tree 612.

At the decision tree 612, the tree memory 308 may set the feedback register 317 F0 to indicate a phase 1 for matched the source address for the packet 106, and may proceed to the decision tree 603, entered at a location 604 AA.

At the decision tree 606, the tree memory 308 may similarly test feedback register 317 F0, and may proceed to a decision tree 614 for a "0" (phase 0, matched the destination address), or to the decision tree 608 for a "1" (phase 1, matched the source address).

At the decision tree 614, the tree memory 308 may similarly test reorder register 304 R0, and may proceed to a decision tree 615 for a "0" (the zeroth network interface 201), or to a decision tree 616 for a "1" (the first network interface 201).

At the decision tree 615, the tree memory 308 may similarly set result register 318 RR0 to indicate that the packet 106 should be sent to its destination address. The tree memory 308 may then proceed with a further decision tree 617. At this point, the tree memory 308 has identified the packet 106 as having come from one network 102 and being destined for the other network 102; hence, it should be sent on to its destination.

At the decision tree 616, the tree memory 308 may similarly set result register 318 RR0 to indicate that the packet 106 should be discarded. The tree memory 308 may then proceed with a further decision tree 621. At this point, the tree memory 308 has identified the packet 106 as having come from one network 102 and being destined for the same network 102; hence, it has already reached its destination via that network 102, and may proceed to a following decision tree for parsing the protocol type, as described with reference to figure 5B.

At the decision tree 617, the tree memory 308 may similarly set result register 318 RR1 to indicate that the packet 106 should be output on the first network interface 201. The tree memory 308 may then proceed with a further decision tree 618.

RECOGNITION OF OTHER PACKET INFORMATION

In a preferred embodiment, the switch 105 may recognize other packet information and use that information for switching. Two examples are illustrative:

The packet 106 may comprise information that tells the switch 105 how to route the packet; this is sometimes called "source route bridging". Thus for example, the source host 103 may determine onto which networks 102 the packet 106 must be switched, and in what order, and may provide that information in a routing information field in the packet 106. The switch 105 must generally determine if the routing information field in the packet 106 indicates that the packet 106 should be switched between two networks 102 the switch 105 is coupled to. If so, the switch 105 should retransmit the packet 106 from one network 102 to the other network 102, but if not, the switch 105 should generally ignore the packet 106.

Figure 7A shows an example format for a packet, and figure 7B shows an example network, for an example of source route bridging.

In this example, the packet 106 comprises a routing information field 701 (RIF), that comprises a length value and a sequence of networks 102 and switches 105 forming a route from the source host 103 to the destination host 103. A value for the final switch 105 in the RIF 701 may be zero to indicate that the packet 106 may at that point be delivered to its destination host 103. In a preferred embodiment, the RIF 701 may also comprise other values that are known in the art, but are not described here because they are not necessary for an understanding of the invention.

One particular switch 702 will serve for this example. As each switch 105 knows which networks 102 it is coupled to, and which switch 105 it is, the example switch 702 knows which networks 102 for which it should route packets 106. When a packet 106 comprising a RIF 701 is recognized by the switch 702, it parses the RIF 701 and looks for a route that includes two networks 102 to which it is coupled and its own switch number.

In a first subexample, the packet 106 comprises a RIF 701, and the RIF 701 comprises a pair of networks 102 and the switch number for the example switch 702; the pair of networks 102 are coupled to the example switch 702. Accordingly, the switch 702 recognizes the packet 106 and switches it from a first network 102 in the RIF 701, parsed as above, to the next network 102 in the RIF 701.

In a second subexample, the packet 106 comprises a RIF 701, but the RIF 701 does not comprise a pair of networks 102 for which the example switch 702 should route packets 106. Accordingly, the switch 702 simply discards the packet 106.

In a preferred embodiment, the switch 105 may load the length value found in the RIF 701 into a counter 321, and decrement the counter 321 repeatedly while reading data words from the packet 106. When the counter 321 reaches zero, a forced return operation will occur, and the tree memory 308 will be found in a state where the entire RIF 701 has been processed, but no pair of networks 102 for which the switch 105 should route packets 106 has been found. Accordingly, the switch 105 will simply discard the packet 106.

Another example shows parsing of access control lists.

The switch 105 may be provided with an access control list that tells the switch 105 which devices are allowed to transmit messages to destinations on particular networks. Thus for example, a designated network may prohibit some or all of its hosts 103 from transmitting to destination hosts 103 on other networks 102, or may prohibit some or all hosts 103 on other networks 102 from transmitting to destination hosts 103 on that network 102. The switch 105 may be provided with an access control list that tells it which source addresses (or destination addresses, or combinations of source and destination addresses) are allowed. The switch 105 must generally determine if the destination address for each packet 106 is allowed. If so, the switch 105 should process the packet 106 normally (possibly switching it), but if not, the switch 105 should generally prohibit the packet 106 from reaching its designated destination, typically by refusing to switch it.

Figure 7C shows first and second example access control lists.

An access control list 751 may comprise an identifier 752, a set of permissions 753 (which may explicitly permit access, explicitly deny access, or limit access to particular protocols), and a set of host addresses 754 (which may be source host addresses or destination host addresses). As with switching packets 106 in response to destination host addresses, the switch 105 may permit, deny, or limit access in response to an active access control list and in response to the source and destination host addresses in a packet 106.

In a preferred embodiment, the switching engine 206 may parse the packet 106 and recognize the destination host address and the source host address. In addition to determining to which output network interface 201 the packet 106 should be switched, the switching engine 206 may also determine (in response to an active access control list) whether switching the packet 106 would violate access control. If so, the switch 105 may take appropriate action, such as discarding the packet or issuing a warning message.

In a preferred embodiment, active access control lists may be converted by the high-level processor 208 from the high-level memory 209 into the tree memory 308 similarly to routing tables.

TREE PROGRAM GENERATOR

Figure 8 shows a block diagram of data structures used in a tree program generator.

As noted herein, the high-level processor 208 may comprise a tree program generator 801 for converting information from a routing table 802 in high-level memory 209 into functional subsections ("subtrees") 803 in the tree memory 308, each of which may parse and recognize a portion of each packet 106. The tree program generator 801 may reside in high-level memory 209 and may be executed by the high-level processor 209.

In a preferred embodiment, the high-level processor 208 may comprise a set of console commands, to be entered by an operator at an input device coupled thereto. The console commands may be interpreted by the high-level processor 208 and may comprise commands for initializing the routing tables, forcing recomputation of the routing tables, displaying information about the switch 105, and placing tree memory programs into the tree memory 308.

In a preferred embodiment, the tree memory 308 may comprise a static section 805 and a dynamic section. The static section 805 may comprise information relating to classification of packets 106 by protocol, and may be assembled into the tree memory 308 in response to known information about protocol formats. The dynamic section may comprise information relating to routing and other information (such as access control) about the networks 102 to which the switch 105 is coupled, and may be dynamically generated and placed into the tree memory 308 in response to network information the switch 105 gleans from the network 102.

The high-level processor 208 may prepare a routing table in the high-level memory 209, in response to network information the switch 105 gleans from the network 102. In a preferred embodiment, the high-level processor 208 may prepare instructions for the tree memory 308 (i.e., it may prepare data for loading into the tree memory 308) under control of software for converting the routing table into tree memory instructions, herein a "tree program generator".

In a preferred embodiment, the high-level processor 208 may maintain the routing table dynamically, i.e., updating it in response to new information from the network 102 so that it is always current. The high-level processor 208 may occasionally generate a new set of tree memory instructions in response to the routing table, and place the new set of tree memory instructions into the tree memory 308. For example, the high-level processor 208 may generate the new set of tree memory instructions in response to events that are likely to cause the tree memory 308 to be "out of date", such as major changes in the routing table, and may also periodically, such as in response to a timer, recognize that sufficient time has passed to require the tree memory 308 to be updated.

In a preferred embodiment, the tree program generator may divide the tree memory 308 into a set of functional subsections ("subtrees"), each of which may parse and recognize a portion of each packet 106. For example, a first subtree 803 may parse and recognize information relating to protocol classification, a second subtree 803 may parse and recognize information relating to source-route bridging, and a third subtree 803 may parse and recognize information relating to a particular set of destination addresses. Each subtree may be coupled to the static section 805 of the tree memory 308.

Since each subtree 803 may comprise an independent program for parsing and recognition of information about the packet 106, the tree program generator 801 may independently generate information for each subtree 803, and place those subtrees 803 in the tree memory 308. In particular, the tree program generator 801 may independently generate information regarding each set of destination addresses, and may generate a subtree for each such set.

In a preferred embodiment, the tree program generator 801 may generate a separate functional subtree 803 for each packet protocol type. As host addresses for each protocol type are parsed and recognized, the high-level processor 208 may add them to the routing table 802 using a weighted tree representation 804. The high-level processor 208 may generate a weighted tree 804 of addresses, weighted by usage so that a minimal number of comparisons may generally be needed to recognize each address.

For example, in a weighted tree 804, a likely host 102 address may be placed near the top of the weighted tree 804, so that it may be disposed of early in testing. If hosts A, B, C, D, E, F and G are added to the weighted tree 804, but host G receives the vast bulk of packets 106, host G should be placed at the top of the weighted tree 804. Because the likely host address is more common, testing for it early should reduce the average number of tests to be performed. Weighted trees are known in the art, as are methods for generating them.

The tree program generator 801 may also perform destination aggregation. Where there are plural destinations that can all be switched in response to a common subset of the full address, the tree program generator 801 may generate a single functional subtree 803 to recognize the common subset and switch the packet 106 uniformly in response thereto. For example, if two different destinations are always switched to the same output network interface 201, the tree program generator 801 may generate a single functional subtree 803 to recognize their common subset and switch to that output network interface 201, regardless of whether differential processing will occur elsewhere along the path to the final destination, after the packet 106 is switched.

The tree program generator 801 may also perform common subtree elimination. Prior to placing a functional subtree 803 to the tree memory 308, the tree program generator 801 may review the subtree 803 and combine any nodes that are identical. In a preferred embodiment, this operation may be performed before converting the weighted tree 804 to tree memory format.

The tree program generator 801 may also perform other known optimizations on the functional subtrees 803 before placing them to the tree memory 308, such as peephole optimization and other forms of optimization known in the art.

The tree program generator 801 may then generate the weighted tree 804 by generating instructions in a tree memory format, forming those instructions into a functional subtree 803, and linking that functional subtree 803 to other functional subtrees 803 in the tree memory 308 or to the static section 805 in the tree memory 308. Where necessary, the tree program generator 801 may trim the set of functional subtrees 803 to fit into the tree memory 308, for example by removing rare cases and converting them into calls on the high-level processor 208 to complete the parsing of that packet 106.

In a preferred embodiment, the switching engine 206 may also comprise a watchdog timer (not shown), that must be reset periodically. Watchdog timers are known in the art. If the watchdog timer is not reset, an interrupt may be generated for the switching engine 206, the switching processor 205 may seize control of switching the packet 106, and the high-level processor 208 may be interrupted to take over switching the packet 106. The watchdog timer prevents the switching engine 206 from entering an endless loop for a particular packet 106; it thus also serves as a check on the tree program generator 801 so that functional subtrees 803 with endless loops therein are not loaded into the tree memory 308 (or at least are recognized when the tree memory 308 attempts to execute them).

In a preferred embodiment, the high-level processor 208 may place diagnostic functional subtrees 803 into the tree memory 308, present test packets 106 to these diagnostic functional subtrees 803 for testing, and examine the results produced by the tree memory 308. This allows the high-level processor 208 to test the tree memory 308.

As noted herein, it may occur that the tree memory 308 is not large enough to hold a tree program 803 for matching the entire set of destination addresses. Accordingly, the tree program generator 801 may periodically generate tree programs 803, in response to observed traffic patterns, that are limited to the size of the tree memory 308, and that will have the minimal (or at least near-minimal) likelihood of a destination address not being matched by the tree memory 308. When a destination address is not matched by the tree memory 308, it may call upon the high-level processor 208 to match the destination address using the complete routing table.

INSTRUCTION DECODER

As described herein, the instruction 314 may comprise an eight bit data word. The instruction 314 may comprise a clock-in bit, for indicating that the instruction decoder 316 should direct the packet pointer 410 to be incremented to point to a next byte of the packet 106, and a checksum bit, for indicating that the instruction decoder 316 should direct the checksum device 305 to incorporate the next byte of the packet 106 in a checksum.

In a preferred embodiment, a remaining six bits of the instruction 314 may comprise an instruction opcode, for designating one of a plurality of possible instructions for the instruction decoder 316 to implement. Instruction opcodes are known in the art.

In a preferred embodiment, the instruction opcode may comprise one of a set of instruction opcodes for implementing processor tasks suited to switching processors. Such sets of instruction opcodes are known in the art. The following list of operations designated by such instruction opcodes is preferred. (Each operation is followed by its hexadecimal opcode value in parenthesis.)

NOP (00). No operation; do nothing.

CALL (01). Call a subroutine: load the return address register 328 with the current tree memory address, and transfer control to the next tree memory address. Subroutine calls are not nested in a preferred embodiment. A RET (return) instruction 314, or a return forced by a predefined condition, returns control to the location after the CALL instruction 314.

HANG (02). Stop operation, and generate an error signal that the switching processor 205 may detect.

RET (03). Return from a subroutine: use the contents of the return address register 328 as the next tree memory address and force a ">" comparison result. Because the RET instruction 314 forces a ">" result, it is common to compare with hexadecimal FF before a CALL instruction 314 so the "<" or "=" branches are taken for the call.

NEXT_DMA (04). Instruct the DMA device 302 to input the next packet 106.

AND_PIPE (05). Perform a logical "AND" of the holding register 303 with the next data value 313 from the tree memory 308, and store the result in the holding register 303.

LD_COUNT1 (06). Load the first counter register 321 with a data word from the holding register 303. A forced return occurs when the counter register 321 reaches zero. This allows the tree memory 308 to set a counter to indicate a number of data words of the packet 106 to examine, and continue to examine those data words in a loop until the counter reaches zero.

LD_COUNT0 (07). Same as the LD_COUNT1 instruction 314, except that the zeroth counter register 321 is loaded.

As noted herein, a "forced return" occurs when a counter 321 reaches zero. The location in the return address register 329 is selected as the next address for the tree memory 308, and the ">" output from the comparator 307 is forced to be enabled. This allows counting down of a variable length fields, for example, by loading a length value for the field into a counter 321 and calling a subroutine that processes each data word in the field.

When the counter 321, a forced return occurs, and processing of the variable length field is complete.

SET_DEC (08). Enable the zeroth and first counter register 321 to decrement. Once loaded with a nonzero value and enabled, a counter register 321 is decremented by one each time a RD_BYTE instruction is executed.

RST_DEC (09). Disable the zeroth and first counter register 321 from decrementing.

LD_SPAGE (0A). Load the scratchpad page register (not shown) with the next data value 313 from the tree memory 308. The page register is automatically incremented when the LD_SREG_15 or RD_SREG_15 instruction 314 is executed, and is automatically loaded with the next data value 313 from the tree memory 308 when the DONE instruction 314 is executed.

The page register indicates which set of memory locations are being used for the reorder registers 304 and feedback registers 317. In a preferred embodiment, bit 7 of the page register indicates whether the page is a set of reorder registers 304 or a set of feedback registers 317.

XOR_SREG_B (0B). Perform a logical "XOR" of the holding register 303 with the contents of scratchpad register 0B (either a reorder register 304 or a feedback register 317, depending upon the page register).

RD_RAND (0C). Read an 8-bit pseudorandom number into the holding register 303, and perform a logical "AND" with the next data value 313 from the tree memory 308.

RD_CKSUM (0D). Read the output from the checksum device 305 into the holding register 303, and clears the output from the checksum device 305.

DONE (0E). Set the "DONE" signal, indicating that the switching engine 206 is done.

DEC_COUNT (0F). Decrement whichever of the zeroth or first counter registers 321 contains a nonzero value.

LD_RSLT_n (1n, n = 0 to F). This is a set of 16 opcodes. Load the nth result register 318 with the next data value 313 from the tree memory 308. In a preferred embodiment, there are 16 result registers 212, labeled 0 to F in hexadecimal.

In a preferred embodiment, certain of the result registers 212 have predetermined meaning, such as a packet classification code, an output network interface, input and output packet header length, a memory address of the packet 106 for use by the rewrite engine 319, and a status code of the switching engine 206 for use by the switching processor 205.

LD_SREG_n (2n, n = 0 to F). This is a set of 16 opcodes. Load the nth scratchpad register with a data value. As noted herein, the designated scratchpad register may be a reorder register 304 or a feedback register 317, depending on the contents of the page register. The data value to be loaded depends on the most significant bit of the page register. If 0, the next data value 313 from the tree memory 308 is used. If 1, the next data word from the packet 106 is used.

RD_SREG_n (3n, n = 0 to F). This is a set of 16 opcodes. Read the nth scratchpad register into the holding register 303. The contents of the scratchpad register are logical "AND"-ed with the next data value 313 from the tree memory 308 before storing into the holding register 303.

PARALLEL OPERATION OF THE SWITCHING PROCESSOR AND ENGINE

The switching processor 205 and the switching engine 206 may be considered to collectively comprise a parallel processor for quickly switching packets 106.

A general purpose processor generally comprises an instruction fetch element for fetching instructions from an instruction memory, one or more execution elements for executing the instructions that are fetched, a data fetch element for fetching data from a data memory for execution, and a write back element for writing results of execution back to the data memory.

The switching processor 205 and switching engine 206 may be considered to comprise similar elements, where packets 106, rather than data words, are the elements for fetch and execution. In this view, the instruction fetch element may comprise the network interface 201 and related means for retrieving a packet 106 from a network 102. The execution element may comprise the switching engine 206; a preferred embodiment of the invention may comprise more than one switching engine 206, operating in conjunction with the switching processor 205. The data fetch element may comprise the rewrite engine 319 and means for adjusting the packet header after the switching engine 206 has completed. The write back element may comprise packet 106 postprocessing and means for moving the packet 106 to an output queue for switching.

SWITCHING ENGINE SPEED

The switching engine 206 is capable of fetching two data elements, comparing them, testing a result of a prior comparison, and executing an instruction in response to that result, all in a single clock cycle. The switching engine 206, operating in cooperation with the switching processor 205 and the high-level processor 208, is capable of switching about 300 kilopackets per second or more when operating with a clock cycle of about 30 nanoseconds (for the switching engine 205, twice that for the switching processor 206, and much greater for the high-level processor 208).

The switching engine's speed compares favorably with a switching speed of about 50 to 100 kilopackets per second achieved by devices having a similar clock cycle but not using a switching engine 206 as described herein.

Alternative Embodiments

While preferred embodiments are disclosed herein, many variations are possible which remain within the concept and scope of the invention, and these variations would become clear to one of ordinary skill in the art after perusal of the specification, drawings and claims herein.

CLAIMS

1. A device for switching packets, comprising

a first memory coupled to a network interface, said memory being large enough to hold a packet data word;

a comparator having a first input coupled to said first memory and having a second input;

a second memory having a first input coupled to a comparison output of said comparator, and having a second input, said first and second inputs collectively referencing a location in said second memory;

at least part of said location comprising a next data word and being coupled to said second input of said comparator;

at least part of said location comprising a next address and being coupled to said second input of said memory; and

at least part of said location comprising a next instruction and being coupled to an instruction decoder.

2. A device as in claim 1, comprising

a set of counters, wherein a counter comprises means for decrementing upon reading a data word of said packet;

means for coupling an address to said second memory upon reaching a predetermined counter value, without requiring an explicit test and branch instruction.

3. A device as in claim 1, comprising

a set of feedback registers coupled to an output of said second memory and to said instruction decoder.

4. A device as in claim 1, comprising

a set of reorder registers coupled to said first memory and to said instruction decoder.

5. A device as in claim 1, comprising

a set of result registers coupled to said second memory and to said instruction decoder; and

a rewrite engine coupled to said first memory and to said set of result registers.

6. A device as in claim 1, wherein a packet having said packet data word may comprise one of a plurality of packet transmission protocols.

7. A device as in claim 1, wherein

said comparison output comprises a plurality of output signals;

said second memory comprises a plurality of memory sections, each coupled to at least one of said plurality of output signals, whereby exactly one of said plurality of memory sections is referenced by said plurality of output signals.

8. A device as in claim 1, wherein said instruction decoder comprises

a next word circuit coupled to at least part of said next instruction;

a checksum bit circuit coupled to at least part of said next instruction, said checksum bit circuit being coupled to a checksum device;

an opcode circuit coupled to at least part of said next instruction, to said done bit, and to said checksum bit;

said opcode circuit configured to recognize a first instruction for setting said done bit to a first predetermined value; and

said opcode circuit configured to recognize a second instruction for setting said checksum bit to a second predetermined value, whereby said checksum device operates in response to said second instruction.

9. A device as in claim 8, wherein said opcode circuit is configured to recognize a third instruction for testing an output of said checksum device.

10. A device as in claim 1, wherein said instruction decoder comprises

an opcode circuit coupled to at least part of said next instruction, to a counter, and to a return location register;

said opcode circuit configured to recognize a CALL instruction for calling a subroutine, and responsive to said CALL instruction by placing a value in said return location register;

a circuit coupled to said counter and configured to recognize a predetermined value held therein, and configured to retrieve a value from said return location register and to forcing a predetermined result from said comparator in response thereto.

11. A device as in claim 10, wherein said counter is configured to change state each time a packet data word is read from said first memory.

12. A device as in claim 10, wherein said counter is configured to change state each time a packet data word is read from said first memory, responsive to an enabling circuit, and wherein said opcode circuit is coupled to said enabling circuit and configured to put said enabling circuit in a predetermined state in response to an instruction.

13. A device as in claim 10, wherein said opcode circuit is configured to recognize a RETURN instruction for returning from a subroutine, and responsive to said RETURN instruction by retrieving a value from said return location register and by forcing a predetermined result from said comparator.

14. A device as in claim 1, wherein said instruction decoder comprises

an opcode circuit coupled to at least part of said next instruction, to a memory page register, and to a third memory having a plurality of sets of addressable reorder registers and a plurality of sets of addressable feedback registers;

said memory page register comprising a first circuit indicating a choice between said reorder registers and said feedback registers;

said memory page register comprising a second circuit indicating a choice of one of said plurality of sets of reorder registers and one of said plurality of sets of feedback registers; and

said opcode circuit configured to recognize a first set of instructions, each for addressing and altering one of said reorder registers, and a second set of instructions, each for addressing and altering one of said feedback registers.

15. A device as in claim 14, wherein said opcode circuit is configured to alter said memory page register in response to an instruction.

16. A device as in claim 7, wherein

said plurality of output signals comprise a less than signal, an equal to signal, and a greater than signal; and

said plurality of memory sections comprises a section activated by said less than signal, a section activated by said equal to signal, and a section activated by said greater than signal.

17. A device as in claim 7, wherein said second memory comprises a location in each one of said plurality of memory sections for each address coupled to said memory.

18. A device for switching packets, comprising

means for receiving a packet from a first one of a plurality of network interfaces;

a tree memory comprising a set of locations each having a next data word, a next address and a next instruction, said set of locations comprising a first region with static routing information about a network, said network being coupled to said first one network interface;

means for receiving dynamic routing information about said network;

means for compiling said dynamic routing information into a second region in said set of locations; and

means for sending said packet to a second one of said plurality of network interfaces in response to said tree memory.

19. A device as in claim 18, comprising

means for identifying routing information in said packet in response to said tree memory; and

means for directing said means for sending to switch said packet in response to said means for identifying.

20. A device as in claim 19, comprising

means for receiving dynamic routing information about a network, said network being coupled to said first one network interface;

means for compiling said dynamic routing information into a region in said second memory.

21. A device as in claim 20, said second memory comprising static routing information about said network.

22. A device as in claim 18, wherein said dynamic routing information comprises information about locations of devices coupled to said network or information about access control for devices coupled to said network.

23. A device as in claim 18, wherein said static routing information comprises information about a protocol used on said network.

24. A device for switching packets, comprising

means for receiving a packet from a first one of a plurality of network interfaces;

means for preparing an interface register in response to said packet;

a tree memory having a set of locations each having a next data word, a next address and a next instruction;

an instruction decoder coupled to said next instruction and to a result register;

means for signaling said tree memory to process said packet;

means for rewriting said packet in response to said result register;

means for selecting a second one of said plurality of network interfaces in response to said result register;

means for sending said packet to said second one network interface.

25. A device for switching packets, comprising

means for receiving a packet from a first one of a plurality of network interfaces;

means for sending said packet to a second one of said plurality of network interfaces;

means for switching said packet from said first one network interface to said second one network interface;

said means for switching having a clock cycle time defined to equal a shortest time needed to decode a processor instruction, and having a clock cycle rate defined to equal an inverse of said clock cycle time;

said means for switching having a packet switching rate defined to equal an average rate of switching packets from said first to said second one network interface, said average being true for a packet traffic distribution that is not predetermined, and said average being sustainable over a substantial period of time;

said clock cycle rate divided by said packet switching rate being less than about 100 clock cycles per packet switched.

26. A device as in claim 25, wherein said clock cycle time is not less than about 30 nanoseconds and said packet switching rate is greater than about 300,000 packets per second.

27. A device as in claim 25, wherein said packet traffic distribution is a normal distribution for packets being switched on said first one network interface.

28. A device for switching packets, comprising

means for receiving information from a network interface coupled to a network, said information comprising destination addresses;

means for converting said information to tree programs for a tree memory; and

a tree memory for executing said tree programs.

29. A device as in claim 28, comprising means for placing said tree program in a tree memory.

30. A device as in claim 28, comprising means for triggering said means for generating, responsive to a timer.

31. A device as in claim 28, comprising means, responsive to said information, for triggering said means for generating.

32. A device as in claim 28, wherein said tree memory comprises

a comparator having a first input coupled to said first memory and having a second input;

a second memory having a first input coupled to a comparison output of said comparator, and having a second input, said first and second inputs collectively referencing a location in said second memory;

at least part of said location comprising a next data word and being coupled to said second input of said comparator;

at least part of said location comprising a next address and being coupled to said second input of said memory; and

at least part of said location comprising a next instruction and being coupled to an instruction decoder.

33. A device as in claim 28, wherein said means for converting comprises

means for generating a tree program for recognizing a set of destination addresses in said information;

means for placing said tree program in a tree memory for execution.

34. A device as in claim 28, wherein said means for converting comprises

means for generating a weighted tree of destination addresses; and

means for generating a tree program responsive to said weighted tree, wherein said tree program comprises at least one call upon a high-level processor for processing a packet, and wherein said tree program is limited to a predetermined size, and wherein said tree program is structured to have a minimum likelihood per packet of executing said call.

35. A method of packet switching, comprising

coupling a data word from a packet received from a first one of a plurality of network interfaces to a first input of a comparator;

addressing a memory in response to an output of said comparator;

retrieving an output of said memory;

coupling at least part of said output to a second input of said comparator;

coupling at least part of said output to an address input of said memory;

coupling at least part of said output to an instruction decoder, said instruction decoder being coupled to a processing element;

repeating said steps at least until said processing element prepares a result data word indicative of a second one of said plurality of network interfaces, and said instruction decoder recognizes a part of said output as indicative of readiness to switch said packet; and

sending said packet to said second one of said plurality of network interfaces.

36. A method for switching packets, comprising

receiving a packet from a first one of a plurality of network interfaces;

performing a plurality of tree memory operations, each said tree memory operation comprising simultaneously (a) retrieving a first data word from said packet, (b) comparing a second data word from said packet with a test data word, (c) executing a processor instruction in response to a prior tree memory operation, and (d) selecting a next tree memory operation in response to said prior tree memory operation;

at least one said step of executing comprising preparing a result data word indicative of a second one of said plurality of network interfaces; and

sending said packet to said second one of said plurality of network interfaces.

AMENDED CLAIMS

[received by the International Bureau on 17 October 1995 (17.10.95);
original claims 1,2,8,10, 14,17,18,20,21,24,30-32 and 35 amended;
remaining claims unchanged (11 pages)]

1. A device for switching packets, comprising

a first memory coupled to a network interface, said first memory being large enough to hold a packet data word;

a comparator having a first input coupled to said first memory and having a second input;

a second memory having a first input coupled to a comparison output of said comparator, and having a second input, said first and second inputs collectively referencing a location in said second memory;

at least part of said location comprising a next data word and being coupled to said second input of said comparator;

at least part of said location comprising a next address and being coupled to said second input of said second memory; and

at least part of said location comprising a next instruction word, said next instruction word being coupled to an instruction decoder.

2. A device as in claim 1, comprising

a set of counters, wherein a counter comprises means for decrementing upon reading a data word of said packet;

means for, when said counter has not reached a predetermined counter value, coupling said next address to said second memory, and when said counter reaches said predetermined counter value, coupling a selected address to said second memory.

3. A device as in claim 1, comprising

a set of feedback registers coupled to an output of said second memory and to said instruction decoder.

4. A device as in claim 1, comprising

a set of reorder registers coupled to said first memory and to said instruction decoder.

5. A device as in claim 1, comprising

a set of result registers coupled to said second memory and to said instruction decoder; and

a rewrite engine coupled to said first memory and to said set of result registers.

6. A device as in claim 1, wherein a packet having said packet data word may comprise one of a plurality of packet transmission protocols.

7. A device as in claim 1, wherein

said comparison output comprises a plurality of output signals;

said second memory comprises a plurality of memory sections, each coupled to at least one of said plurality of output signals, whereby exactly one of said plurality of memory sections is referenced by said plurality of output signals.

8. A device as in claim 1, wherein said instruction decoder comprises

AMENDED SHEET (ARTICLE 19)

a next word circuit coupled to at least part of said next instruction word;

a checksum bit circuit coupled to at least part of said next instruction word, said checksum bit circuit being coupled to a checksum device;

an opcode circuit coupled to at least part of said next instruction word, to a done bit, and to said checksum bit;

said opcode circuit configured to recognize a first instruction for setting said done bit to a first predetermined value; and

said opcode circuit configured to recognize a second instruction for setting said checksum bit to a second predetermined value, whereby said checksum device operates in response to said second instruction.

9. A device as in claim 8, wherein said opcode circuit is configured to recognize a third instruction for testing an output of said checksum device.

10. A device as in claim 1, wherein said instruction decoder comprises

an opcode circuit coupled to at least part of said next instruction word, to a counter, and to a return location register;

said opcode circuit configured to recognize a CALL instruction for calling a subroutine, and responsive to said CALL instruction by placing a value in said return location register;

a circuit coupled to said counter and configured to recognize a predetermined value held therein, and configured to retrieve a value from said return location register and to forcing a predetermined result from said comparator in response thereto.

11. A device as in claim 10, wherein said counter is configured to change state each time a packet data word is read from said first memory.

12. A device as in claim 10, wherein said counter is configured to change state each time a packet data word is read from said first memory, responsive to an enabling circuit, and wherein said opcode circuit is coupled to said enabling circuit and configured to put said enabling circuit in a predetermined state in response to an instruction.

13. A device as in claim 10, wherein said opcode circuit is configured to recognize a RETURN instruction for returning from a subroutine, and responsive to said RETURN instruction by retrieving a value from said return location register and by forcing a predetermined result from said comparator.

14. A device as in claim 1, wherein said instruction decoder comprises

an opcode circuit coupled to at least part of said next instruction word, to a memory page register, and to a third memory having a plurality of sets of addressable reorder registers and a plurality of sets of addressable feedback registers;

said memory page register comprising a first circuit indicating a choice between said reorder registers and said feedback registers;

said memory page register comprising a second circuit indicating a choice of one of said plurality of sets of reorder registers and one of said plurality of sets of feedback registers; and

said opcode circuit configured to recognize a first set of instructions, each for addressing and altering one of said reorder registers, and a second set of instructions, each for addressing and altering one of said feedback registers.

15. A device as in claim 14, wherein said opcode circuit is configured to alter said memory page register in response to an instruction.

16. A device as in claim 7, wherein

said plurality of output signals comprise a less than signal, an equal to signal, and a greater than signal; and

said plurality of memory sections comprises a section activated by said less than signal, a section activated by said equal to signal, and a section activated by said greater than signal.

17. A device as in claim 7, wherein said second memory comprises a location in each one of said plurality of memory sections for each address coupled to said second memory.

18. A device for switching packets, comprising

means for receiving a packet from a first one network interface of a plurality of network interfaces;

a tree memory comprising a set of locations each having a next data word, a next address and a next instruction word, said set of locations comprising a first region comprising a tree program for routing packets in response to a set of static routing information about a network coupled to said first one network interface;

means for receiving dynamic routing information about said network;

means for compiling said dynamic routing information into a second region in said set of locations; and means for sending said packet to a second one of said plurality of network interfaces in response to said tree memory.

19. A device as in claim 18, comprising

means for identifying routing information in said packet in response to said tree memory; and

means for directing said means for sending to switch said packet in response to said means for identifying.

20. A device as in claim 19, comprising

means for receiving dynamic routing information about a network, said network being coupled to said first one network interface;

means for compiling said dynamic routing information into a region in said tree memory.

21. A device as in claim 20, said tree memory comprising static routing information about said network.

22. A device as in claim 18, wherein said dynamic routing information comprises information about locations of devices coupled to said network or information about access control for devices coupled to said network.

23. A device as in claim 18, wherein said static routing information comprises information about a protocol used on said network.

AMENDED SHEET (ARTICLE 19)

24. A device for switching packets, comprising

means for receiving a packet from a first one of a plurality of network interfaces;

means for preparing an interface register in response to said packet;

a tree memory having a set of locations each having a next data word, a next address and a next instruction word;

an instruction decoder coupled to said next instruction word and to a result register;

means for signaling said tree memory to process said packet;

means for altering said packet in response to said result register;

means for selecting a second one of said plurality of network interfaces in response to said result register;

means for sending said packet to said second one network interface.

25. A device for switching packets, comprising

means for receiving a packet from a first one of a plurality of network interfaces;

means for sending said packet to a second one of said plurality of network interfaces;

means for switching said packet from said first one network interface to said second one network interface;

said means for switching having a clock cycle time defined to equal a shortest time needed to decode a processor instruction, and having a clock cycle rate defined to equal an inverse of said clock cycle time;

said means for switching having a packet switching rate defined to equal an average rate of switching packets from said first to said second one network interface, said average being true for a packet traffic distribution that is not predetermined, and said average being sustainable over a substantial period of time;

said clock cycle rate divided by said packet switching rate being less than about 100 clock cycles per packet switched.

26. A device as in claim 25, wherein said clock cycle time is not less than about 30 nanoseconds and said packet switching rate is greater than about 300,000 packets per second.

27. A device as in claim 25, wherein said packet traffic distribution is a normal distribution for packets being switched on said first one network interface.

28. A device for switching packets, comprising

means for receiving information from a network interface coupled to a network, said information comprising destination addresses;

means for converting said information to tree programs for a tree memory, said tree memory comprising a set of registers disposed in a tree structure and said tree programs comprises a set of instructions disposed in said tree structure and having a comparison and branch at a plurality of locations thereof; and

means for executing said tree programs.

29. A device as in claim 28, comprising, means for placing said tree programs in said tree memories.

30. A device as in claim 28, comprising means for triggering said means for converting said information to tree programs for a tree memory, responsive to a timer.

31. A device as in claim 28, comprising means, responsive to said information, for triggering said means for converting said information to tree programs for a tree memory.

32. A device as in claim 28, wherein said tree memory comprises

a comparator having a first input coupled to said tree memory and having a second input;

a second memory having a first input coupled to a comparison output of said comparator, and having a second input, said first and second inputs collectively referencing a location in said second memory;

at least part of said location comprising a next data word and being coupled to said second input of said comparator;

at least part of said location comprising a next address and being coupled to said second input of said memory; and

at least part of said location comprising a next instruction word and being coupled to an instruction decoder.

33. A device as in claim 28, wherein said means for converting comprises

means for generating a tree program for recognizing a set of destination addresses in said information;

means for placing said tree program in a tree memory for execution.

34. A device as in claim 28, wherein said means for converting comprises means for generating a weighted tree of destination addresses; and

means for generating a tree program responsive to said weighted tree, wherein said tree program comprises at least one call upon a high-level processor for processing a packet, and wherein said tree program is limited to a predetermined size, and wherein said tree program is structured to have a minimum likelihood per packet of executing said call.

35. A method of packet switching, comprising

coupling a data word from a packet received from a first one of a plurality of network interfaces to a first input of a comparator;

addressing a memory in response to an output of said comparator;

retrieving an output of said memory;

coupling at least part of said output of said memory to a second input of said comparator;

coupling at least part of said output of said memory to an address input of said memory;

coupling at least part of said output of said memory to an instruction decoder, said instruction decoder being coupled to a processing element;

repeating said steps of coupling a data word, addressing, retrieving, coupling to a second input, coupling to an address input, and coupling to a processing element, at least until said processing element prepares a result data word indicative of a second one of said plurality of network interfaces, and said instruction decoder recognizes a part of said output as indicative of readiness to switch said packet; and sending said packet to said second one of said plurality of network interfaces.

36. A method for switching packets, comprising

receiving a packet from a first one of a plurality of network interfaces;

performing a plurality of tree memory operations, each said tree memory operation comprising simultaneously (a) retrieving a first data word from said packet, (b) comparing a second data word from said packet with a test data word, (c) executing a processor instruction in response to a prior tree memory operation, and (d) selecting a next tree memory operation in response to said prior tree memory operation;

at least one said step of executing a processor instruction comprising preparing a result data word indicative of a second one of said plurality of network interfaces; and

sending said packet to said second one of said plurality of network interfaces.

Statement Under Article 19

The inventions of claims 28-31 provide specialized apparatus capable of switching packets at high speed. For example, in one preferred embodiment, the information for switching received from a network interface comprises information indicating how to distinguish the output port to which to route a packet in response to early bytes of the packet header. That information is compiled into one or more tree programs--for example, tree programs recognizing those early bytes and indicating the proper output port as soon as possible. Those tree programs are then executed by the switching engine until updated information is received.

US A 5,311,509 (Heddes), cited in the search report only as relevant to claims 28-31, shows a method for transforming messages from user frames, to fixed-length cells, and back to user frames, so that the fixed-length cells can be switched. User frames are stored in buffers, from which fixed-length cells are read and prepended with header information. Col. 4, lines 9-21. Fixed-length cells are processed on-the-fly, and the parameters in the cell headers are extracted and presented to the header processor. Col. 4, lines 48-51. The header processor, in response to the header information in the fixed-length cells, generates and manages buffers in a set of FIFO stacks (figure 10).

Although Heddes does generate buffers in response to header information, Heddes does not generate "tree programs", i.e., programs for a decision tree memory, as defined in the specification. The tree memory recited in claim 28 comprises a set of registers disposed in a tree structure; the tree programs recited in claim 28 comprise a set of instructions disposed in the tree structure and having a comparison and branch at a plurality of locations thereof. Heddes merely allocates a set of buffers, each of which specifies a fixed-length cell header in full.

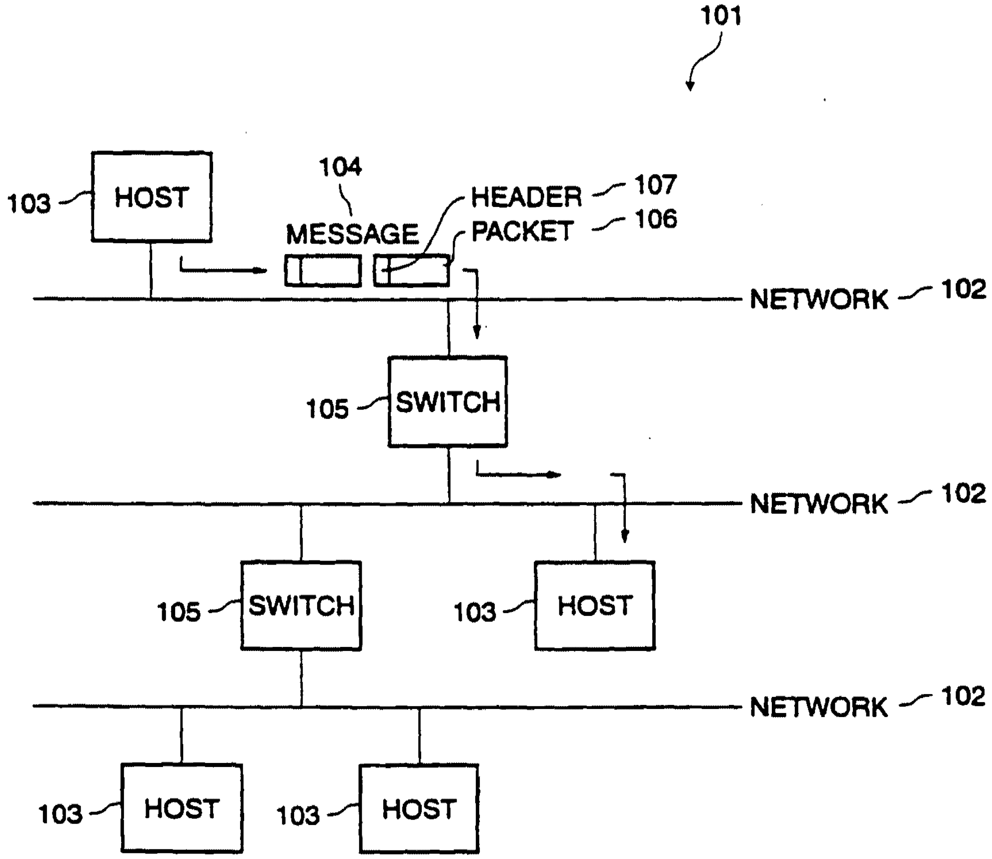

101
103
HOST
104
MESSAGE
HEADER
107
PACKET
106
NETWORK
102
105
SWITCH
NETWORK
102
105
SWITCH
103
HOST
NETWORK
102
103
HOST
103
HOST

FIG. 1

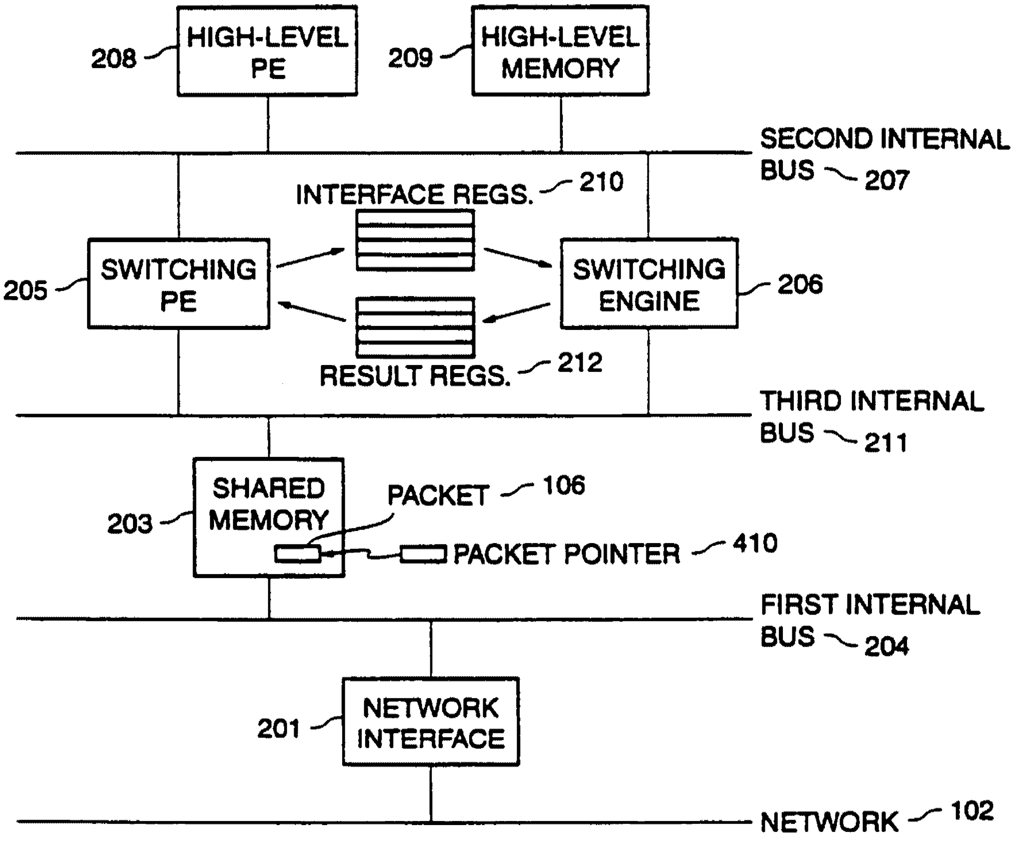

208
HIGH-LEVEL PE
209
HIGH-LEVEL MEMORY
SECOND INTERNAL BUS 207
INTERFACE REGS. 210
205
SWITCHING PE
SWITCHING ENGINE
206
RESULT REGS. 212
THIRD INTERNAL BUS 211
203
SHARED MEMORY
PACKET 106
PACKET POINTER 410
FIRST INTERNAL BUS 204
201
NETWORK INTERFACE
NETWORK 102

FIG. 2

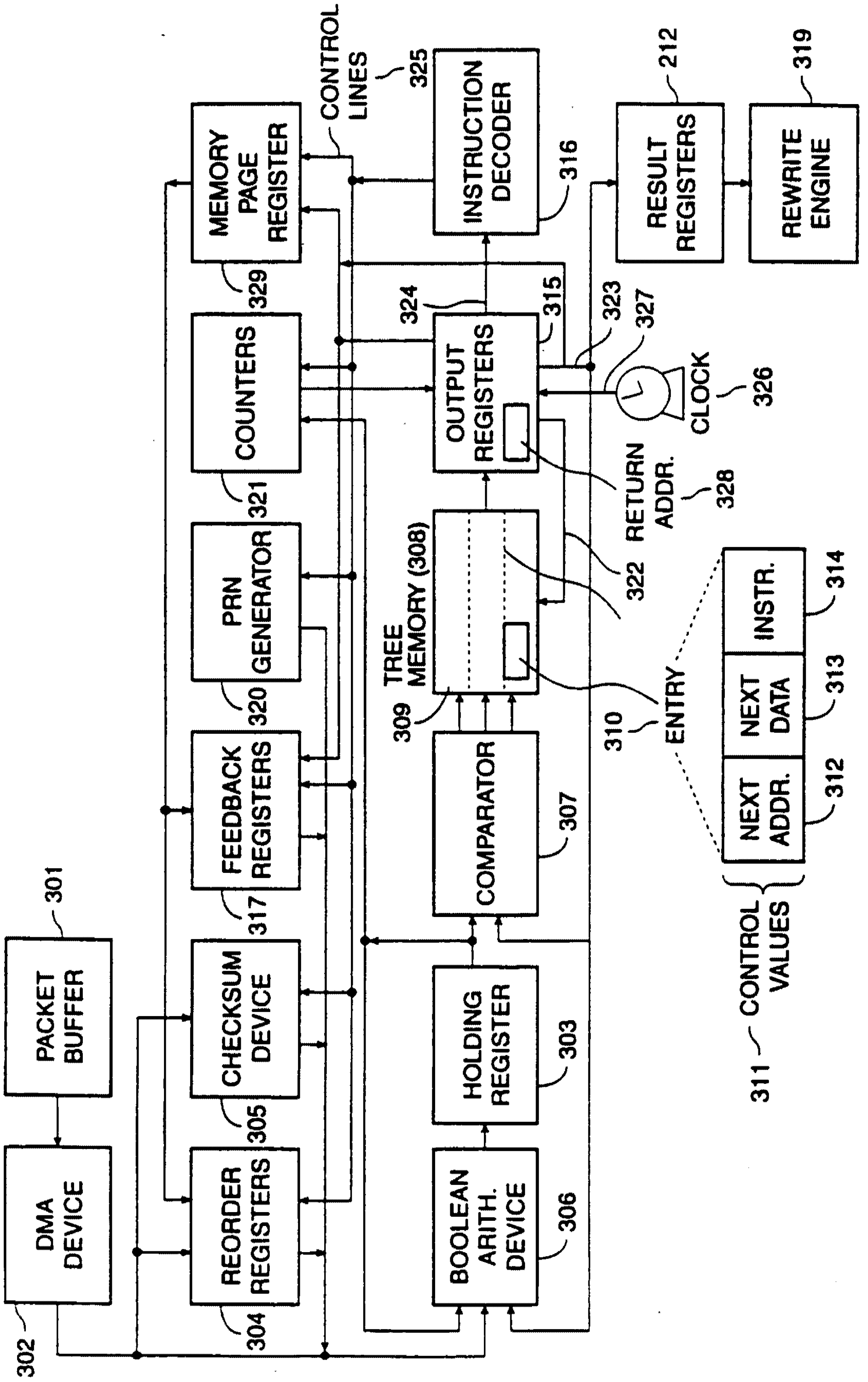

PACKET BUFFER
301
DMA DEVICE
302
REORDER REGISTERS
304
CHECKSUM DEVICE
305
FEEDBACK REGISTERS
317
PRN GENERATOR
320
COUNTERS
321
MEMORY PAGE REGISTER
329
CONTROL LINES
325
INSTRUCTION DECODER
316
OUTPUT REGISTERS
315
324
323
327
CLOCK
326
RETURN ADDR.
328
TREE MEMORY (308)
309
322
ENTRY
310
COMPARATOR
307
HOLDING REGISTER
303
BOOLEAN ARITH. DEVICE
306
RESULT REGISTERS
212
REWRITE ENGINE
319
NEXT ADDR.
312
NEXT DATA
313
INSTR.
314
CONTROL VALUES
311

FIG. 3

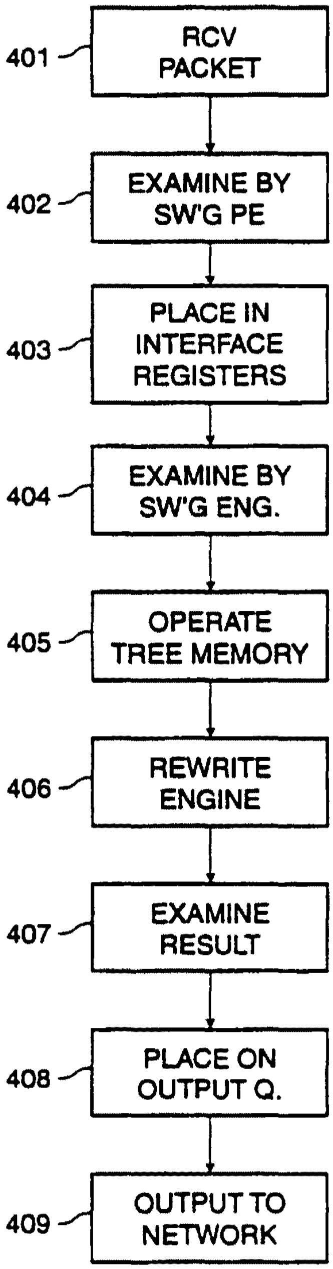

401
RCV PACKET
402
EXAMINE BY SW'G PE
403
PLACE IN INTERFACE REGISTERS
404
EXAMINE BY SW'G ENG.
405
OPERATE TREE MEMORY
406
REWRITE ENGINE
407
EXAMINE RESULT
408
PLACE ON OUTPUT Q.
409
OUTPUT TO NETWORK

FIG. 4

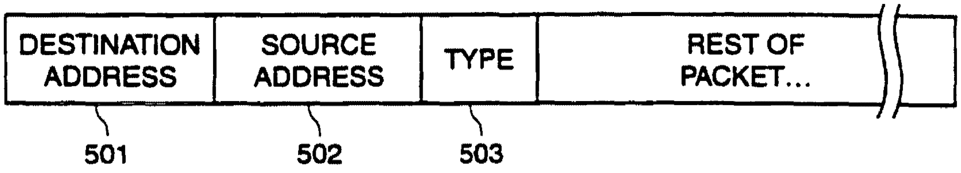

DESTINATION ADDRESS
SOURCE ADDRESS
TYPE
REST OF PACKET...
501
502
503

FIG. 5A

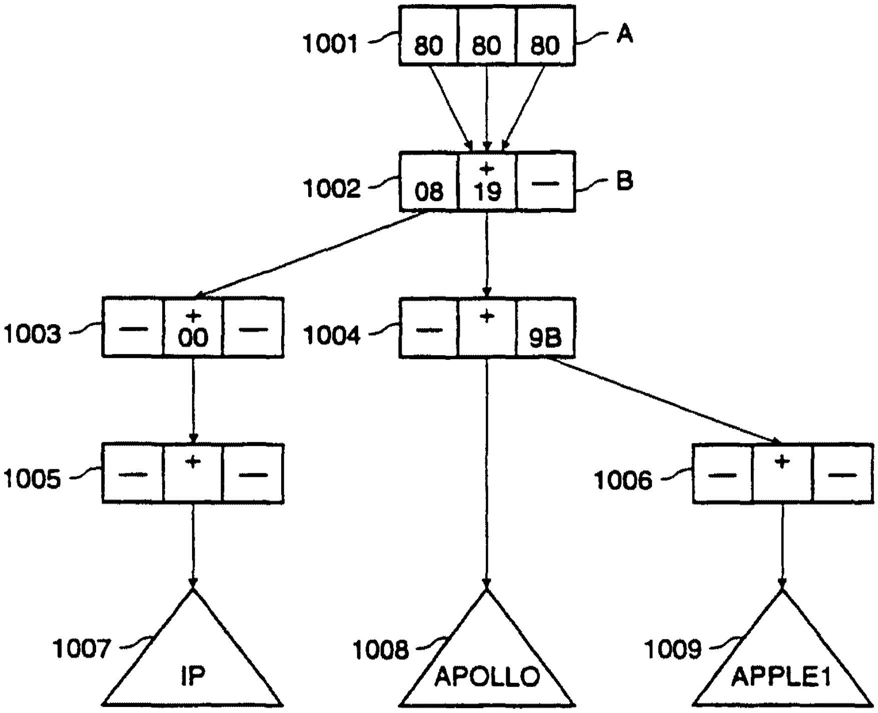

1001
80
80
80
A
1002
08
+
19
—
B
1003
—
+
00
—
1004
—
+
9B
1005
—
+
—
1006
—
+
—
1007
IP
1008
APOLLO
1009
APPLE1

FIG. 5B

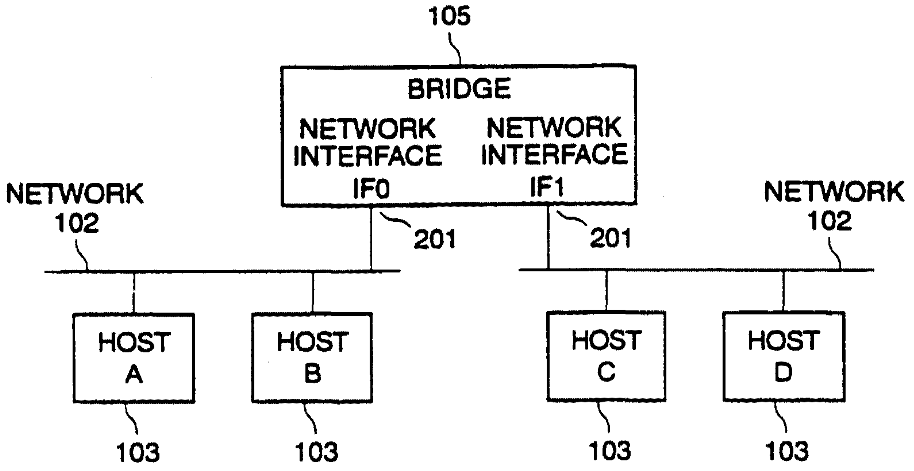

105
BRIDGE
NETWORK INTERFACE IF0
NETWORK INTERFACE IF1
NETWORK 102
201
201
NETWORK 102
HOST A
HOST B
HOST C
HOST D
103
103
103
103

FIG. 6A

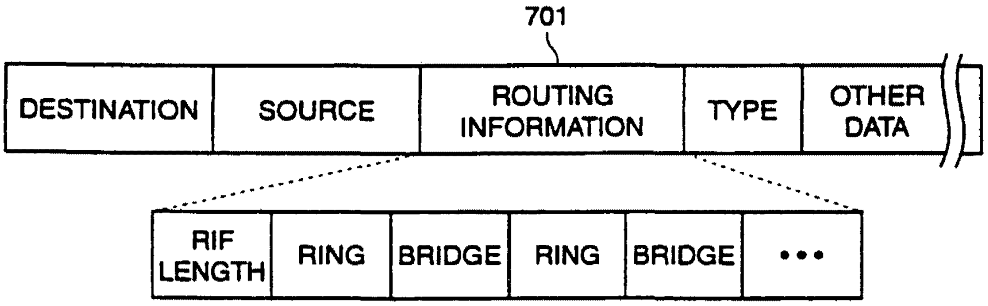

701
DESTINATION
SOURCE
ROUTING INFORMATION
TYPE
OTHER DATA
RIF LENGTH
RING
BRIDGE
RING
BRIDGE
•••

FIG. 7A

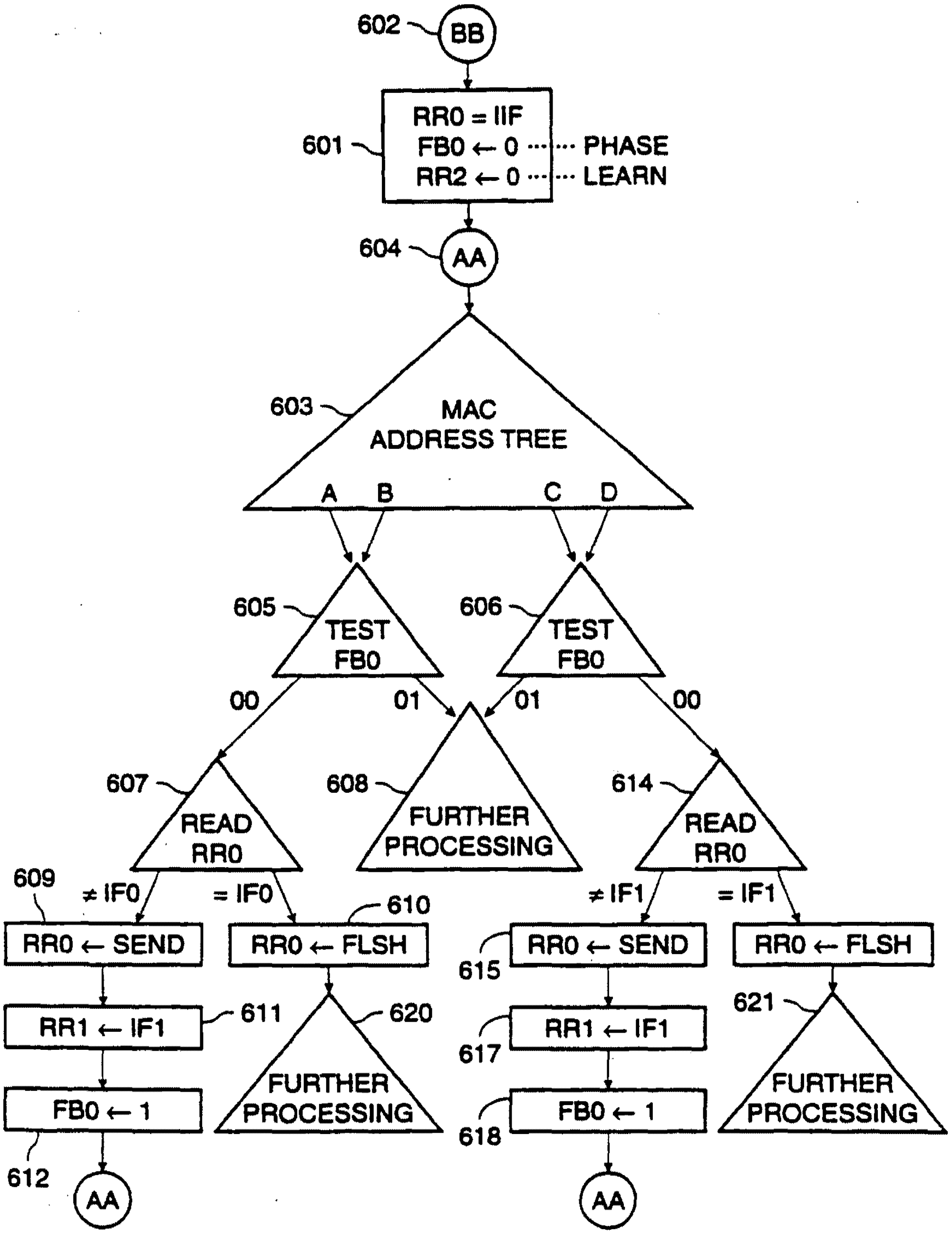
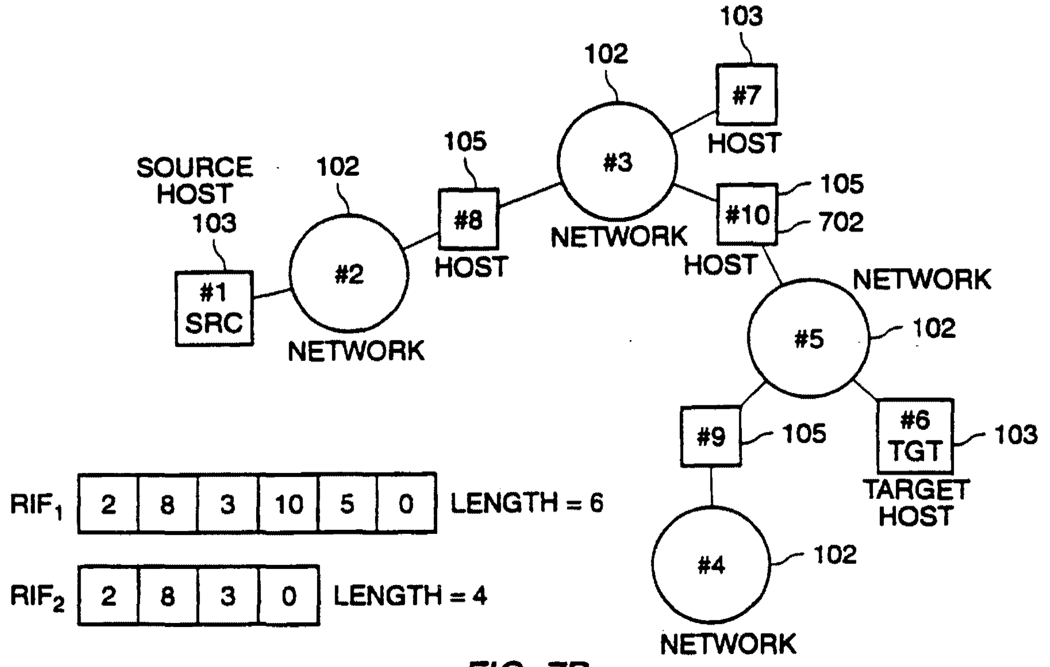

602
BB
601
RR0 = IIF
FB0 ← 0
RR2 ← 0
PHASE
LEARN
604
AA
603
MAC ADDRESS TREE
A
B
C
D
605
TEST FB0
606
TEST FB0
00
01
01
00
607
READ RR0
608
FURTHER PROCESSING
614
READ RR0
609
≠ IF0
= IF0
610
≠ IF1
= IF1
RR0 ← SEND
RR0 ← FLSH
615
RR0 ← SEND
RR0 ← FLSH
RR1 ← IF1
611
620
617
RR1 ← IF1
621
FB0 ← 1
FURTHER PROCESSING
618
FB0 ← 1
FURTHER PROCESSING
612
AA
AA

FIG. 6B



RIF_1	2	8	3	10	5	0

LENGTH = 6

RIF_2	2	8	3	0

LENGTH = 4

FIG. 7B

751 ACCESS CONTROL LIST

752 753 754

TYPE	PERMISSIONS	SET OF HOST ADDRESSES
"1"	"PERMIT"	"160.89.32.1"
"101"	"PERMIT IP", "DENY TCP"	"131.108.0.0", "0.0.255.2"
⋮		

FIG. 7C

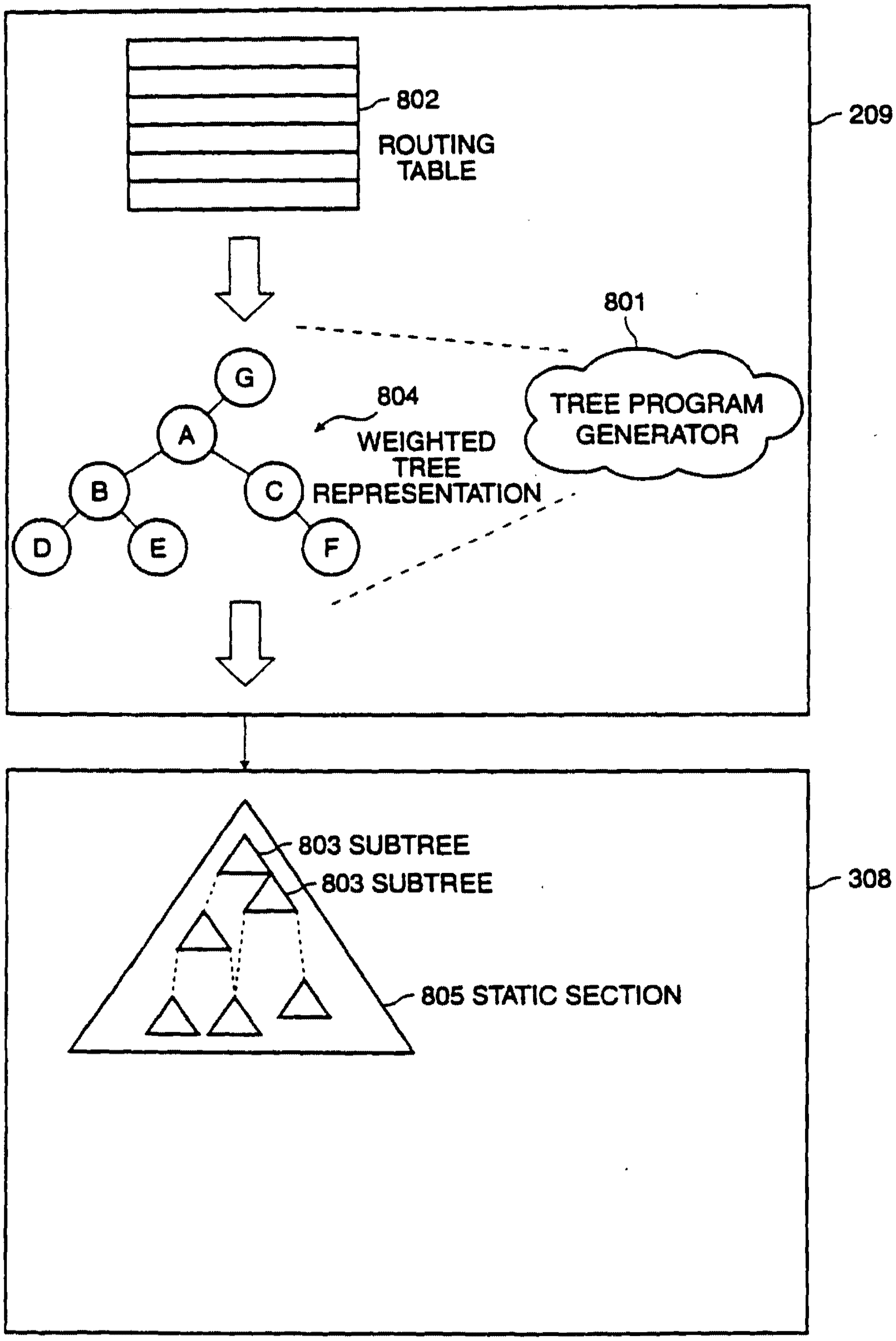

802
ROUTING TABLE
209
801
TREE PROGRAM GENERATOR
G
A
B
C
D
E
F
804
WEIGHTED TREE REPRESENTATION
803 SUBTREE
803 SUBTREE
308
805 STATIC SECTION

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/05444

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 12/56
US CL : 370/060.000, 094.100
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/060.000, 094.100,060.100,058.100,058.200,058.300

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

NONE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 5,311,509 (HEDDES ET AL.) 10 May 1994, columns 1-6.	28-31

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

03 JULY 1995

Date of mailing of the international search report

18 AUG 1995

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer
DANG TON

Telephone No. (703) 305-4739

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04Q

A2

(11) International Publication Number: WO 99/48303

(43) International Publication Date: 23 September 1999 (23.09.99)

(21) International Application Number: PCT/US99/05900

(22) International Filing Date: 18 March 1999 (18.03.99)

(30) Priority Data:
09/040,898 18 March 1998 (18.03.98) US

(71) Applicant: CISCO TECHNOLOGY, INC. [US/US]; 170 West Tasman Drive, San Jose, CA 95134 (US).

(72) Inventors: COX, Dennis; 6800 McNeil Drive #828, Austin, TX 78729 (US). MCCLANAHAN, Kip; 3112 Kerbey Lane, Austin, TX 78703 (US).

(74) Agent: SHOWALTER, Barton, E.; Baker & Botts, L.L.P., 2001 Ross Avenue, Dallas, TX 75201-2980 (US).

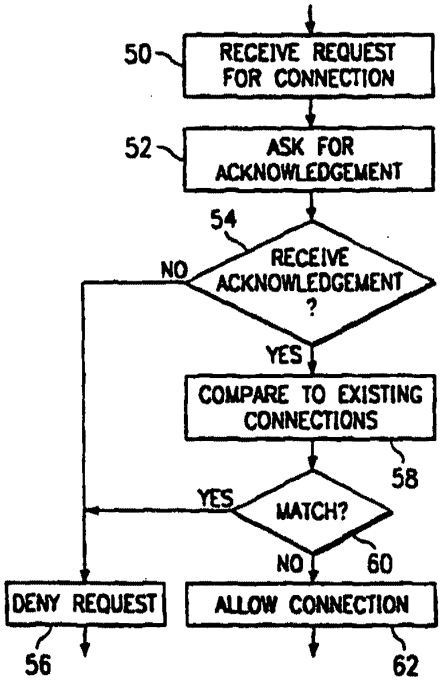
(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD FOR BLOCKING DENIAL OF SERVICE AND ADDRESS SPOOFING ATTACKS ON A PRIVATE NETWORK

(57) Abstract

A method is provided for blocking attacks on a private network (12). The method is implemented by a routing device (10) interconnecting the private network (12) to a public network (14). The method includes analyzing an incoming data packet from the public network (14). The incoming data packet is then matched against known patterns where the known patterns are associated with known forms of attack on the private network (12). A source of the data packet is then identified as malicious or non-malicious based upon the matching. In one embodiment, one of the known forms of attack is a denial of service attack and an associated known pattern in unacknowledged data packets. In another embodiment, one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network (12).


50
RECEIVE REQUEST FOR CONNECTION
52
ASK FOR ACKNOWLEDGEMENT
54
RECEIVE ACKNOWLEDGEMENT ?
NO
YES
COMPARE TO EXISTING CONNECTIONS
58
MATCH?
YES
NO
60
DENY REQUEST
ALLOW CONNECTION
56
62

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD FOR BLOCKING DENIAL OF SERVICE AND ADDRESS SPOOFING ATTACKS ON A PRIVATE NETWORK

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to communication systems, and more particularly to a method for blocking denial of service and address spoofing attacks on a private network.

BACKGROUND OF THE INVENTION

Corporate and other private networks often provide external access outward and inward through Internet gateways, firewalls or other routing devices. It is important for these routing devices to defend the private network against attackers from the outside as well as to allow access to the private network by authorized users. However there are numerous forms of attack on conventional routing device that can incapacitate the devices and interfere with an associated private network. The problem of keeping unauthorized persons from accessing data is a large problem for corporate and other information service management. Routing devices, such as gateways, firewalls and network routers lack important safeguards to block or prevent attacks. In particular, the number of denial service attacks have risen dramatically in recent years. Further, IP spoofing incidents occur with increasing frequency.

A denial of service attack consists of repeatedly sending requests for connections to different hosts through and/or behind the routing device. Typically, the host will wait for acknowledgment from the requester.

Because a host can only handle a finite number of requests (for example, 1 to n, where n depends on the resources available to the host), the attacker can crash or "flood" a host with requests to the point of disrupting network service (host/server/port) to users.

Another form of attack is address spoofing which can be used by unauthorized third parties to gain access to a private network. This attack involves the attacker identifying a valid internal network address within the private network. The attacker then requests access to the private network through the routing device by spoofing that internal network address. Conventional routing devices typically are not sophisticated enough to determine that such a request should be denied (i.e., because an external request can not originate from an internal address) and will allow access to the attacker. Address spoofing attacks can be carried out against various types of networks and network protocols such as IPX/SPX, MAC layer, Netbios, and IP.

It is therefore advantageous to provide facilities within a routing device that block denial of service, address spoofing and other attacks on an associated private network.

SUMMARY OF THE INVENTION

In accordance with the present invention, a method for blocking denial of service and address spoofing attacks on a private network is disclosed that provides significant advantages over conventional network routing devices.

According to one aspect of the present invention, the method is implemented by a routing device interconnecting the private network to a public network. The method includes analyzing an incoming data packet from the public network. The incoming data packet is then matched against known patterns where the known patterns are associated with known forms of attack on the private network. A source of the data packet is then identified as malicious or non-malicious based upon the matching. In one embodiment, one of the known forms of attack is a denial of service attack and an associated known pattern is unacknowledged data packets. In another embodiment, one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network.

A technical advantage of the present invention is the enabling of a routing device to the identify a denial of service attack and to block such an attack from tying up the routing device.

Another technical advantage of the present invention is enabling a routing device to identify an address spoofing attack and to block such an attack.

A further technical advantage of the present invention is an ability for the routing device to track information about the attacker to allow preventive measures to be taken.

Other technical advantages should be readily apparent to one skilled in the art from the following figures, description, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, in which like reference numbers indicate like features, and wherein:

FIGURE 1 is a block diagram of an communication system including a routing device and an associated private network;

FIGURE 2 is a flow chart of one embodiment of a method for blocking attacks on a private network according to the present invention;

FIGURE 3 is a flow chart of one embodiment of a method for blocking an address spoofing attack according to the present invention; and

FIGURE 4 is a flow chart of one embodiment of a method for blocking a denial of service attack according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a block diagram of an communication system including a routing device 10 and an associated private network 12. Routing device 10 provides a connection between corporate private network 12 and an Internet cloud 14. Routing device 10 can include a gateway, firewall or other device interconnecting private network 12 and Internet cloud 14. In operation, routing device 10 allows internal users within private network 12 to gain access to Internet cloud 14. Routing device 10 also allows external users connected to Internet cloud 14 to gain access to private network 12. A significant and growing problem is that an attacker 16 may try to gain access to or disrupt private network 12 through Internet cloud 14.

Denial of service and address spoofing are two common forms of attack that might be used by attacker 16. In general, a denial service attack is one in which attacker 16 attempts to prevent others from using private network 12. A denial service attack works if routing device 10 spends all of its time processing requests and cannot respond quickly enough to satisfy additional requests. An Address spoofing attack is on in which attacker 16 fakes an internal address to get around or into standard address filtering schemes. According to the present invention, routing device 10 is enabled with a method for blocking these and other types of attacks by analyzing incoming data packets.

Thus, one possible occurrence is that attacker 16 will try to get into private network 12 by spoofing an address that exists inside private network 12. This is intended to allow attacker 16 to gain access and impersonate an internal user. When a packet from attacker 16 reaches routing device 12, an attack blocking component, according to the present invention, will notice that the address matches one that exists within private network 12. Because incoming packets should not be the same as outgoing packets, the attack blocking component can deny access to private network 12 and record the information about the attack for use by the system administrator. Attacker 16 can also try to deny access to all external users by conducting a denial of service attack. This involves attacker 16 flooding private network 12 or routing device 10 by sending an extremely large number of packets. For example, attacker 16 may send 30,000 or more packets. According to the present invention, the attack blocking component of routing device 10 can notice that the first packet is spoofed or that it cannot be acknowledged and ignore all other packets. Further, routing device 10 can use diagnostic detection tools (e.g., trace root, ping, NS lookup) to pinpoint attacker 16 and notify the system administrator. In general, according to the present invention, routing device 10 can be enabled to intelligently analyze incoming packets, match the packets against known patterns for attack strategies and respond accordingly to malicious packets.

FIGURE 2 is a flow chart of one embodiment of a method for blocking attacks on a private network according to the present invention. As shown, an incoming packet is analyzed by the routing device in step 20. In step 22, the routing device analyzes the incoming packet against known patterns. Based upon this pattern matching, in step 24, the routing device can identify the data packet and its source as malicious or non-malicious. The known patterns used in step 22 can be built using knowledge about various types of attacks. This knowledge can be recorded in the form of patterns that are then stored in a database or other storage device accessible by the routing device. The routing device can then match the analyzed packets against the patterns to determine whether or not some type of attack is being made. If an attack is identified, the routing device can identify the source of that packet as malicious and treat the source [illegible]

[illegible]

[illegible]

internal IP address of the private network cannot be accessing the private network from an external point. Consequently, in step 38, the routing device drops the packet and does not route it to the network. In step 40, the routing device analyzes the packet header for the history of the packet in order to obtain some information about the source of the packet. Then, in step 42, the routing device takes an appropriate defensive action against that packet. For example, the routing device can refuse to accept any more packets from the real source of the packet. In this case, the defensive action can include adding the offending IP address to a cache of IP addresses and then not allowing access to the router device for any IP address in the cached list. Further, the routing device can store information about the attack for later use and for analysis for administrators of the private network. For example, information concerning the packet origination, destination or content can be stored internally to the router device or sent to a syslog server for later analysis.

FIGURE 4 is a flow chart of one embodiment of a method for blocking a denial of service attack according to the present invention. As shown, in step 50, the routing device receives a request for a connection. Then, in step 52, the routing device asks for an acknowledgment from the requestor. In step 54, the routing device checks whether or not an acknowledgment has been received. If one is not received within a specified period of time, the routing device moves to step 56 and denies the request. This denial ensures that the routing device does not churn on pending requests even though acknowledgments have not been received within reasonable amounts of time.

If an acknowledgment is received in step 54, the routing device moves to step 58 and compares the requested connection to existing connections. Then, in step 60, the routing device determines if there is a match between the requested connection and one of the existing connections. If so, the routing device moves to step 46 and denies the request. The request is denied because one source should not have more than one connection through the routing device to the private network. If, in step 60, there is no match, then the routing device can allow the connection in step 62. The method of FIGURE 4 prevents the routing device from being tied up by multiple requests from one source and thereby blocks the denial of service attack.

In general, the method of the present invention can be integrated as a component of a gateway, firewall or other routing device. In one implementation, the present invention can work off of a variable size cache file that holds network addresses. For blocking spoofing, each incoming address can be held in the cache file and checked to see if the incoming address matches an network address that is on the private network. If the incoming address matches, then the request can be denied. Also, a message can be sent to a system log which, rather than being written to a file, can be written to a console to prevent the log from getting overloaded and crashing the routing device. Further, an optional E-mail message or page can be sent to a specified address or number in the case of an attack. If an attack happens more than once on the same address in the span of a certain period of time (for example, five minutes), then the number of messages can be limited to prevent overloading of the E-mail or paging service. An optional shutdown mechanism can also be in place that will enable the routing device to automatically shut down certain services if attacks continued.

Denial of service attacks are generally easier to trace. However, when such an attack is also spoofed, the problem becomes very difficult to stop. According to the present invention, an incoming address can be checked against the cache file and a quick search can be performed to see if the address is already in a list of pending addresses. If so, the request packet can be discarded. An address is removed from the list if a successful acknowledge packet is sent back or a variable time limit is reached. The number of matching addresses that are allowed in the list can be a variable set by the system administrator.

Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made thereto without departing from the sphere and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method for blocking attacks on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

analyzing an incoming data packet from the public network;

matching the incoming data packet against known patterns, the known patterns associated with known forms of attack on the private network; and

identifying a source of the data packet as malicious or non-malicious based upon the matching.

2. The method of Claim 1, wherein one of the known forms of attack is a denial of service attack and an associated known pattern is unacknowledged data packets.

3. The method of Claim 1, wherein one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network.

4. The method of Claim 1, wherein the public network is the Internet.

5. The method of Claim 4, wherein the routing device is a firewall providing access to the Internet.

6. A method for blocking an address spoofing attack on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

receiving an incoming data packet from the public network;

comparing a source address of the data packet against known internal addresses of the private network;

determining if the source address matches a known internal address;

if there is no match, routing the data packet to the private network;

if there is a match, dropping the data packet.

12. A method for blocking a denial of service attack on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

receiving a request for a connection from the public network;

requesting an acknowledgment from an initiator of the request;

determining whether an acknowledgment has been received;

if an acknowledgment is not received, denying the request;

if an acknowledgment is received, comparing the request to existing connections;

if there is a match between the request and an existing connection, denying the request;

if there is not match between the request and an existing connection, allowing the connection and routing packets to the private network.

13. The method of Claim 12, wherein the public network is the Internet.

14. The method of Claim 13, wherein the routing device is a firewall providing access to the Internet.

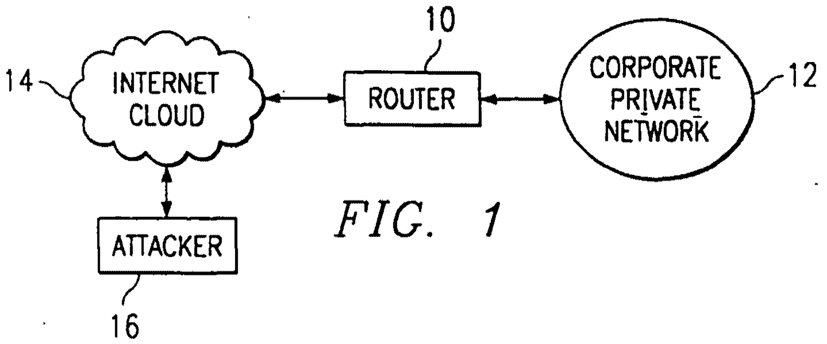


FIG. 1

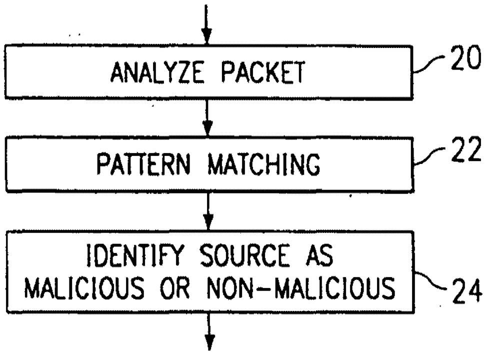


FIG. 2

FIG. 3

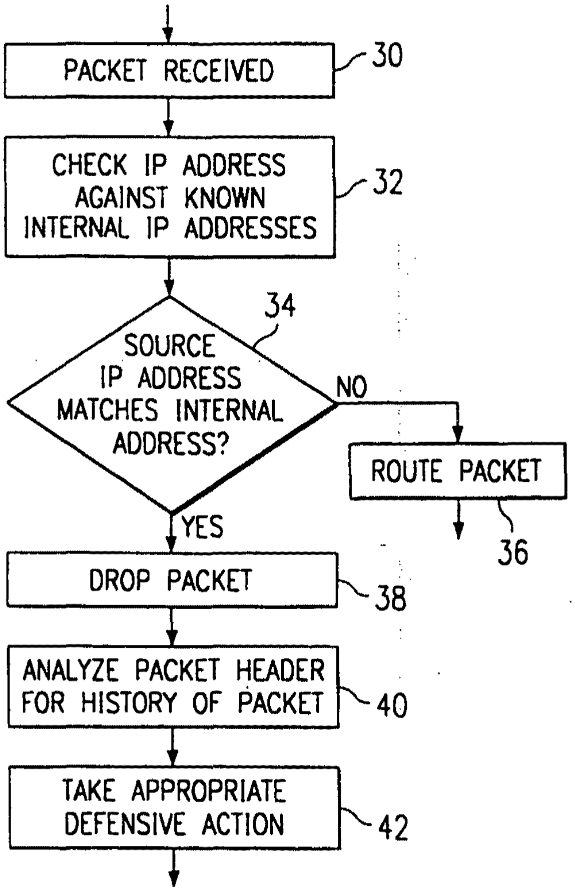
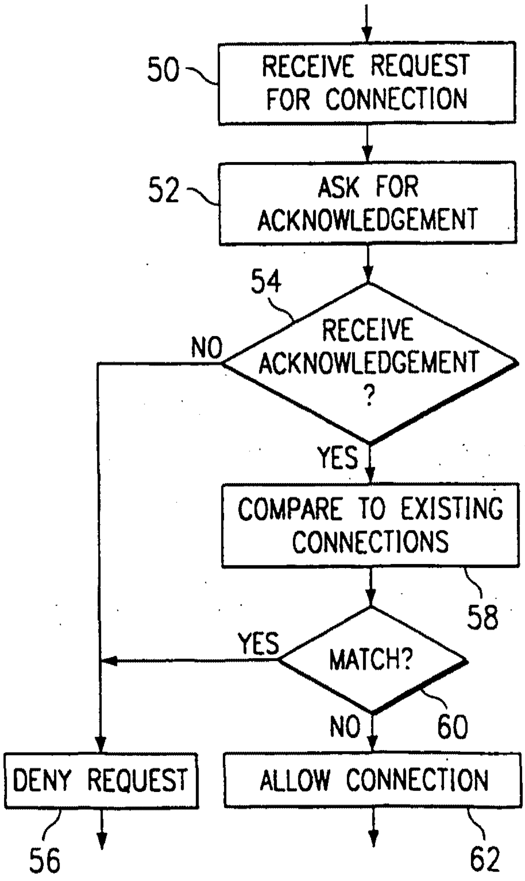

PACKET RECEIVED
30
CHECK IP ADDRESS AGAINST KNOWN INTERNAL IP ADDRESSES
32
34
SOURCE IP ADDRESS MATCHES INTERNAL ADDRESS?
NO
ROUTE PACKET
36
YES
DROP PACKET
38
ANALYZE PACKET HEADER FOR HISTORY OF PACKET
40
TAKE APPROPRIATE DEFENSIVE ACTION
42

FIG. 4


50
RECEIVE REQUEST FOR CONNECTION
52
ASK FOR ACKNOWLEDGEMENT
54
RECEIVE ACKNOWLEDGEMENT ?
NO
YES
COMPARE TO EXISTING CONNECTIONS
58
YES
MATCH?
60
NO
DENY REQUEST
ALLOW CONNECTION
56
62

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 1/00, 13/00, H04L 9/00, 29/06, 12/56, G06F 9/46

A2

(11) International Publication Number: **WO 00/02114**

(43) International Publication Date: 13 January 2000 (13.01.00)

(21) International Application Number: PCT/SE99/01202

(22) International Filing Date: 2 July 1999 (02.07.99)

(30) Priority Data:
9802415-1 2 July 1998 (02.07.98) SE

(71) Applicant *(for all designated States except US)*: EFFNET GROUP AB [SE/SE]; Gustavslundsvägen 151 B, S-167 51 Bromma (SE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: SUNDSTRÖM, Mikael [SE/SE]; Södra Kungsgatan 24, S-972 35 Luleå (SE). JOHANSSON, Olof [US/US]; 10 B Queen Circle, Newark, DE 19702 (US). LINDHOLM, Joel [SE/SE]; Kårhusvägen 5:407, S-977 54 Luleå (SE). BRODNIK, Andrej [SI/SI]; Gotska 6, 1000 Ljubljana (SI). CARLSSON, Svante [SE/SE]; Gnejsstigen 4, S-977 53 Luleå (SE).

(74) Agents: PETRI, Stellan et al.; Ström & Gulliksson AB, P.O. Box 4188, S-203 13 Malmö (SE).

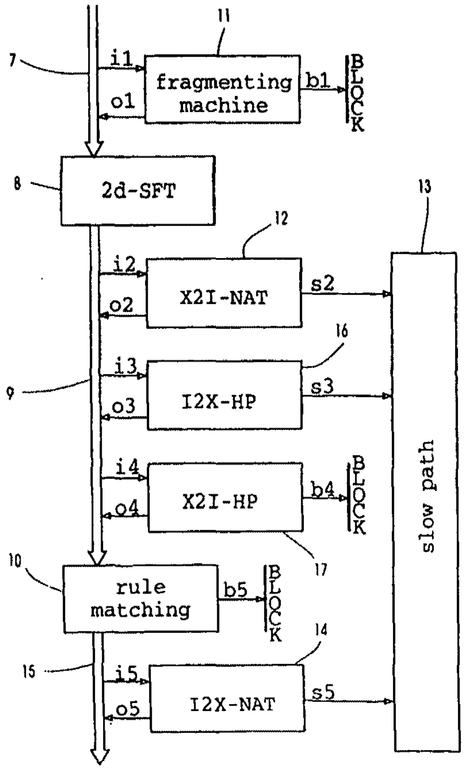
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: FIREWALL APPARATUS AND METHOD OF CONTROLLING NETWORK DATA PACKET TRAFFIC BETWEEN INTERNAL AND EXTERNAL NETWORKS

(57) Abstract

A firewall (3) for controlling network data packet traffic between internal and external networks (1, 5, 4), comprising filtering means selecting from a total set of rules, in dependence of the contents in data fields of a data packet being transmitted between said networks, a rule applicable to the data packet, in order to block said packet or forward said packet through the firewall (3). A 2-dimensional address lookup means (8) performs a 2-dimensional lookup of the source and destination addresses of the packet in a set of address prefixes, each prefix having a subset of rules of the total set of rules, in order to find a prefix, via its representation, associated with said source and destination addresses, and rule matching means (10) for rule matching, on the basis of the contents of said data fields, in order to find the rule applicable to the data packet.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

TITLE: FIREWALL APPARATUS AND METHOD OF CONTROLLING NETWORK DATA PACKET TRAFFIC BETWEEN INTERNAL AND EXTERNAL NETWORKS

Field of the Invention

The present invention relates generally to a firewall apparatus and a method of controlling network data packet traffic between internal and external networks, and more particularly to a firewall apparatus comprising filtering means for selecting from a total set of rules, depending on the contents in data fields of a data packet to be transmitted between said networks, a rule applicable to the data packet, in order to block said packet or forwarded the packet through the firewall, and a method thereof.

Description of the Prior Art

An important issue for most Internet connected organisations is security and consequently firewalls are becoming an important part in most computer and network security strategies in most organisations. Users accessing the webserver or other public services of the organisation must not be able to gain access to internal services such as accounting systems, Internet information servers and other possibly sensitive company information. The service of the systems must not be interrupted - servers and work-stations need to be protected against denial-of-service (DOS) tags from users on the Internet.

A firewall, or filtering router, is a device that works basically the same way as a router. That is, it receives packets on an in-interface, inspects the packets destination address, and forwards the packet on the correct (with respect to the destination address) out-interface. However, a firewall performs a much more thorough inspection of each packet. The source and destination address, source and destination ports, protocol field, flags, and options are also inspected and compared to a list of firewall rules. Depending on which rule matches the packet, the firewall might decide not to forward the packet, for instance if a blocking rule is matched.

In addition to unauthorized access there are other threats that arise when an organisation is connected to the Internet. The bottom line is that data received from unknown sources cannot be trusted. Scanning for viruses and trojan horses in email and webpages are duties performed by some prior art firewalls.

Further, as network bandwidth is increasing, the performance of the firewalls are becoming an important issue.

Firewalls can work on many different levels and provide different kind of functionality for scanning data passing it. However, the basic functionality of all firewalls is to implement filtering based on the contents of the network (IP=Internet Protocol) and transport (UDP, TCP=Transmission Control Protocol and ICMP=Internet Control Message Protocol) layer headers. Without such IP filtering all other functionality, such as data scanning, is useless, that is users on the internal network might just as well configure their network applications not to go through the scanner to connect to remote servers and thus bypass all security functionality.

Companies or organisations are connected to the Internet for different reasons, for example in order to publish information about a company, its products and services on the web, get access to information available on the Internet, and correspond via email.

The company often has internal information that users on the Internet must not be able to access, such as Internet information servers, file servers etc. The most common configuration is to allow connections from the

Internet to a set of servers (web, email, and other public services), but to deny access to other hosts (for example intranet servers). To achieve this a ″demilitarised zone″ (DMZ) is established. Connections to computers in the DMZ can be made from the Internet as well as from the intranet, but access to the intranet from the Internet is restricted. In prior art networks an internal network, such as an intranet is connected to the demilitarised zone via a firewall and the DMZ is connected to the Internet via a router. Consequently, network traffic can pass freely between the Internet and the DMZ, which is completely unprotected from users on the intranet. A reason for this is that prior art firewalls also lack the possibility to connect more than two networks - an internal and an external network.

Other firewalls have three network interfaces. Here, restrictions can be made concerning traffic between the Internet and the DMZ as well as the intranet. Some restrictions are made for traffic to and from hosts in the DMZ, for example the web server only needs to be accessible on the HTTP (Hypertext Transfer Protocol) port. Internet users should not be able to connect to any other services. However, users on the intranet might want to be able to access the web server in more ways than the Internet users for administrative purposes, thus more access should be granted in between these two networks. Similar rules are needed for the email server; SMTP (Simple Mail Transfer Protocol) connections should be allowed from the Internet, but reading email should only be possible for certain allowed hosts on the intranet, and possibly also from some host on the Internet.

In a firewall environment the number of machines in the DMZ is for example 30. The rules for the machines in the DMZ can be different for each machine, but the number of rules per machine is fairly low, for example 10-15. More rules might apply for traffic from the intranet to the DMZ, but these are likely to be more general. Thus, a fairly low number of rules are valid for all machines in the DMZ.

Further, rules regarding traffic between the Internet and the intranet(s) are in most cases few, if any at all. Most traffic should be blocked. However, traffic initiated from the intranet might be allowed.

As the number of users on the Internet grows, the public servers will be visited more frequently, causing more traffic. The traffic to and from the intranet increases as the intranet users are taking part of the increasing amounts of information available on the Internet. Consequently, bandwidth requirements is increasing. This puts greater demands on the performance of the firewalls used.

Thus, the main task for a firewall is packet filtering, that is given an IP packet and a set of rules, which rule should be applied on this packet? If several rules match the same packet a policy needs to be defined to specify which rule to pick. There are two prior art solutions known to this problem. One solution is to pick the rule matching the most number of fields of a packet, and if two rules match the same number of fields, but different ones, an order needs to be specified between them. This is used in the packet classification algorithm by Borg and Flodin, Borg, N. Flodin, Malin, packet classification, June 1997; Borg, N., A Packet Classifier for IP Networks, Masters Lic., Luleå University of Technology, February 1998. Another solution is to define an order between the rules and use that order to define which rule to pick. An advantage of the second solution is that it gives better flexibility when defining filter rules, and the NetBSD firewall code utilise this method.

A filter rule comprises a set of criteria that has to be fulfilled, and an action to perform when they are fulfilled. The criteria are based on IP source and destination addresses (32-bit prefixes), IP protocol field (8 bit-integer), whether or not the packet has IP options set, and what these options are (integer) due IP/TCP source and destination port numbers (2 16-bit integer ranges), TCP header flags (3 bits), ICMP header type and code fields (2 8-bit integers), what interface the packet was read from (8 +8 bits), and what interface the packet is to be forwarded to (8 + 8 bits).

Most firewalls today do not address the rule matching problem in particular. It is common to have a linked list (or an array) of rules, comparing the packet with each and every one of these until a match is found. However, this is not efficient. Another approach is hashing of the rules. Further, if the method for resolving ambiguities among the rules, that is two rules match the same packet, most implementations solve the problem by defining the first or last matching rule as the one to follow.

A prior art firewall, PIX firewall by Cisco Systems, is a connection oriented security device that protects an internal network from an external network. The PIX firewall is a very expensive device and it has an upper limit of about 16000 simultaneous connections. The main part of the PIX firewall is a protection scheme based on the adaptive security algorithm (ASA), which offers stateful connection oriented security. ASA tracks the source and destination address, TCP sequence numbers, port numbers, and additional TCP flags of each packet. This information is stored in a table, and all inbound and outbound packets are compared against entries in the table. Hence, information of each connection established has to be stored during the lifetime of the connection, and thus, the number of connections possible are defined by the memory capacity available. A fully loaded Cisco PIX firewall can operate at about 90 Mbite/s. However, the Cisco PIX firewall also supports port address translation (PAT), whereby more than 64000 internal hosts can be served by a single external IP address.

A prior art packet filter called ipf (IP filter) is included with the standard distribution of net BSD 1.3.

The rule sets in ipf are split up on the interfaces on which they are valid. Furthermore, the rules are checked twice, first when the packet enters the host and second when it leaves the host. Rules only valid for inbound packets are not added to the list of rules checked at the output port, and vice versa. The data structure is basically an optimised linked list.

The Exokernel, Engler, D., Kaashoek, M. F., O'Tool Jr, J., Exokernel: An operating system architecture..., Proceedings of the 15th ACM symposium on Operating Systems principles, December 1995, uses a different approach to handle packet demultiplexing called DPF, Angler, D., Kaashoek, M. F., DPF: Fast, flexible message demultiplexing..., Engler, D., Kaashoek, M. F., Computer Communication Review, Vo. 26, No. 4, October 1996. The rules are written in a special programming language, and thereafter, the are compiled. The compiler knows about all the rules specified, the generated code can be optimised for the expected traffic patents.

Summary of the Invention

It is an objective of the present invention to provide an improved firewall apparatus and a method of controlling network traffic between internal and external networks providing an efficient address lookup and rule matching process in order to achieve an effective and fast IP packet filtering, and an unlimited number of possible connections through the firewall.

This is accomplished by the firewall apparatus and method according to the invention, wherein the set of rules needed to be searched linearly is reduced by segmenting the rule set. The firewall according to the invention comprises 2-dimensional address lockup means performing a two step lookup, first of source and destination addresses of the packet in a set of address prefixes. Each prefix is associated with a subset of rules of a total set of rules. A liner search is performed on the resulting subset of rules in order to find the rule applicable to the present data packet.

Another object of the invention is to provide a fragment machine enabling filtering of all fragments in a fragmented packet.

Still another object of the invention is to provide network address translation means translating internal source addresses to external source addresses of a packet transmitted from the firewall or external source addresses to internal source addresses of a packet transmitted into the firewall.

Another further object of the invention is to provide network address translation means translating internal source addresses to external source addresses of a packet transmitted from an internal network to an external network, or external source addresses to internal source addresses of a packet transmitted from the external network to the internal network.

Still another object of the invention is to provide hole punching means performing a temporary exception from an external-to-internal blocking rule for a connection initiated from the internal network, wherein a returned channel for packets transmitted from the external network to the internal network are established through the firewall.

A further object of the invention is to provide a firewall capable of handling at least 1000 unique rules.

Advantageous of the firewall and the method thereof according to the present invention are the unlimited number of possible simultaneous connections, the fast IP filtering, and the great number of possible rules supported.

Another object of the firewall according to the invention is to provide a firewall comprising a router.

Brief Description of the Drawings

In order to explain the invention in more detail and the advantages and features of the invention preferred embodiments will be described in detail below, reference being made to the accompanying drawings, in which

FIG 1 is shows common network topology comprising the firewall according to the invention,

FIG 2 is a block diagram of the firewall according to the invention,

FIG 3 is an illustrative view of a partition of a two dimensional dense chunk,

FIG 4 is an illustrative view of the data structure according to the invention,

FIG 5 is an illustrative view of a class (0,0) tile,

FIG 6 is an illustrative view of a class (1,1) tile,

FIG 7 is an illustrative view of a class (1,2) tile,

FIG 8 is an illustrative view of a class (2,1) tile,

FIG 9 is an illustrative view of a class (1,3+) tile,

FIG 10 is an illustrative view of a class (3+,1) tile,

FIG 11 is an illustrative view of a class (2+,2+) tile,

FIG 12 shows an example of an unsuccessful search for a particular query key in a Patricia Tree containing six keys, and

FIG 13 shows the Patricia Tree resulting from an insertion of the query key from the unsuccessful search according to FIG 12.

Detailed Description of the Invention

An example of a modern network topology from a company's or an organisation's point of view is shown in FIG 1. An internal network 1, such as an Intranet comprises several network nodes 2 such as PCs, workstations, file servers etc, which are connected to a firewall 3. Companies or organisations connected to an external network 4 (Internet) intend to publish company related information, such as products and services, on the web, get access to information published by other companies or organisations on the Internet, and correspond via email. However, the company might have internal information that users on the Internet not are allowed to access, for example information available via the Intranet information servers, file servers etc. Thus, to allow Internet users to access public information they are allowed to be connected to a limited set of servers, for example the web, email etc., and denied to access information on other hosts, such as Intranet servers. The public servers are available in a "Demilitarised Zone" (DMZ) 5, which is connected to the firewall 3. Further, the firewall 3 is connected to the Internet via a router 6, and, hence, connections to nodes in the DMZ 5 can be made from the external network or Internet 4 as well as from the Intranet 1, but accesses to the Intranet 1 from the Internet 4 is restricted.

In the following description, numerous specific details, are provided in detail in order to give a more thorough description of the present invention. It will be obvious for those skilled in the art that the present invention may be practiced without these specific details. Some well-known features are not described in detail so as not to make the present invention unclear.

One embodiment of the firewall and the different modules in the fast path and how the filtered packets flows through according to the invention is shown in FIG 2.

In a simple case a packet is received from a network 1, 4, or 5 in a firewall input connection 7 and is applied to the input of 2-dimensional address lookup means or a 2d-SFT block 8. A intermediate connection 9 connects the 2d-SFT and rule matching means or block 10, wherein the packet is either passed (down) or blocked b5. However, in order to work properly the firewall according to the invention has a number of additional modules.

In this embodiment a lookup of source address and destination address are performed in the 2d-SFT block 8, resulting in a rule or actually a short list of rules. The rule list remains in the rule matching block 10 until the list is searched and a matching rule is found. Additionally, information of whether the packet might need to be processed by the other modules or not are generated by the 2d-SFT lookup. Some of these decisions are taken during the rule matching which means that the rule matching actually starts before entering the block, as illustrated in FIG 2. The 2d-SFT block 8 is described in detail below.

When a packet is too large to be sent over a link, it is fragmented. This means that everything that follows the IP header is cut into pieces (fragments) and each fragment is supplied with its own IP header. The additional fragments flag and the fragment offset is set in each fragment to indicate if it is the last fragment or not, and to record where the data of the fragment fits into the original (unfragmented) packet.

When a packet is fragmented, only the first fragment, the fragment header, contains the transport header (TCP, UDP, or ICMP header). This means that the following fragments can not be matched against a rule involving for example ports.

According to the invention, a fragment machine 11 collects fragments from each fragmented packet until the fragment header arrives (fragment does not necessarily arrive in order). Then, the pieces of information present only in the fragment header are stored in the entry associated with that fragmented packet, and the collected fragments are applied to the output o1, connected to the connection 7, with the fragment header first. Each fragment that is transmitted from the fragment machine is supplied with the fragment header information, so that it can be processed by the filter just as if it was an unfragmented packet. The additional fragments flag and the fragment offset are checked to determine if the packet is applied to the input i1 - connected to the connection 7 - of the fragment machine 11 or not.

When all fragments of a fragmented packet has been received in the fragment machine 11, the entry for the packet is removed.

At some points, the fragment machine might also decide to block fragments. This happens when broken fragmented packets arrives (possibly as a result of an attack), if the number of collected fragments exceeds a certain limit, or simply as a result of garbage collection (old entries are removed to make place for new ones).

Network Address Translation (NAT) is commonly used when a company have an network with many internal IP addresses and only a few external (real) IP addresses. Some parts of IP address space are reserved for internal addresses, such as 10.*.*.*, 192.168.*.*, and 172.16.*.*. These addresses can freely be used on internal/private networks. However, they must never be visible on the external. Therefore, the firewall is setup to translate internal source addresses to external source addresses as packet goes from the internal to an external network. For packets going in the other direction, the external destination address is translated to an internal address as the packets goes through the firewall. In order to map many internal addresses onto a few external addresses, ports are also used.

For example, the firewall is setup to map internal addresses from 10.1.0.0 to 10.1.255.255 (2^{16} addresses) to external addresses 194.22.187.0 to 194.22.187.255 (2^8 addresses) using ports 20000 to 20255 (2^8 ports).

When a connection is initiated from 10.1.1.1 port 4000 to 130.240.64.46 port 6000, an address a and a port p, so that (a,p) does not collide with any other NAT connection, is picked from the address and port range. Then, each outgoing, internal to external (I2X), packet from that connection, the source address 10.1.1.1 and port 4000 are replaced by a and p respectively. For each incoming, external to internal (X2I) packet, the destination address a and port p are replaced by 10.1.1.1 and 4000, respectively.

In this way, the 256 external addresses together with the 256 ports can represent the 65536 addresses of the internal network.

As a result from the 2d-SFT lookup, also information about if a packet is subject to an external to internal address translation is achieved, and the packet is applied on the input i2 of an X2I-NAT block 12 performing the external to internal address translation. Therefore, the overhead for performing X2I-NAT lookup is removed on all packets not requiring translation. For packets where X2I-NAT lookup is performed, the packets are sent to slow path means 13 via its slow path output s2 in the case of failure since updates of the NAT data structure are dealt with therein. When a successful X2I-NAT lookup is performed, the address and ports are changed and a rule matching of the new source-destination pair is retrieved before the packet is sent to the next filtering step via its output o2.

Also, as a result from the 2d-SFT lookup or from the X2I-NAT lookup, it is clear if the packet is subject to internal to external (I2X) address translation. This is performed basically in the same way as X2I-NAT, but is performed as the last filtering step. A packet subject to internal to external (I2X) address translation received from the output connection 15 of the rule matching block 10 is applied on the input i5 of an I2X-NAT block 14, performing the internal to external address translation. For packets where I2X-NAT lookup is performed, the packets are sent to the slow path means 13 via its slow path output s5 in the case of failure since updates of the NAT data structure are dealt with therein. When a successful I2X-NAT lookup is performed, the address and ports are changed and the packet is transmitted to the appropriate network via its output o2 and the output connection 15.

The reason for having X2I-NAT as the first step after 2d-SFT lookup and I2X-NAT as the last step is that filtering rules are given with respect to internal addresses, which are fixed, and not NAT address, which are assigned dynamically.

Usually, most of the traffic that goes from an external network 4 to an internal network 1 is blocked, to protect the internal network. However, hosts on the internal network are usually allowed to access hosts on the external network 4. In order to receive any return traffic from the external, a temporary exception from the external-to-internal blocking rule must be made for connections initiated from the internal network. This is referred to as hole punching (HP), i.e a hole for returning packets are punched through the firewall. The hole exists only during the lifetime of the connection, and does only affect packets from the connection.

Hole punching also keep track of the TCP sequence numbers in order to protect hole punched connections from being hijacked. Therefore, it is necessary both to perform HP lookup on outbound (I2X) packets performed by an I2X-HP block 16 and inbound (X2I) packets performed by an X2I-HP block 17.

As a result from the 2d-SFT lookup or from X2I-NAT lookup, we know if the packet is subject to internal to external (I2X) or external to internal (X2I) hole punching. This means that we can avoid the overhead from performing HP lookups on packets that can not be subject to hole punching. An outbound packet subject to hole punching is applied to an input i3 of the I2X-HP block 16, whereby the source and destination addresses and ports, and the protocol, are looked up in order to find an existing state. If no such state exists, the packet is sent to the slow path means 13 via its slow path output s3, wherein the HP data structure is updated and a state is created. If a matching state is found, TCP-sequence numbers etc are update before the packet is sent to the next filtering step via another output o3.

The X2I-HP is performed in the same way. An inbound packet subject to hole punching is applied to an input i4 of the X2I-HP block 17, whereby the source and destination addresses and ports, and the protocol, are looked up in order to find an existing state. If no such state exists, an attempt to send the packet through a non-existent hole in a blocking rule has been made and the packet is blocked at its output b4. If a matching state is found, it is updated before the packet is sent to the next filtering step via another output o4.

Again referring to the 2d-SFT block 8, in dependence of the contents in data fields of a data packet being transmitted between said networks, a rule applicable to the data packet is selecting from a total set of rules, whereby said packet is blocked or forwarded through the firewall. In order to reduce the set of rules to be searched linearly, the rule set is segmented. According to the invention, this is performed by means of a 2-dimensional lookup of the source and destination addresses of the packet in a set of address prefixes, wherein each prefix has a subset of rules of the total set of rules, in order to find a prefix associated with the source and destination addresses. Then, based on the contents of said data fields, a rule matching is performed by the rule matching means 10 in order to find the rule applicable to the data packet.

When performing the 2-dimensional lookup of the addresses, each rule is seen as covering a rectangular area of a 2-dimensional plane, wherein the offset and size of the rectangle is determined by the address prefixes and prefix lengths. Hence, the lookup is considered to be the same problem as finding the rectangle surrounding a point in the plane. To simplify the lookup, a restriction is made to assure that each point in the plane is covered by one and only one rectangle, resulting in an easier lookup procedure.

After the 2-dimensional address lookup is performed the lookup continues with a resulting subset of rules associated with the current prefix found. The address fields are, however, not used in the final rule matching. Thus, if a rule is not valid for the addresses of the current packet it is not in the list of rules resulting from the address lookup.

Since each rule is represented by a rectangle covering a part of the total address space and several rules may be applicable to the same addresses, the rectangles may overlap. However, in order to make the method according to the invention to operate in the proper way overlapping rectangles are not allowed. Consequently, in order to fulfil the non-overlap criteria the following steps have to be performed:

1. For each rule, create the rectangle in the address space.

2. Create a set containing only the newly created rectangle. This set will be called the compare set.

3. For all rectangles already in the plane; compare it to each rectangle in the compare set.

4. If they are overlapping, cut out the non-overlapping parts. The rule list of the overlapping parts is assigned the rule from the new rectangle appended at the end thereof.

5. For all parts - if the part was a part of the rectangle already on the plane, return it to the plane. If not, add it to the set of rectangles to be compared.

6. If the compare set is none-empty, return to step 3. Rectangles already in the plane and which have already been compared can be left out.

7. At this state the compare set is empty. If any rectangles were overlapping the new one they are split up into smaller parts if needed, with the common parts having rule lists containing the new rule.

In another method to fulfil the non-overlap criteria there is not just a set of rectangles in the plane. Instead, each rectangle contains, apart from its co-ordinate and rule list index, a set of rectangles or subrectangles. Each of the subrectangles have an additional set of subrectangles. However, sometimes it is necessary to refer to the same subrectangle and to traverse a directed Acyclic graph (DAG) of rectangles depth.

There is always one root rectangle covering the whole plane. This represents the default to follow if all other comparison fail. The rule action is either blocked or allowed to pass depending on the configuration.

A rectangle called root is the root rectangle to which a rectangle new is to be added.

If the root and the new rectangles are of the same size the rules in the new rectangle is added to the rule list associated with the root rectangle.

Iterate over all subrectangles of the root rectangle. If the new rectangle can be completely covered by any of these, make a recursive call with the subrectangle as the root instead and then return.

Once again, iterate over all subrectangles in the root rectangle.

If a subrectangle can be completely contained in the new rectangle, it is moved from the root rectangle to the new rectangle. The rule list of the subrectangle and all rectangles under it needs to be modified to include the rule of the new rectangle as well.

If the subrectangle intersects with the new rectangle, a new rectangle is created comprising the common part of the two. The rule list of the intersecting rectangle is a combination of the original ones. Then, the new rectangle is added to both the original subrectangle and the new rectangle.

Once all rectangles are added to the DAG the graph can be traversed and the list of prefix-defined rectangles that is needed by the two dimensional lookup building code can be produced. The intersecting rectangle will be a proper prefix defined rectangle, but the rest of the surrounding rectangle after the subrectangles have been cut out may not be properly defined by prefixes.

When the data structure is used for filtering lookups as described above, the lookup is made in two steps. First a two dimensional address lookup is performed, resulting in an integer number. This integer is an index into an array of rules, wherein each rule specifies which fields to compare and what action to perform if a match was found. Each rule has a next field indicating which rule to continue with in case of a mismatch. The traversing of the rule list is continued until a match is found, and when proper actions are taken in order to block or forward the packet.

The 2-dimensional prefix problem is solved as follows.

The address space or universe **U** is a 2 dimensional space consisting of integer pairs (s,d) satisfying: $0 \leq s < 2^{32}$, $0 \leq d < 2^{32}$.

A subset R of **U** satisfying: $(s,d) \in R$ if $s_0 \leq s < s_1$, $d_0 \leq d < d_1$, wherein $(s_0,d_0),(s_1,d_1) \in$ **U** is called a rectangle. Further, the pair of points $[(s_0,d_0),(s_1,d_1)]$ uniquely defines *R*.

A rectangle defined by $[(s_0,d_0),(s_1,d_1)]$, where $s_1-s_0 = s_1-2^{i_s} * k_s = 2^{i_s}$ and $d_1-d_0 = d_1-2^{i_d} * k_d = 2^{i_d}$ for some non negative integers i_s, i_d, k_s, and k_d is called a prefix.

Given a point $(s,d) \in$ **U** and a set of prefixes **P** = $\{P_1,P_2,\ldots,P_n\}$, such that **P** is a partition of U, the 2 dimensional prefix matching problem is the problem of computing i such that $(s,d) \in P_i$.

The source-destination part of the firewall filtering problem is represented as a 2-dimensional prefix matching problem, where the set **P** is obtained by converting the routing table and the filtering rules into a partition of prefixes. Since each packet to be filtered requires a prefix matching, it becomes necessary to find a representation of P such that the prefix matching can be computed efficiently.

A number of prefixes that partitions a small 32 x 32 bits universe is shown in FIG 3. Black squares 18 represents bits set (representatives) and white squares 19 represents not set bits. Note: point (0,0) is located in the upper left corner in FIG 3.

For each prefix $P = [(s_0,d_0),(s_1,d_1)] \in$ **P** the point $p_0=(s_0,d_0)$ is chosen as a representative of P. Further, let **p** = $\{p_1, p_2, . . ., p_n\} = \{(s_1,d_1),(s_2,d_2),\ldots,(s_n,d_n)\}$ denotes the set of representatives of the prefixes in **P**.

Given a point $(s_d,d_d) \in \mathbf{U}$, for each $(s,d) \in \mathbf{U}$, such that $s_d \geq s$ and $d_d \geq d$, (s_d,d_d) is a dominating point of (s,d), or alternatively, (s,d) is dominated by (s_d,d_d).

Given a pair of points $(s_1,d_1),(s_2,d_2) \in \mathbf{U}$, the distance between the points under the norm L_∞ is given by:

$$\lim k \to \infty \sqrt[k]{|s_1 - s_2|^k + |d_1 - d_2|^k} = \max(|s_1 - s_2|, |d_1 - d_2|)$$

Now, given a point $p=(s,d)$, the problem of finding the matching prefix in $\mathbf{P}$ is equivalent to the problem of finding the closest dominating point p in $\mathbf{p}$ under the norm L_∞, i.e. the dominating point of $p_i \in \mathbf{p}$ of p minimizing the L_∞-distance between p_i and p. Hence, it is sufficient to represent only the dominating points instead of the prefixes themselves.

As shown in FIG 4, the set $\mathbf{p}$ is conceptually represented as a $2^{32} \times 2^{32}$ points bit matrix, where bit p is set if $p \in \mathbf{p}$. To reduce the space required for the representation, we actually represent $\mathbf{p}$ as a four level 2^{8+8}-ary tree. Each level is (again) conceptually represented as a $2^8 \times 2^8$ bits bit matrix where bit (s,d) is set if there is a dominating point in the sub-tree below. That is, at level 1 (the top level), bit (s,d) represents the presence or absence of a dominating point in the rectangle $[(2^{24}*s,2^{24}*d),(2^{24}*(s+1), 2^{24}*(d+1))]$ of $\mathbf{U}$.

The actual representation of a level is a 2-dimensional dense chunk or simply a 2d-chunk. How and when a level can be represented by a 1-dimensional dense chunk is discussed later. A 2d-chunk consists of 32 x 32 tiles, where each tile represents 8 x 8 bits. Since the points defining a tile are dominating points of prefixes, not all 2^{64} kinds of tiles are possible. In fact, we impose a restriction on the tiles so that only 677 different kinds are possible.

If there is a point in a tile T (a point in some of the sub-universes represented by one of the bits in the tile) having its closest dominating point in another tile T_d then all points in T have their closest dominating points in T_d. The definition of a dominating point is extended to a dominating tile. The tile T_d is called a dominating tile of T, or alternatively, tile T is dominated by the tile T_d.

In order to fulfil the requirement of the previous definition the following lemma is needed.

If P = $[(s_0,d_0),(s_1,d_1)]$ is a prefix satisfying $s_1-s_0>1$, then $[(s_0,d_0),(s_0+2^i,d_1)]$ and $[(s_0+2^i,d_0),(s_1,d_1)]$, wherein $s_1-s_0=2^i$ for some none-negative integer i, are also prefixes. The lemma for the other dimension is symmetrical.

By the lemma above, a prefix can be cut into 2 parts whenever required. Hence, given a set of prefixes $\mathbf{P}_d$ with representatives in the tile T_d we can repeatedly cut them until all prefixes has their endpoints in the same tile, in both dimensions, to fulfil the requirement above. This is called tile cutting and a crucial part of the construction of dense2d chunks.

The different kinds of tiles are divided into seven classes shown in FIG 5-11. For each class the/a tile is shown as a bit matrix in (asterisks represents bits that can be either 0 or 1). For each bit set (not *) and tile class there are also lines indicating the guaranteed boundaries of the subset dominated by that bit (point). Note that a set bit in a tile can typically dominate points in other tiles to the right and/or below. We also give the number of different kinds of tiles in the class and distinguish between natural and restricted tile classes. Finally, we describe how the tiles are represented/encoded in the dense2d chunk.

A class (0, 0) tile is shown in FIG 5. No bit is set: natural, 1 kind, and always dominated by a tile T_d from class (1, 1), (1, 2), (2, 1), (1, 3+), or (3+, 1). Finding the dominating point of a point in bit (s_b,d_b) in a class (0, 0) tile is exactly the same as finding the dominating point of the corresponding point in bit (s_b,d_b) of its dominating tile T_d. Hence, a class (0, 0) tile can, and should, always be encoded exactly the same way as its dominating tile T_d.

A class (1, 1) tile is shown in FIG 6. One bit is set: natural, 1 kind, and possibly dominates class (0, 0) tiles to the right and/or below. Since all points within this tile has the same closest dominating point, we simply encode a reference to that point within the tile itself

A class (1, 2) tile is shown in FIG 7. Two bits in the first row (D-dimension) are set: natural, 1 kind, and possibly dominates class (0, 0) tiles below. Can not dominate class (0, 0) tiles to the right.

There are two closest dominating points of the points in this tile, one for the points in the left half, and one for the points in the right half. We encode references to both these dominating points as an array of length 2, and can then use the left/right half of the query point as indices.

A class (2, 1) tile is shown in FIG 8. Two bits in the first column (S-dimension) are set: natural, 1 kind, and possibly dominates class (0, 0) tiles to the right. Can not dominate class (0, 0) tiles below. There are two closest dominating points of the points in this tile, one for the points in the top half, and one for the points in the bottom half. References to both these dominating points are encoded as an array of length 2, and can then use the top/bottom half of the query point as indices.

A class (1, 3+) tile is shown in FIG 9. Three or more bits in the first row are set: natural, 24 kinds, and possibly dominates class (0, 0) tiles below. Can not dominate class (0, 0) tiles to the right. There may be many dominating points of the points in this class of tiles. It is necessary to encode the kind of the tile since there are 24 different kinds of tiles. Further, for each bit set in the first row, a pointer to the dominating point below (if there is only one) or to the next level chunk (if there several dominating points) are encoded. Finally, a reference to the first pointer is encoded (a base pointer). In this way, the dominating point (or a reference to the next level chunk) of a query point (s,d) can be found by simply inspecting in which column the d is and together with the kind of the chunk perform a table lookup to retrieve a pointer offset x, and finally retrieve the pointer x pointers away from the base pointer. Note that any next level chunk only needs to be one (D-)dimensional since all representatives in the tile lies on the same S-co-ordinate.

A class (3+, 1) tile is shown in FIG 10. Three or more bits in the first column are set: natural, 24 kinds, and possibly dominates class (0, 0) tiles to the right. Can not dominate class (0, 0) tiles below. There may be many dominating points of the points in this class of tiles. It is necessary to encode the kind of the tile since there are 24 different kinds. Further, for each set bit in the first column, a pointer to the dominating point below (if there is only one) or to the next level chunk (if there several dominating points) are encoded. Finally, a reference to the first pointer is encoded (a base pointer). In this way, the dominating point (or a reference to the next level chunk) of a query point (s,d) can be found by simply inspecting in which row the s is and together with the kind of the chunk perform a table lookup to retrieve a pointer offset x, and finally retrieve the pointer x pointers away from the base pointer. Note that any next level chunk only needs to be one (S-)dimensional since all representatives in the tile lies on the same D-co-ordinate.

A class (2+, 2+) tile is shown in FIG 11. Two or more bits are set in both the first row and the first column: restricted, 625 kinds, can not dominate another tile, and can not be dominated by another tile. There are typically many dominating points in this class of tiles. The encoding is performed exactly as for class (1, 3+) and (3+, 1) tiles. However, a restriction is imposed to reduce the number of different kinds before performing the actual encoding. The first task is to impose a restriction similar to the tile restriction of Definition 8 on each bit. Then a pair of bit vectors of length 8, Sv and Dv, is computed wherein

S_i = 1, if there is a bit set in the *i*th row, and
0, otherwise
D_i = 1, if there is a bit set in the *i*th column, and
0, otherwise

A new tile is finally created, by computing the product of Sv and Dv^T using matrix multiplication, and encoded.

As in class (1, 3+) and (3+, 1) tiles, one dimensional sub-levels may be provided also in this case. It is checked whether all representatives in a bit, containing more than one representative, is in the same row in **U**, which means that the S-dimension collapses, or on the same column in **U**, which means that the D-dimension collapses.

A further description of the data structures used in the firewall for representing NAT and HP entries.

In both cases, the pair of IP addresses *saddr* and *daddr*, the pair of ports *sport* and *dport*, and the protocol *proto* of the processed packet are used as key in the lookup. The first step in the lookup is to compute a hash value. This is accomplished using very simple and fast instructions such as bit shifts bit-wise logical operators. Using the hash value as index, a 16 bits pointer is then retrieved from a large array (the Hash table).

The pointer is either 0, which means that the lookup failed (empty) or refers to the root of a Patricia tree, which is a very efficient data structure for representing small sets of keys. If the pointer refers to a Patricia tree, a key is built by concatenating the bit patterns of *saddr*, *daddr*, *sport*, *dport*, and *proto*. The key is then used when searching the Patricia tree as described in the next section.

A Patricia Tree, is a binary tree that treats query keys as bit arrays, and uses a bit index in each internal node to direct the branching. Searching is accomplished by traversing the tree from the root to a leaf. When visiting an internal node with bit index i, bit i of the query key is inspected to determine whether to continue the search in the left (if the bit is 0) or right (if the bit is 1) sub-tree. The traversal stops when arriving at a leaf. To determine if the query key is present in the table or not, the query key is then compared to the key stored in that leaf. If the two keys are equal, the search is successful.

FIG 12 illustrates an example of an unsuccessful search for the query key 001111 in a Patricia Tree containing six keys. Bits no. 0, 2, and 3 are inspected during the traversal, which ends at the leaf with key 011101. As the query and leaf keys are compared, a mismatch is detected in bit no. 1.

With respect to the bit indices stored in the internal nodes, a Patricia Tree is heap ordered. That is, any internal node, except the root, has a bit index greater than the bit index of its parent. It follows that all keys stored in a sub-tree rooted at a node with bit index i are identical up to, and including, bit i-1.

Insertion is accomplished by first performing an unsuccessful search, and recording the index i of the first mismatching bit in the comparison of the query and leaf key. Two new nodes are then created, a new internal node with index i and a leaf node for the query key. Depending on whether the i th bit of the query key is 0 or 1, the leaf is stored as the left or right sub-tree, respectively, of the internal node. By using the other sub-tree field as link field, the internal node is then inserted directly above the node with smallest bit index larger than i in the path traversed from the root to the leaf.

FIG 13 shows the Patricia Tree resulting from inserting the query key from the unsuccessful search of the previous example in FIG 12. A new internal node with bit index 1 is created, and inserted between the nodes with bit indices 0 and 2, in the path traversed from the root.

The Patricia Hashing used for hole punching works exactly as described above -a simple Hash table lookup followed by a Patricia tree lookup. Most of the time, a leaf is reached directly, which means that it is not necessary to build a bit array from the parameters - these are compared directly to corresponding fields in the structure containing/representing the Patricia leaf.

One lookup function *hp_lookup(iaddr, xaddr, iport, xport, protc)* is provided that are used both for I2X-HP and X2I-HP. The only difference between these are the order in which the parameters are given. For I2X-HP, the function call is *hp_lookup(saddr, daddr, sport, dport, proto)* and for X2I-HP the call is *hp_lookup(daddr, saddr, dport, sport, proto)*.

The lookup function returns a reference to a structure containing the Patricia leaf key, i.e. *iaddr*, *xaddr*, *iport*, *xport*, and *proto*, and a couple of other fields representing the state of the connection, for example TCP sequence numbers.

The Patricia Hashing for NAT is slightly more complicated than for HP. The reason is that three different addresses and ports, *iaddr*, *naddr*, *xaddr*, *iport*, *nport*, *xport*, are involved, as opposed to HP where only two addresses and ports are involved. This means that the difference between I2X and X2I becomes a little more tricky than just swapping addresses and ports in the lookup.

The problem is solved by letting the least significant bit of the hash value reflect if the lookup is I2X or X2I (this is essentially the same as using two hash tables). The structure containing the Patricia leaf keys for a NAT connection is the same for I2X and X2I and it contains all three addresses and ports.

There are two lookup functions, *nat_i2x_lookup(saddr, daddr, sport, dport, proto)* and *nat_x2i_lookup(saddr, daddr, sport, dport, proto)*. Both functions uses the arguments to compute a hash value where the least significant bit is set to accordingly. If the resulting pointer refers to a Patricia node (internal node), the addresses, ports, and protocol are concatenated to create the bit array needed for traversing the Patricia tree. When the leaf structure is reached, the addresses, ports, and protocol are compared to the corresponding fields in the leaf.

When a packet is subject to I2X-NAT:

saddr (of the packet) is compared to *iaddr* (of the leaf structure)
daddr is compared to *xaddr*
sport is compared to *iport*
dport is compared to *xport*
proto is compared to *proto*

If all of these matches, the lookup is successful, and the source address and port, *saddr* and *sport*, of the packet are replaced by *naddr* and *nport* (of the leaf structure), respectively, before the packet is forwarded.

When a packet is subject to X2I-NAT:

saddr (of the packet) is compared to *xaddr* (of the leaf structure)
daddr is compared to *naddr*
sport is compared to *xport*
dport is compared to *nport*
proto is compared to *proto*

If all of these matches, the lookup is successful, and the destination address and port, *daddr* and *dport*, of the packet are replaced by *iaddr* and *iport* (of the leaf structure), respectively, before the packet is sent to the next processing step.

Updates of the HP and NAT data structures are performed by the EffNIX kernel (previously NetBSD) running on the BSP (processor 1) but most of the lookups are performed by the forwarding kernel running on the AP (processor 2). There are only one instance of the HP data structure and one instance of the NAT data structure. These resides in shared memory and are accessed by the two processors simultaneously. This results in a very interesting synchronisation problem - one writer and one reader. The synchronisation is solved by letting the update routines invalidate the leafs structures and nodes before changing anything (writing). The lookup routines checks that the accessed leafs and nodes are valid before and after they have been accessed, and also that they have not been changed during the access. If a race occurs and is detected (all dangerous race conditions are detected) the lookup fails and the packet is sent to the BSP and dealt with there (either a successful lookup followed by processing is performed, or the data structures are updated).

It should be apparent that the present invention provides a firewall apparatus and a method of controlling network data packet traffic between internal and external networks that fully satisfies the aims and advantages set forth above.

Although the invention has been described in conjunction with a specific embodiment thereof, this invention is susceptible of embodiments in different forms, with the understanding that the present disclosure is to be considered as an exemplification of the principles of the invention and is not intended to limit the invention to the specific embodiment illustrated.

CLAIMS

1. A firewall (3) for controlling network data packet traffic between internal and external networks (1,5,4), comprising filtering means for selecting from a total set of rules, in dependence of the contents in data fields of a data packet being transmitted between said networks a rule applicable to said data packet, in order to block said packet or to forwarded said packet through the firewall (3), characterized by 2-dimensional address lookup means (8) for a 2-dimensional lookup of said source and destination addresses of the packet in a set of address prefixes, each prefix having a subset of rules of the total set of rules, in order to find a prefix, via its representation, associated with said source and destination addresses, and rule matching means (10) for rule matching - on the basis of the contents of said data fields in order to find the rule applicable to said data packet.

2. A firewall according to claim 1, characterized in that said 2-dimensional address lookup means (8) comprises means for finding the prefix associated with said source and destination addresses by determining the closest dominating point p in **p** under the norm L_∞, i.e. the dominating point of $p_i \in \mathbf{p}$ of p minimising the L_∞-distance between p_i and p.

3. A firewall according to claim 2, characterized in that

the source and destination addresses are represented by a point (s,d) $\in \mathbf{U}$, wherein **U** is a 2 dimensional address space represented by integer pairs (s,d) satisfying: $0 \leq s < 2^{32}$, $0 \leq d < 2^{32}$,

the prefixes $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ is a partition of the address space **U**, and each prefix P_i is a logical rectangle R in the address space **U** defined by $[(s_0,d_0), (s_1,d_1)]$, where $s_1-s_0 = s_1-2^{i_s} * k_s = 2^{i_s}$ and $d_1-d_0 = d_1-2^{i_d} * k_d = 2^{i_d}$ for some non negative integers i_s, i_d, k_s, and k_d,

said logical rectangle R being a subset of **U** satisfying: $(s,d) \in R$ if $s_0 \leq s < s_1$, $d_0 \leq d < d_1$, wherein $(s_0,d_0), (s_1,d_1) \in$ **U**, and the pair of points $[(s_0,d_0), (s_1,d_1)]$ uniquely defines said rectangle *R*.

4. A firewall according to claim 2 or 3, characterized in that

for each prefix $P = [(s_0,d_0),(s_1,d_1)] \in$ **P**, the point $p_0=(s_0,d_0)$ is a representative of P, and **p** $= \{p_1, p_2, . . ., p_n\} = \{(s_1,d_1),(s_2,d_2),\ldots,(s_n,d_n)\}$ is the set of representatives of the prefixes in **P**, wherein given a point $(s_d,d_d) \in$ **U**, for each $(s,d) \in$ **U**, wherein $s_d \geq s$ and $d_d \geq d$, (s,d) is dominated by (s_d,d_d).

5. A firewall according to claim 3, characterized in that, given a pair of points $(s_1,d_1),(s_2,d_2) \in$ **U**, the distance between the points under the norm L_∞ is given by:

$$\lim k \to \infty \sqrt[k]{|s_1 - s_2|^k + |d_1 - d_2|^k} = \max(|s_1 - s_2|, |d_1 - d_2|).$$

6. A firewall according to any of the preceding claims, characterized by a fragment machine (11) comprising fragment collecting means for collecting packet fragments from a fragmented packet until a fragment header of said packet is received, fragment header storing means for storing in an entry means information present in a fragment header field of the packet, fragment forwarding means for forwarding packet fragments provided with fragment header information starting with the fragment

header, wherein each fragment is processed by the filtering means as a regular unfragmented packet.

7. A firewall according to any of the preceding claims, characterized by network address translation means (12,14) for translating, in dependence of the information in the prefix, internal source addresses to external source addresses of a packet transmitted out through the firewall (3), or external source addresses to internal source addresses of a packet transmitted in through the firewall (3).

8. A firewall according to any of the claims 1-6, characterized by network address translation means (12, 14)for translating, in dependence of the information in the prefix internal source addresses to external source addresses of a packet transmitted from the internal network (1) to the external network (4), or external source addresses to internal source addresses of a packet transmitted from the external network (4) to the internal network (1).

9. A firewall according to any of the preceding claims, characterized by hole punching means (16,17) for determining, on the basis of the information in the prefix, if said packet is subject to a temporary exception from an external-to-internal blocking rule for a connection initiated from the internal network, wherein a return channel for packets transmitted from the external network (4) to the internal network (1) is established through the firewall during the lifetime of the connection.

10. A firewall (3)for controlling network data packet traffic between internal and external networks (1,5,4), comprising filtering means for selecting from a total set

of rules, in dependence of the contents in data fields of a data packet being transmitted between said networks, a rule applicable tc the data packet, in order to block said packet or to forwarded the packet through the firewall (3), characterized by a fragment machine (11) comprising fragment collecting means for collecting packet fragments from a fragmented packet until a fragment header of said packet is received, fragment header storing means for storing in an entry means information present in a fragment header field of the packet, fragment forwarding means for forwarding packet fragments provided with fragment header information starting with the fragment header, wherein each fragment is processed by the filtering means as a regular unfragmented packet.

11. A method of controlling network data packet traffic between internal (1,5) and external networks (4) through a firewall (3), comprising the steps of,

selecting from a total set of rules, in dependence of the contents in the data fields of a data packet being transmitted between said networks, a rule applicable to the data packet,

applying said rule on said packet, and

depending on the rule, blocking said packet or forwarding said packet through the firewall (3),

characterized in that said filtering comprises the further steps of:

performing a 2-dimensional lookup of the source and destination addresses of the packet in order to find a prefix, via its representation, associated with said source and destination addresses in a set of address prefixes, each prefix having a subset of rules of the total set of rules,

and on the basis of the contents of said data fields of the packet, performing a rule matching on the subset of rules in order to find the rule applicable to the data packet.

12. A method according to claim 11, characterized in that preceding the step of selecting a rule applicable to the data packet it comprises the further steps of:

collecting packet fragments from a fragmented packet until a fragment header of said packet is received,

storing in an entry means information present in a fragment header field of the packet, and

forwarding packet fragments provided with fragment header information starting with the fragment header, wherein each fragment is processed by the filtering means as a regular unfragmented packet.

13. A method according to claim 11 or 12, characterized in that preceding the step of performing a rule matching it comprises the further step of:

in dependence of the information in the prefix, translating the external source address to an internal source address of a packet to be transmitted in through the firewall (3).

14. A method according to any of the preceding claims 11-13, characterized in that preceding the step of performing a rule matching it comprises the further step of:

depending on the information in the prefix, translating the external source address to an internal source address of a packet to be transmitted from the external network (4) to the internal network (1,5).

15. A method according to any of the preceding claims 11-14, characterized by the further step of:

depending on the information in the prefix translating the internal source address to an external source address of a packet to be transmitted out through the firewall (3).

16. A method according to any of the preceding claims 11-15, characterized by the further step of:

depending on the information in the prefix translating the internal source address to an external source address of a packet to be transmitted from the internal network (4) to the external network (1).

17. A method according to any of the preceding claims 11-16, characterized in that preceding the step of performing a rule matching it comprises the further steps of:

based on the information in the prefix, determining if said packet is subject to a temporary exception from an external-to-internal blocking rule for a connection initiated from the internal network (1),

if so, establishing a return channel for packets transmitted from the external network (4) to the internal network (1) through the firewall (3), having a duration corresponding to the lifetime of the connection.

18. A method of controlling network data packet traffic between internal and external networks (1,5,4) through a firewall (3), comprising the steps of,

in dependence of the contents in the data fields of a data packet being transmitted between said networks, selecting from a total set of rules a rule applicable to the data packet,

applying said rule on said packet,

and depending on the rule, blocking said packet or forwarding said packet through the firewall (3), characterized in that preceding the step of selecting a rule applicable to the data packet it comprises the further steps of:

collecting packet fragments from a fragmented packet until a fragment header of said packet is received,

storing in an entry means information present in a fragment header field of the packet, and

forwarding packet fragments provided with fragment header information starting with the fragment header, wherein each fragment is processed by the filtering means as a regular unfragmented packet.

19. A method according to any of the preceding claims 11-18, characterized in that the step of performing a 2-dimensional lookup of the source and destination addresses of the packet comprises the further step of:

finding the closest dominating point p in **p** under the norm L_∞, i.e. the dominating point of $p_i \in \mathbf{p}$ of p, which minimises the L_∞-distance between p_i and p.

20. A method according to claim 19, characterized in that

the source and destination addresses are represented by a point $(s,d) \in \mathbf{U}$, wherein **U** is a 2 dimensional address space represented by integer pairs (s,d) satisfying: $0 \leq s < 2^{32}$, $0 \leq d < 2^{32}$,

the set of prefixes $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ is a partition of the address space **U**,

each prefix P_i is a logical rectangle R in the address space **U** defined by $[(s_0,d_0),(s_1,d_1)]$, where $s_1-s_0 = s_1-2^{i_s} * k_s = 2^{i_s}$ and $d_1-d_0 = d_1-2^{i_d} * k_d = 2^{i_d}$ for some non negative integers i_s, i_d, k_s, and k_d, wherein the logical rectangle R is a subset of **U** satisfying: (s,d) ∈ R if $s_0 \leq s < s_1$, $d_0 \leq d < d_1$, wherein (s_0,d_0), $(s_1,d_1) \in$ **U**, and the pair of points $[(s_0,d_0),(s_1,d_1)]$ uniquely defines said rectangle *R*,

for each prefix $P = [(s_0,d_0),(s_1,d_1)] \in$ **P**, the point (s_0,d_0) is a representative of P, and **p** = $\{p_1, p_2, \ldots, p_n\} = \{(s_1,d_1),(s_2,d_2),\ldots,(s_n,d_n)\}$ are the set of representatives of the prefixes in **P**, wherein given a point $(s_d,d_d) \in$ **U**, for each (s,d) ∈ **U**, wherein $s_d \geq s$ and $d_d \geq d$, (s,d) is dominated by (s_d,d_d), and

given a pair of points $(s_1,d_1),(s_2,d_2) \in$ **U**, the distance between the points under the norm L_∞ is given by:

$$\lim k \to \infty \sqrt[k]{|s_1 - s_2|^k + |d_1 - d_2|^k} = \max(|s_1 - s_2|,|d_1 - d_2|).$$

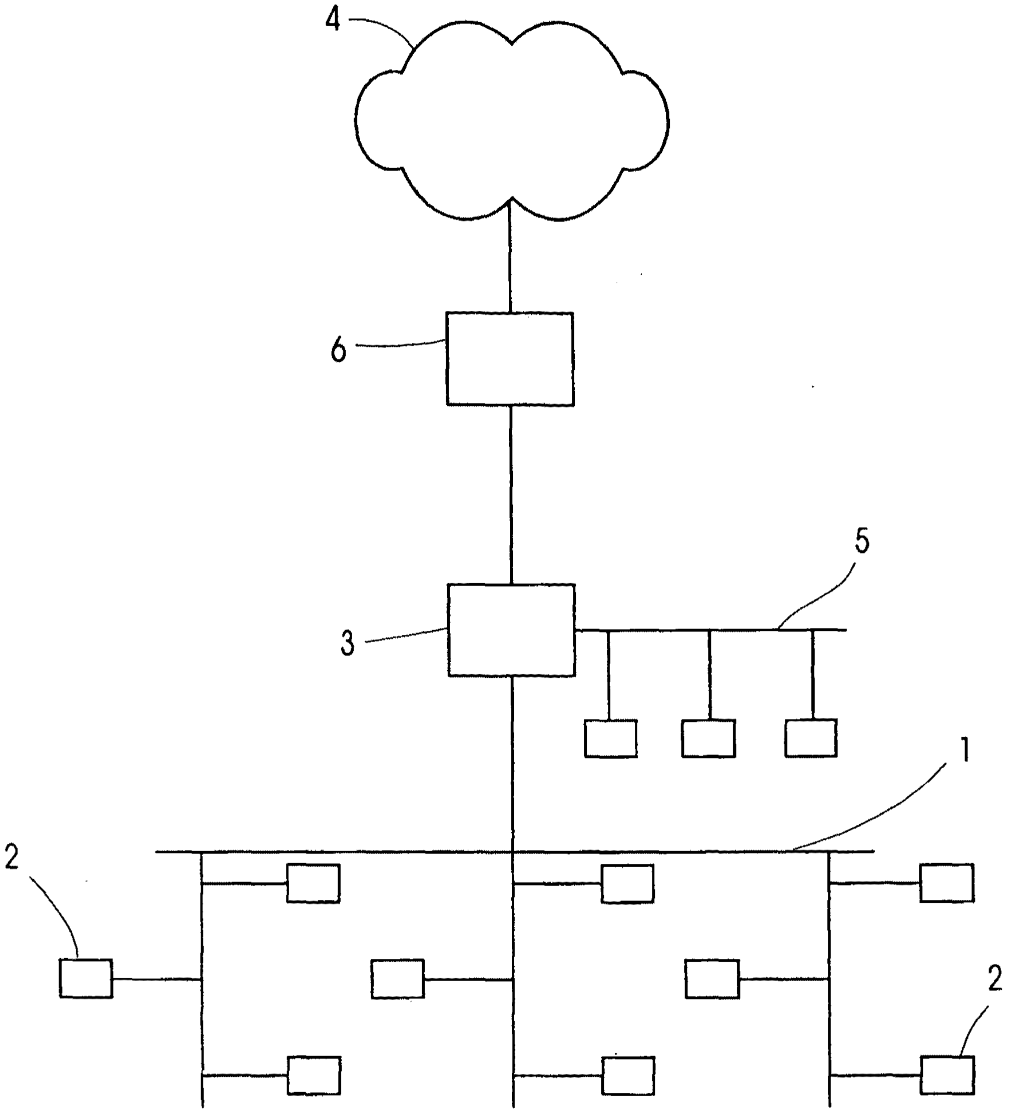


FIG. 1

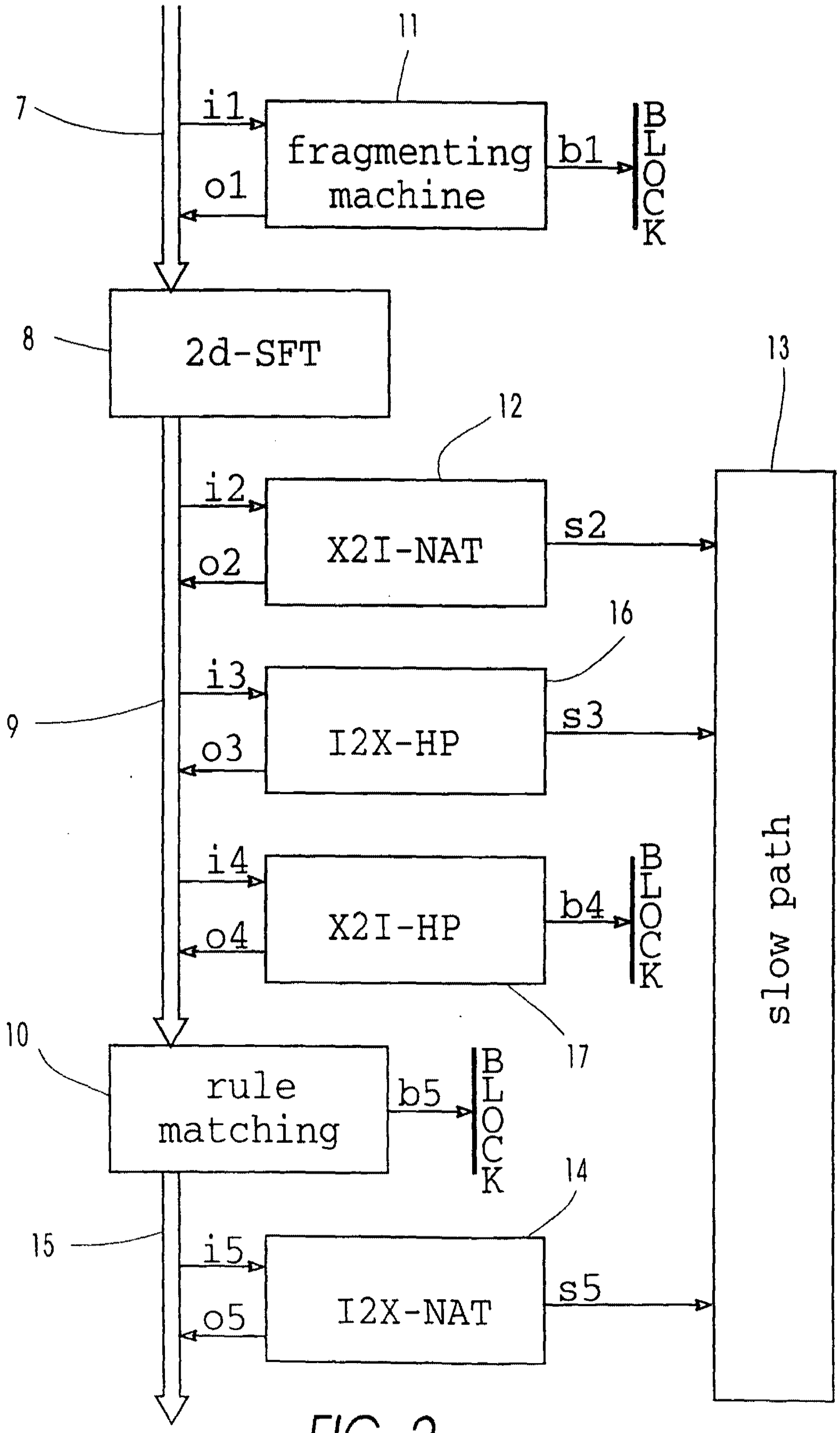

11
7
i1
fragmenting machine
o1
b1
BLOCK
8
2d-SFT
13
12
i2
X2I-NAT
s2
o2
16
i3
9
I2X-HP
s3
o3
i4
X2I-HP
b4
BLOCK
o4
slow path
10
17
rule matching
b5
BLOCK
14
15
i5
I2X-NAT
s5
o5

FIG. 2

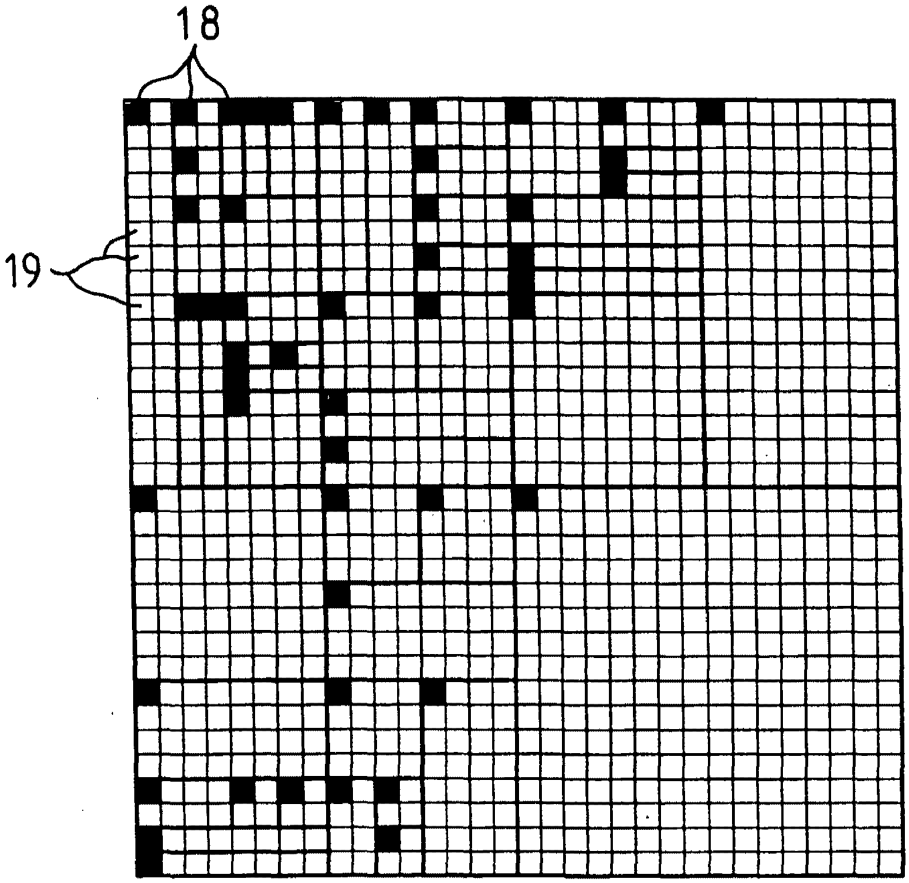


FIG. 3

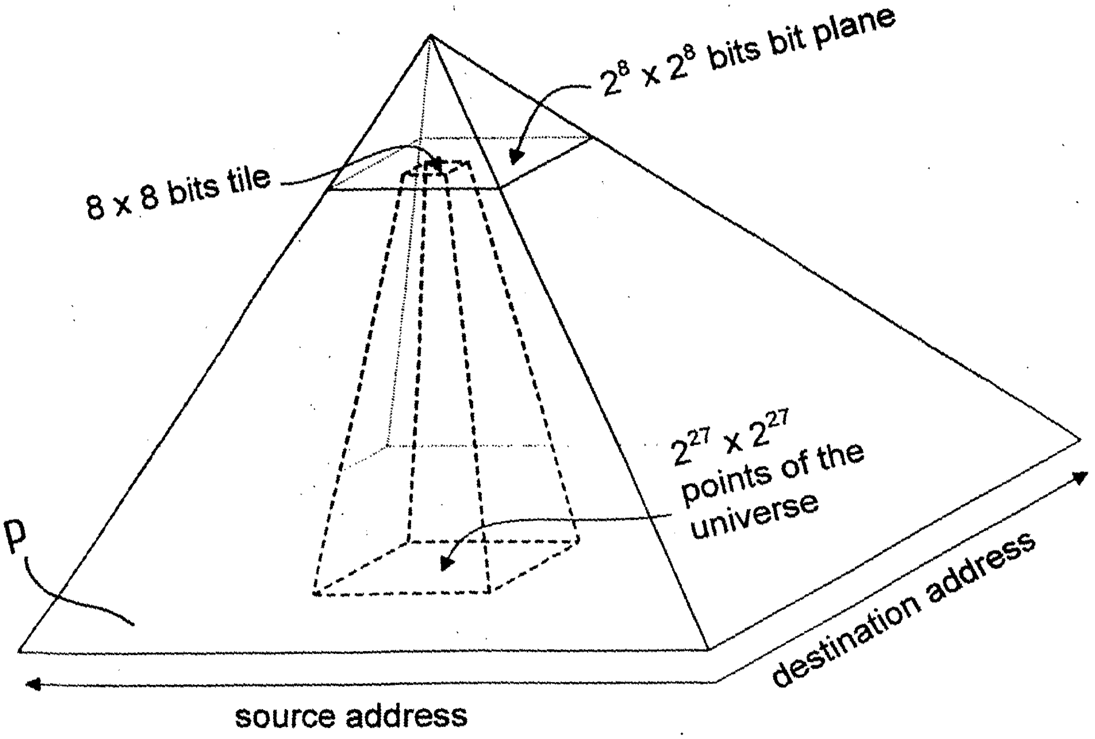

2^8 x 2^8 bits bit plane
8 x 8 bits tile
2^{27} x 2^{27} points of the universe
P
destination address
source address

FIG. 4

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIG. 5

1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

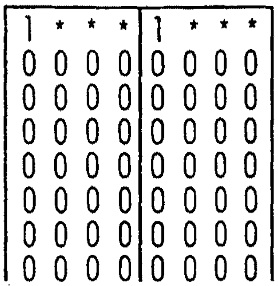
FIG. 6

1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIG. 7

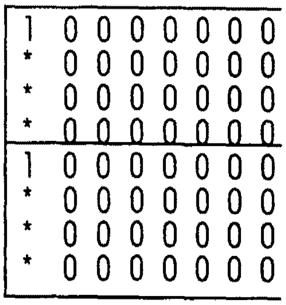
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIG. 8



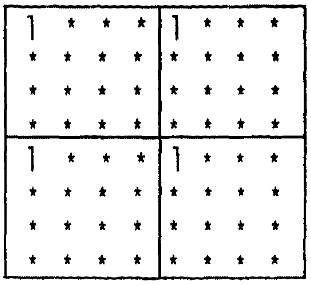
1	*	*	*	1	*	*	*
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIG. 9



1	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0

FIG. 10



1	*	*	*	1	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
1	*	*	*	1	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

FIG. 11

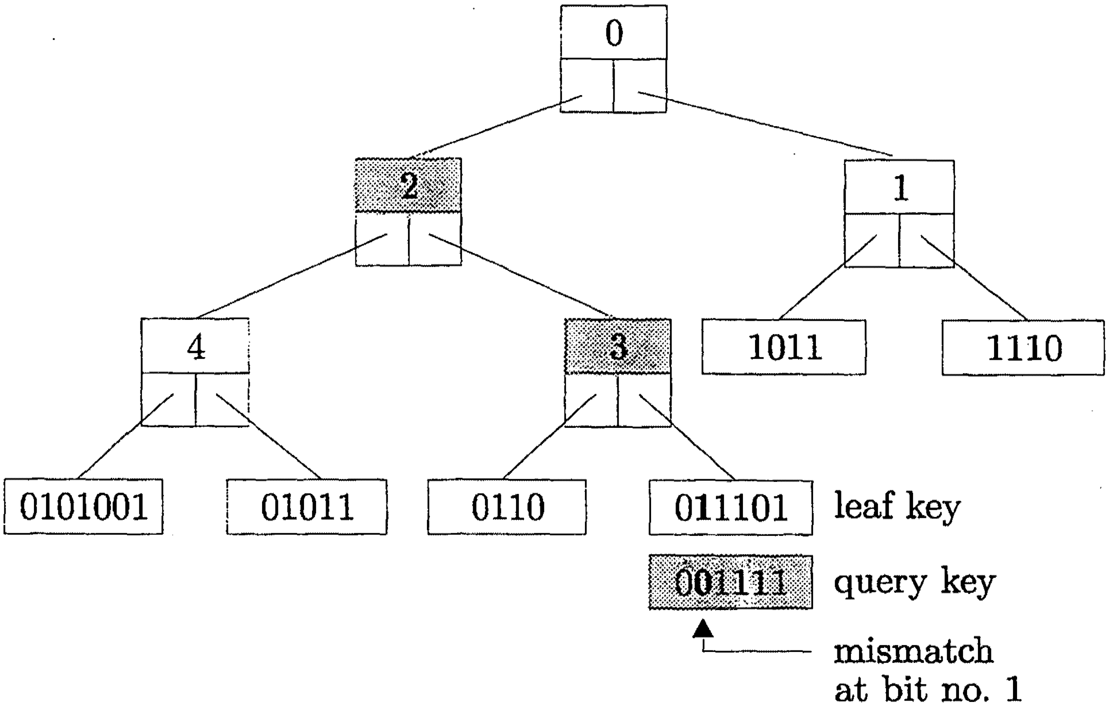

0
2
1
4
3
1011
1110
0101001
01011
0110
011101
leaf key
001111
query key
mismatch
at bit no. 1

FIG. 12

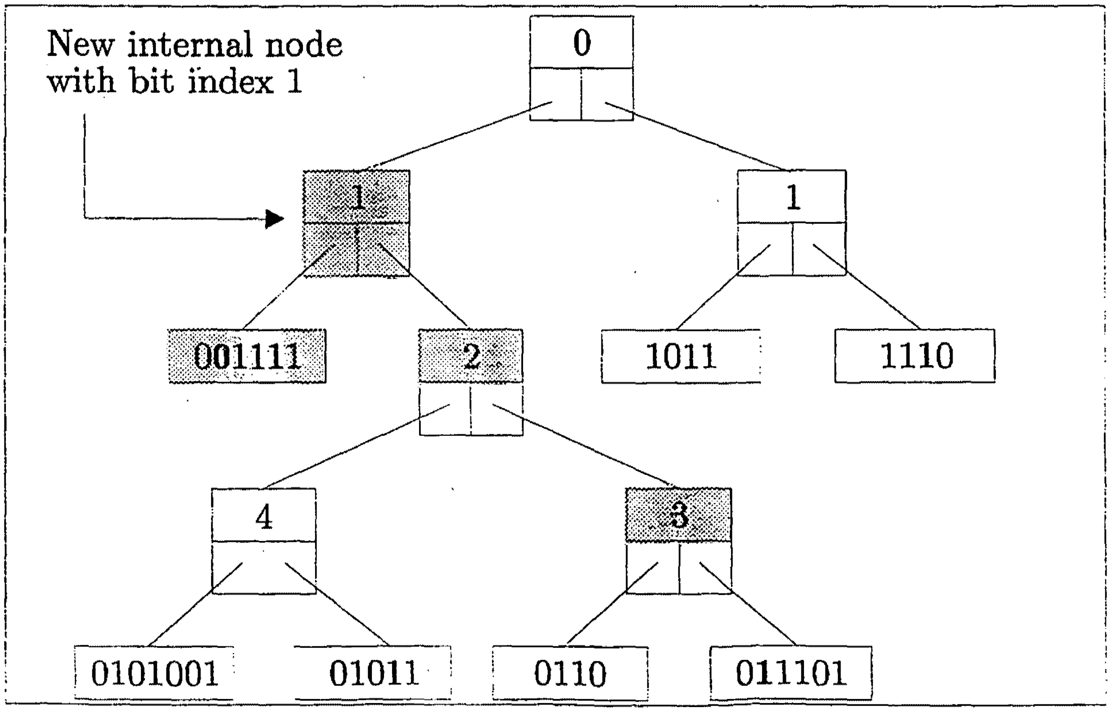

New internal node
with bit index 1
0
1
1
001111
2
1011
1110
4
3
0101001
01011
0110
011101

FIG. 13

07-10-00

d

FORM PTO-1082

Attorney Docket No 802-001

BOX PATENT APPLICATION
THE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D C 20231

Sir

Transmitted herewith for filing is the patent application of
Inventor(s): Andrew K. Krumel
For (Title): REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

Enclosed are.

[x] 13 sheets of informal drawings.

[x] An assignment of the invention to 802 Systems, Inc

[ ] A certified copy of a ______________________ application

[x] A declaration and power of attorney

[x] A verified statement to establish small entity status under 37 CFR 1 9 and 37 CFR 1.27.

[ ] Applicant(s) claim convention priority under 35 U.S.C. 119 based on ______________ application
Serial No. ______________ filed ______________.

[ ] ______________________________

	(Col 1)	(Col. 2)	SMALL ENTITY			OTHER THAN A SMALL ENTITY	
FOR:	NO. FILED	NO EXTRA	RATE	FEE	OR	RATE	FEE
BASIC FEE				$380.00	OR		760 00
TOTAL CLAIMS	66 -20 =	46	x 9=	$414.00	OR	x 18=	$
INDEP. CLAIMS	2 - 3 =	0	x 39=	$	OR	x 78=	$
MULTIPLE DEPENDENT CLAIM(S) PRESENTED =			+ 130=	$	OR	+ 260=	$
			TOTAL	$ 794.00	OR	TOTAL	$

* If the difference in Col 1 is less than zero, enter "0" in Col 2

[x] Please charge Deposit Account No. 50-0251 or backup account 12-2175 in the amount of $794 00
An additional copy of this sheet is enclosed.

[ ] A check in the amount of $__________ to cover the filing fee is enclosed

[x] The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175 An additional copy of this sheet is enclosed

[x] Any additional filing fees required under 37 CFR 1.16.
[x] Any patent application processing fees under 37 CFR 1 17

[x] The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175. An additional copy of this sheet is enclosed

[x] Any patent application processing fees under 37 CFR 1 17.
[ ] The issue fee set in 37 CFR 1 18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b).
[ ] Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Dated July 7 2000 ALoudermilk Reg No 32,788
Attorney of Record: Alan R. Loudermilk

CERTIFICATE OF "EXPRESS MAIL" UNDER 37 CFR 1.10

"EXPRESS MAIL" Mailing Label Number EL401103166US Date of Deposit July , 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1 10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D C 20231

Maria Margaretich
(Typed or Printed Name of Person Mailing Paper or Fee) (Signature of Person Mailing Paper or Fee)

APPLICATION TRANSMITTAL

Attorney's Docket No. 802-001 **PATENT**

☒ Applicant ☐ Patentee

☐ Application No. ☐ Patent No.

☒ Filed on July 7, 2000 ☐ Issued on

Title: REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) and 1.27(c))—SMALL BUSINESS CONCERN

I hereby state that I am

☒ the owner of the small business concern identified below:

☐ an official of the small business concern empowered to act on behalf of the concern identified below:

Name of Small Business Concern 802 Systems, Inc.

Address of Small Business Concern 1580 Oakland Road, San Jose, CA 95131

I hereby state that the above identified small business concern qualifies as a small business concern, as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office under Sections 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third-party or parties controls or has the power to control both.

I hereby state that rights under contract or law have been conveyed to, and remain with, the small business concern identified above, with regard to the invention described in

☒ the specification filed herewith, with title as listed above.

☐ the application identified above.

☐ the patent identified above.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c), if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

☒ No such person, concern, or organization exists.

☐ Each such person, concern or organization is listed below.

NAME ______________________________

ADDRESS ______________________________

____

☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

NAME ______________________________

ADDRESS ______________________________

____

☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

NAME ______________________________

ADDRESS ______________________________

____

☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

☒ I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Name of Person Signing Andrew K. Krumel

Title of Person if Other Than Owner ______________________________

Address of Person Signing 3635 Pleasant Knoll Drive, San Jose, CA 95148

____

______________________________ Date July 7, 2000

Signature

Attorney's Docket No. 802-001 **PATENT**

☒ Applicant 802 Systems, Inc. □ Patentee

□ Application No. □ Patent No.

☒ Filed on July 7, 2000 □ Issued on

Title: Real Time Firewall/Data Protection Systems and Methods

STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(b))—INDEPENDENT INVENTOR

As a below named inventor, I hereby state that I qualify as an independent inventor, as defined in 37 CFR 1.9(c), for purposes of paying reduced fees to the United States Patent and Trademark Office under Sections 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office, with regard to the invention described in

☒ the specification filed herewith, with title as listed above.

□ the application identified above.

□ the patent identified above.

I have not assigned, granted, conveyed or licensed, and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c), if that person had made the invention, or to any concern that would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

☒ No such person, concern, or organization exists.

□ Each such person, concern or organization is listed below.*

**NOTE: Separate statements are required from each named person, concern or organization having rights to the invention as to their status as small entities. (37 CFR 1.27)*

FULL NAME ______________________
ADDRESS ______________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME ______________________
ADDRESS ______________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME ______________________
ADDRESS ______________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME ______________________
ADDRESS ______________________

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

(check the following item, if desired)

**NOTE:* *The following verification statement need not be made in accordance with the rules published on Oct. 10, 1997, 62 Fed. Reg. 52131, effective Dec. 1, 1997.*

**NOTE:* *"The presentation to the Office (whether by signing, filing, submitting, or later advocating) of any paper by a party, whether a practitioner or non-practitioner, constitutes a certification under § 10.18(b) of this chapter. Violations of § 10.18(b)(2) of this chapter by a party, whether a practitioner or non-practitioner, may result in the imposition of sanctions under § 10.18(c) of this chapter. Any practitioner violating § 10.18(b) may also be subject to disciplinary action. See §§ 10.18(d) and 10.23(c)(15)." 37 C.F.R. § 1.4(d)(2)*

☒ I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Andrew K. Krumel
Name of inventor

Signature of Inventor Date July 7, 2000

Name of inventor

Signature of Inventor Date

Name of inventor

Signature of Inventor Date

Name of inventor

Signature of Inventor Date

Ex.1002

REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

Field of the Invention

The present invention relates to computer security and data protection systems and methods, and more particularly to firewall and data protection systems and methods for filtering packets, such as from the Internet, in real time and without packet buffering.

Background of the Invention

The use of the Internet has exploded in recent years. Small and large companies as well as individual users are spending more time with their computers connected to the Internet. With the advent of Internet technologies, such as cable modems, digital subscriber lines, and other "broadband" access devices, users are connecting their computers to the Internet for extended periods of time.

Such extended or "persistent" connection to the Internet brings many advantages to users in immediate access to the content on the Internet through the use of email, search engines, and the like. Unfortunately, however, persistent access to the Internet exposes connected computers to potential security threats, where intruders and "hackers" may compromise proprietary systems, engage in information theft, or take control of the connected computers remotely. With more sophisticated tools at their disposal, hackers pose security and privacy risks to systems with persistent access to the Internet. Such security risks are even present for computers connected to the Internet for limited periods of time (such as through dial-up, modem connections), though to a lesser degree than the extended access computers.

There are currently many different types of firewall systems available on the market, including proxy servers, application gateways, stateful inspection firewalls, and packet filtering firewalls, each of which provides a variety of strategies and services for data protection. Conventional packet filters typically are computers, routers, or ASICs based on general purpose CPUs. They perform their filtering duties by receiving a packet, buffering the data until a determination can be made, and forwarding the packet as applicable for the particular system.

For example, a dual-homed, Linux-based filter with two network cards might receive a packet completely, evaluate whether it meets specific criteria, and transmit the packet on the other network card. In another example, a router designed for switch mode routing might begin buffering a packet until a decision is made, then forward the packet on the applicable interface while still receiving the packet. With most packet filters, software is used and data is buffered.

Sophisticated computer users working for medium- to large-sized companies have a variety of relatively expensive protection devices and tools at their disposal. Such devices and tools typically screen data packets received from the Internet with sophisticated software-based filtering techniques. Using relatively complex tools for software analysis, each packet is stored in a buffer and examined sequentially with software-based rules, which results in each packet being either accepted (and passed to the computer) or rejected (and disposed of by the software). This software often requires substantial computer knowledge and experience. Users of such devices and tools typically have an expertise in network administration or a similar field, so they can configure, optimize, and even build the complex filtering and security options provided by the software.

While such devices and tools can be quite effective in providing "firewall" protection for sophisticated users of large office systems, they pose several barriers to unsophisticated users of small office and home systems in the growing SOHO market. Current large office systems are expensive, difficult to set up, and require technical skills. What is needed for SOHO systems is a relatively inexpensive, uncomplicated, "plug and play" type of Internet protection system that can be easily connected and configured by relatively unsophisticated users.

Summary of the Invention

In accordance with the present invention, devices, methods and systems are provided for the filtering of Internet data packets in real time and without packet buffering. A stateful packet filtering hub is provided in accordance with preferred embodiments of the present invention. The present invention also could be implemented as part of a switch or incorporated into a router.

A packet filter is a device that examines network packet headers and related information, and determines whether the packet is allowed into or out of a network. A stateful packet filter, however, extends this concept to include packet data and previous network activity in order to make more intelligent decisions about whether a packet should be allowed into or out of the network. An Ethernet hub is a network device that links multiple network segments together at the medium level (the medium level is just above the physical level, which connects to the network cable), but typically provides no capability for packet-type filtering. As is known, when a hub receives an Ethernet packet on one connection, it forwards the packet to all other links with minimal delay and is accordingly not suitable as a point for making filtering-type decisions. This minimum delay is important since Ethernet networks only work correctly if packets travel between hosts (computers) in a certain amount of time.

In accordance with the present invention, as the data of a packet comes in from one link (port), the packet's electrical signal is reshaped and then transmitted down other links. During this process, however, a filtering decision is made between the time the first bit is received on the incoming port and the time the last bit is transmitted on the outgoing links. During this short interval, a substantial number of filtering rules or checks are performed, resulting in a determination as to whether the packet should or should not be invalidated by the time that the last bit is transmitted. To execute this task, the present invention performs multiple filtering decisions simultaneously: data is received; data is transmitted; and filtering rules are examined in parallel and in real time. For example, on a 100 Mbit/sec Ethernet network, 4 bits are transmitted every 40 nano seconds (at a clock speed of 25 MHz). The present invention makes a filtering decision by performing the rules evaluations simultaneously at the hardware level, preferably with a programmable logic device.

The present invention may employ a variety of networking devices in order to be practical, reliable and efficient. In addition, preferred embodiments of the present invention may include constituent elements of a stateful packet filtering hub, such as microprocessors, controllers, and integrated circuits, in order to perform the real time, packet-filtering, without requiring buffering as with conventional techniques. The present invention preferably is reset, enabled, disabled, configured and/or reconfigured with relatively simple toggles or other physical switches, thereby removing the requirement for a user to be trained in sophisticated computer and network configuration. In accordance with preferred embodiments of the present invention, the system may be controlled and/or configured with simple switch activation(s).

Accordingly, one object of the present invention is to simplify the configuration requirements and filtering tasks of Internet firewall and data protection systems.

Another object is to provide a device, method and system for Internet firewall and data protection that does not require the use of CPU-based systems, operating systems, device drivers, or memory bus architecture to buffer packets and sequentially carry out the filtering tasks.

A further object of the present invention is to perform the filtering tasks of Internet firewall protection through the use of hardware components.

Another object is to utilize programmable logic for filtering tasks.

Still another object is to provide a device, method, and system to carry out bitstream filtering tasks in real time.

Yet another object is to perform parallel filtering, where packet data reception, filtering, and transmission are conducted simultaneously.

A further object of the present invention is to perform the filtering tasks relatively faster than current state-of-the-art, software-based firewall/data protection systems.

Another object is to provide a device, method and system for firewall protection without the use of a buffer or temporary storage area for packet data.

Still another object of the present invention is to design a device, method and system that does not require software networking configurations in order to be operational.

A further object of the present invention is to provide a device, method and system for Internet firewall and data security protection that supports partitioning a network between client and server systems.

It is a yet another object of the present invention to provide a device, method and system for Internet firewall and data protection that supports multiple networking ports.

Another object is to maintain stateful filtering support for standard data transmission protocols on a per port basis.

Still another object of is to configure network functionality using predefined toggles or other types of physical switches.

A further object of the present invention is to conduct packet filtering without requiring a MAC address or IP address to perform packet filtering.

Yet another object of the present invention is to facilitate the shortest time to carry out bitstream filtering tasks.

Finally, it is another object of the present invention to be able to perform filtering rules out of order and without the current state-of-the-art convention of prioritizing the filtering rules serially.

Brief Description of the Drawings

The present invention may be more fully understood by a description of certain preferred embodiments in conjunction with the attached drawings in which:

FIGS. 1A and 1B are application level diagrams illustrating exemplary data protection systems in accordance with the present invention;

FIG. 2 is a flow diagram illustrating the components and operations of a preferred embodiment of the present invention;

FIG. 3 is a flow chart illustrating the basic functions of a repeater core and four filter levels in accordance with preferred embodiments of the present invention;

FIG. 4 is a diagram illustrating filtering functions of Level 2 filters in relation to the flow of packet data from internal and external networks in accordance with preferred embodiments of the present invention;

FIG. 5 is a flow chart illustrating packet filtering functions of Level 3 filters in accordance with preferred embodiments of the present invention;

FIG. 6 illustrates the rules by which TCP and UDP packets are evaluated in parallel in accordance with preferred embodiments of the present invention;

FIG. 7 is a diagram illustrating parallel rule evaluation for TCP and UDP packets in accordance with preferred embodiments of the present invention;

FIG. 8 is a flow chart illustrating packet filtering functions of Level 4 filters in accordance with preferred embodiments of the present invention;

FIG. 9 is a block diagram of the hardware components of a preferred embodiment of the present invention;

FIG. 10 is an illustration of an exemplary design of an external case in accordance with preferred embodiments of the present invention;

FIGS. 11 and 12 are flow diagrams illustrating SYN flood protection in accordance with preferred embodiments of the present invention; and

FIG. 13 is a flow chart illustrating the process of "garbage collection" in flood lists in accordance with preferred embodiments of the present invention.

Detailed Description of the Preferred Embodiments

The present invention will be described in greater detail with reference to certain preferred and alternative embodiments. As described below, refinements and substitutions of the various embodiments are possible based on the principles and teachings herein.

FIG. 1A and FIG. 1B illustrate the physical positioning of a stateful packet filtering hub in accordance with the present invention in two exemplary network configurations. The packet filtering hub of the illustrated embodiments preferably serves as an Internet firewall/data protection system (hereafter "data protection system").

With reference to FIG. 1A, in the illustrated embodiment data protection system 1 is coupled through a port to router 2 (or cable modem or other preferably broadband, persistent network connection access device), which is linked through a broadband connection to other computer systems and networks, exemplified by Internet 8 and Internet Service Provider (ISP) 10. Packets of data are transmitted from an ISP, such as ISP 10, via Internet 8 to router 2. The packets are transmitted to data protection system 1, which analyzes the packets in "real time" and without buffering of the packets, while at the same time beginning the process of transmitting the packet to the internal network(s) in compliance with the timing requirements imposed by the Ethernet or other network standards/protocols. If a packet of data satisfies the criteria of the rules-based filtering performed within data protection system 1, which is executed in a manner to be completed by the time the entire packet has been received by data protection system 1, then it is allowed to pass to hub 6 as a valid packet, which may then relay the cleared packet to computers 4a, 4b, 4c, etc. on the internal network. If a packet of data fails to meet the filtering criteria, then it is not allowed to pass as a valid packet and is "junked." Junking is defined as changing bits or truncating data, depending on the type of link, in a manner such that the packet is corrupted or otherwise will be detected by the receiving computers as invalid or unacceptable, etc. Without the intermediate positioning of data protection system 1, the packets would be transmitted directly to unprotected hub 6, thereby exposing computers 4a, 4b and 4c to security risks. It should also be noted that hub 6 is optional in accordance with the present invention; in other embodiments, data protection system 1 may be directly connected to a single computer or may have multiple ports that connect to multiple computers. Similar filtering is performed on packets that are to be transmitted from computers 4a, 4b, and 4c to Internet 8.

With reference to FIG 1B, in this illustrated embodiment data protection system 1 is coupled via one port to DSL router 2 (again, the network access device is not limited to a DSL router, etc.), which provides the broadband connection to Internet 8. As with the embodiment of FIG. 1A, data protection system 1 also is coupled to a number of computers 4a, 4b, etc., on the internal network, and serves to provide filtering for packets between computers 4a and 4b and Internet 8 in the manner described in connection with FIG. 1A. In this embodiment, data protection system 1 is also connected via another port to hub 6, which serves as the main point of contact for incoming connections from the Internet for bastion hosts 5a and 5b, etc. In accordance with this embodiment, packets are transmitted to router 2 and then to data protection system 1. If the packets are approved by data protection system 1 (i.e., passing the filtering rules/checks performed with data protection system 1 while the packet is being received and transmitted), then the packets are allowed to pass as valid packets to computers 4a, 4b and hub 6. (The rules-based filtering process of preferred embodiments of the present invention will be described in more detail hereinafter.) Hub 6 may relay the packets to other internal host computers 5a, 5b, etc., on the local area network (LAN). These computers may include, for example, a Web and FTP server 5a, or a streaming audio server 5b, etc. Thus, in accordance with the illustrated embodiment, packets that passed the filtering rules/checks are passed as valid packets to computers, such as protected internal host computer 4a, which as illustrated may be connected to printer 7. In this particular embodiment, a bastion port is provided that may be used to service more than one bastion host. In other embodiments, different network configurations may be utilized in accordance with the present invention.

FIG. 2 illustrates the general components and operations of certain preferred embodiments of the present invention. Connection to external network 12 is made by physical interface 14. Physical interface (or PHY) 14 preferably is implemented with commercially available, physical layer interface circuits, as are known in the art (such physical layer interface circuits may be off-the-shelf components, as specified in the Ethernet IEEE standard 802.3u.). At a minimum, the data protection system must contain two PHY interfaces, one for the Internet or other external network connection, and one (or more) for the internal network. It should be noted that, in preferred embodiments, PHY controllers are utilized, which implicitly assumes Ethernet-type connections. In other embodiments in accordance with the present invention, other types of PHY interfaces and controllers are utilized for different networking standards.

Repeater core 16 functions as an Ethernet repeater (as defined by the network protocols of the IEEE standard 802.3) and serves to receive packets from external PHY 14, reshape the electrical signals thereof, and transmit the packets to internal PHY 18, which is coupled to internal network 20. While the packet is being received, reshaped, and transmitted between PHYs 14 and 18, however, it is simultaneously being evaluated in parallel with filtering rules to determine if it should be allowed to pass as a valid packet (as will be described in greater detail elsewhere herein). As with the discussion regarding the PHY interfaces and controllers, changes in networking standards may alter the components functionality (such as the characteristics of repeater core 16), but not the basic parallel, real time packet filtering in accordance with the present invention. (In an alternate embodiment, for example, the data protection system may use switch logic or router logic; in full duplex, the same principles apply.)

The parallel filtering preferably consists of packet characteristics logic 22, packet type filters 26, and state rules filters 42. Packet characteristics logic 22 determines characteristics based on packet data (preferably in the form of 4-bit nibbles from PHY 14), whereas packet type filters 26 make filtering decisions generally based on packet type. State rules filters 42 perform rules- based filtering on several levels simultaneously. The results of filtering by packet type filters 26 and state rules filters 42 are combined by aggregator 24, which may be considered a type of logical operation of pass/fail signals (described in greater detail elsewhere herein). In preferred embodiments, if any one or more of the performed filtering rules indicates that the packet should be failed (or not allowed to pass as a valid packet), then the output of aggregator 24 is a fail; otherwise, the packet is allowed and the output of aggregator 24 is a pass. Thus, as packet data is being received and transmitted from PHY 14 to PHY 18 via repeater core 16, it is being evaluated in parallel via packet type filters 26 and state rules filters 42 (depending in part on packet characteristics determined by logic 22 from the data received from PHY 14). In accordance with the present invention, the results of filtering by packet type filters 26 and state rules filters 42 are provided to aggregator 24 by the time that the entire packet reaches repeater core 16, so that, based on the output of aggregator 24, the packet will either be allowed to pass as a valid packet or will be failed and junked as a suspect (or otherwise invalidated) packet.

Packet characteristics logic 22 receives packet data from PHY 14 and examines the packet data to determine characteristics, such as the packet type, datagram boundaries, packet start, packet end, data offset counts, protocols, flags, and receiving port. The packet type may include, for example, what are known in the art as IP, TCP, UDP, ARP, ICMP, or IPX/SPX. Such packet characteristics data is provided to packet type filters 26. Packet type filters 26 preferably make a decision about whether the packet should be passed or failed, with the result being transmitted to aggregator 24. In accordance with preferred embodiments, packet type filters 26 do not require the use of what may be considered an extensible rules system. The filters of packet type filters 26 preferably are expressed as fixed state machines or may be expressed using more flexible rules syntax. What is important is that packet type filtering is performed by filters 26 in the shortest time interval possible and in parallel with the packet data being received and transmitted to internal PHY 18, so that a pass/fail determination may be made prior to the time when the entire packet has been received by repeater core 16.

State rules filters 42 receive packet characteristics data from logic 22 and, based on this data as well as cached/stored connection and communication state information, executes a plurality of rules under the control of rules controller 28, preferably using a plurality of rules engines 36-1 to 36-N, so that a desired set of filtering decisions are promptly made and a pass/fail determination occurs before the entire packet has been received by repeater core 16. State rules filters 42 preserve a cache of information 30 about past network activity (such as IP addresses for established connections, port utilization, and the like), which is used to maintain network connection state information about which hosts have been exchanging packets and what types of packets they have exchanged, etc. Rules controller 28 preferably accesses rules map table 32 based on packet characteristics information, which returns rules dispatch information to rules controller 28. Thus, based on the connection state information stored in connection cache 30 and the characteristics of the packet being examined, rules controller 28 initiates filtering rules via a plurality of rules engines 36-1 to 36-N that simultaneously apply the desired set of filtering rules in parallel. (Preferably, N is determined by the number of rules that need to be performed in the available time and the speed of the particular logic that is used to implement state rules filters 42.)

As will be appreciated, while the packet pass/fail decision is being made in real time, and thus must be concluded by the time that the entire packet has been received, a large of number of filtering rules must be performed quickly and in parallel. Preferably, rules controller 28 utilizes a plurality of rules engines 36-1 to 36-N, which logically apply specific rules retrieved from corresponding storage areas 40-1 to 40-N. Rules controller 28, based on the connection state and packet characteristics, determines which rules should be run based on which information. The rules to be run are then allocated by rules controller 28 to the available rules engines 36-1 to 36-N. As each rules engine 36-1 to 36-N may be required to execute multiple rules in order to complete the filtering decision process in the required time, corresponding queues 34-1 to 34-N are preferably provided. Thus, rules controller 28 determines the list of rules that should be performed (again, depending on the stored connection state and packet characteristics data) and provides the list of rules (and accompanying information to carry out those rules) to the plurality of rules engines 36-1 to 36-N via queues 34-1 to 34-N. Rules engines 36-1 to 36-N, based on the information from the queues 34-1 to 34-N, look up specific rule information from storage areas 40-1 to 40-N, carry out the rules, and preferably return the results to rules controller 28. As the rules are essentially conditional logic statements that notify the data protection system how to react to a particular set of logical inputs, it has been determined that providing a plurality of rules engines may enable the necessary decision making process to quickly provide the outcome of the rules-based filtering by the time the entire packet has been received.

Still referring to FIG. 2, rules controller 28 preferably uses rules map table 32 to dispatch the rules to rules engines 36-1 and 36-N, so that a filtering decision may be reached in the optimal amount of time. In a preferred operation, each rules engine extracts a rule ID from its queue, looks up the rules definition in its own rules table 40-1 to 40-N, evaluates the rule, returns the result to rules controller 28, and looks for another rule ID in its queue 34-1 to 34-N. The results from packet type filter 26 and rules controller 28 are combined into one result via aggregator 24: pass or fail. If a decision is not reached before the end of the packet is transmitted, then in preferred embodiments the packet will be processed as an invalid packet and junked.

It should be appreciated that the data protection system must make a filtering determination before the current packet is completely transmitted. Since the networking standards impose strict timing thresholds on the transit delay of packets, filtering is performed in real time, in parallel and without buffering the packet. (The transit delay threshold is the time it takes to get from the transmitting station to the receiving station.) Given that a filtering decision must be made in real time (before the last bit is received and forwarded to the applicable interfaces), the filter rules are evaluated in parallel by rules engines that possess independent, direct access to the rules set collected in storage areas 40-1 and 40-N, which are preferably implemented as RAM tables. (In a preferred embodiment of the data protection system, the tables are implemented using on-chip, dual port RAM up to 4K in size. A programmable logic device, such as Xilinx Spartan II XC2S100, has 40K dual port synchronous block RAM. For example, an initial 110-bit segment of the rules controller RAM block may be a range table that delineates where each look up code begins and what the number of entries are.) Rules controller 28 dispatches the rules to each rules engine by placing a rules ID entry in a queue. Because each rules engine is assigned its own queue, a pipeline is created allowing the rules engine to continuously run and operate at maximum efficiency.

To operate efficiently the rules engines must also be capable of evaluating rules in any order. In accordance with the preferred embodiments, each rule has a priority and the highest priority result is accepted. Therefore, the rules must be evaluated in any order yet still obtain the same result, as if the rules were being evaluated serially from highest to lowest priority. This operation is accomplished in preferred embodiments by rules map table 32, which notifies rules controller 28 which rule is assigned to which rules engine. Thus, this decision is statically determined by the rules set and the number of rules engines. It should be noted that the rule set in general is greater than the number of rules engines.

FIG. 3 is a flow chart illustrating further aspects of preferred embodiments of the present invention. As previously described, preferred embodiments of the data protection system utilize programmable logic, or other suitable preferably hardware-based logic, to perform a large number of filter rules in parallel and at high speed. Such embodiments may be considered to provide an external interface, for instance, to the Internet, to external network 12, and one or more internal network interfaces, such as to internal network 20 and/or to bastion network 15 (see, for example, FIGS. 1A and 1B). As repeater core 16 (or the PHYs in FIG. 2) receives and transmits packet data, the packet is simultaneously subjected to a plurality of filtering rules. At step 44, the packet characteristics are determined (which, as previously described, may include protocol, addresses, ports, flags, etc.). The filtering rules are based on the packet characteristics, connection state information (depending upon the particular rules), and/or toggle or other physical switch state information. This filtering process may be represented by filtering steps 46, 48, 50 and 52, which, as depicted in FIG. 3, are performed at least in substantial part in parallel, and thus can make filtering decisions by the time the packet has been completely received.

As illustrated, after the packets are transmitted to repeater core 16, their characteristics are analyzed at step 44. Data packets generally consist of several layers of protocols that combine to make a protocol stack. Preferably, each layer of the stack is decoded and the information is passed to various filter blocks, as exemplified in steps 46, 48, 50 and 52. In accordance with the present invention, this filtering process is executed in parallel and in real time. In other embodiments, a variety of filter blocks or rules-based filters may be employed, incorporating parallel execution, real time filtering, etc., as may be necessary to complete the filtering decision in the required time.

Referring again to preferred embodiments illustrated in FIG. 3, Level 2 filters at step 46 may examine information in the link layer header for all incoming packets and decide whether a packet should be junked based on the packet protocol. Level 3 filters at step 48 may examine information in the networking layer headers. (For the IP protocol, these headers would equate to the ARP, RARP, IP, ICMP, and IGMP protocol headers.) While Level 2 filters preferably distinguish the packet type, Level 3 filters at step 48 and Level 4 filters at step 50 preferably distinguish IP datagram characteristics. Level 4 filters at step 50 preferably operate by examining IP, TCP and UDP headers along with data transmitted between the client and server processes, utilizing two techniques: stateful and non-stateful packet filtering. (Level 2, 3 and 4 filters are described in greater detail elsewhere herein.) Preferably a spoof check filter at step 52 detects whether the packet originated from an authorized IP address or not. To determine whether the packet should be allowed to pass as a valid packet, the filters must implement rules in parallel preferably based on programmable logic and register one of two values: pass or fail. After the values are registered, the outcome is collected in result aggregator 24, which logically combines the results to determine if the packet should be allowed to pass as a valid packet or should be denied as an invalid one. If the packet is passed, then repeater core 16 continues to send correct bits. If the packet is failed, then it is junked.

In accordance with preferred embodiments of the present invention as illustrated in FIG. 3, a spoof check is performed at step 52 on all packets entering a port. To prevent IP spoofing, the spoof check filtering of step 52 monitors IP addresses from the internal network and discards any incoming packets with IP source addresses that match internal IP addresses. A spoof check ensures that a host on one network is not trying to impersonate a computer on another network, such as a computer on the Internet assuming the IP address of a computer connected to an internal port. In accordance with preferred embodiments, spoofed packets are always junked by the data protection system. In such embodiments, the data protection system performs this check by keeping track of the IP addresses of packets arriving on the internal and bastion ports. The source and destination addresses of each packet are checked against the known port addresses to ensure they are valid for the appropriate port.

FIG. 3 also illustrates alarm controller 53, which preferably is coupled to result aggregator 24. Alarm controller 53, which could be a separate logic block or within the result aggregator, receives signals indicating when packets are being rejected, either directly from the logic performing the filtering or from result aggregator 24. As described in greater detail elsewhere herein, alarm controller 53 desirably is utilized to provide visual feedback of the system status or operation (such as whether the data protection system is under attack) via LED(s) 54 (or other light source, LCD or other type of alphanumeric or graphic display, etc.). For instance, a LCD may provide an additional mechanism for entering security configurations, such as specific protocols to allow a reference clock. Alarm controller 53 also may be coupled to an audio feedback device, such as speaker 55, which similarly may be used to provide audio feedback of the system status or operation. For example, if a packet is rejected, a first visual indication may be provided via LED(s) 54 (e.g., yellow light); if packets are being rejected in a manner or at a rate that suggests an internal computer is under attack, then a second visual indication may be provided via LED(s) 54 (e.g., a red light). Similarly, first and second tones or other audible indicators (different tones, volumes, sequences, etc.) may be provided via speaker 55 to indicate the detected condition). In preferred embodiments, such feedback, audio and/or visual, may maintain the alert state until reset by the user, such as by depressing a toggle. Thus, if the internal system has been determined to be under attack while the user is away, this fact will be made known to the user when the user returns and sees and/or hears the visual and/or audio feedback. It also should be noted that alarm controller 53 also may generate a UDP packet (indicated by the dashed line that is coupled to internal network 20) that informs the internal client computer of the attack or suspected attack, thereby providing an additional optional mechanism to inform the user of suspect activity.

FIG. 4 illustrates exemplary packet filtering functions of Level 2-type filtering in relation to the flow of packet data from internal and external networks. External PHY 12 receives packet electrical signals off the physical wire or other medium. Similarly, internal PHYs 18 and 58 receive packet electrical signals from internal network 20 or bastion network 15, respectively. Packet data comes in from one of PHYs 12, 18 or 58 to PHY controller 56. PHY controller 56 in general receives incoming data from network PHYs 12, 18 or 58, detects collisions, indicates the start and end of packet data, and forwards the packet data to other appropriate components of the data protection system (such as described herein). From PHY controller 56, data from the packet being received, along with information indicating which PHYs are active (i.e., on which PHY a packet is being received and to which PHYs the packet is being transmitted, etc.), and the packet is reshaped and transmitted in real-time via block 60 (i.e., the packet is not received into a buffer, after which it is sequentially processed to determine if the packet should be allowed to pass, etc., as in conventional firewalls). In the case of a packet received from Internet 8, the packet is received by PHY controller 56 from external PHY 12, and reshaped and transmitted in real-time to the internal PHY 18 and/or bastion PHY 58.

As will be appreciated, block 60 in essence performs the repeater functionality of passing the incoming data to the non-active PHYs after reformatting the preamble. Block 60 also preferably receives "junk" or "pass" signals from the filtering components and a collision detection signal from PHY controller 56. In preferred embodiments, a "jam" signal is propagated to each PHY upon detection of a collision. A packet is invalidated for all PHYs that belong to a network category that receives a "junk" signal. (For example, if the packet is invalidated for internal networks, then the packet is invalidated for all internal network ports.) Preferably, block 60 also receives a single output signal from result aggregator 24 for each PHY category (i.e., internal or external). As will be explained in greater detail hereinafter, result aggregator 24 generates the signals provided to block 60 depending on "junk" or "pass" signals from each filter component.

In accordance with the present invention, the packet is also simultaneously routed through a plurality of filtering steps. In the exemplary illustration of Level 2 filters in FIG. 4, the packet type is determined at step 64. At step 64, the network packet is examined to determine the enclosed Level 3 datagram type, such as ARP, RARP, IP, or IPX. This information is used to perform Level 2 filtering and to decide how to deconstruct the enclosed datagram to perform

Level 3 filtering. If an unknown packet type is received from the external network, then the packet preferably is junked if filtering is enabled. Unknown packet types received from the internal network preferably are forwarded to other hosts on the internal network and may be forwarded to the bastion port but are not forwarded to the external network.

If it is a known packet type, then it is routed through additional filtering steps based on particular packet protocols. In the illustrated embodiment, at step 66, if the packet is an Address Resolution Protocol (ARP) type packet, then it is passed. At step 68, if the packet is a Reverse Address Resolution Protocol (RARP) type packet and is from external PHY 12 and the op code is 3, then it is junked; otherwise, it is passed as indicated at step 70. As is known in the art, RARP generally is a protocol used by diskless workstations to determine their address; in accordance with preferred embodiments, RARP responses are the only RARP packets allowed to enter internal networks from external hosts. At step 72, if the packet is an Internet Protocol (IP) type packet, is from the external PHY and has been broadcast, then it is junked. (For example, broadcast packets from the external network preferably are not allowed; a broadcast packet is determined by examining the IP address or the physical layer address). Otherwise, the process proceeds to step 74. Step 74 preferably examines the IP header, which contains a protocol fragment where an application can place handling options. Certain options (such as the illustrated list) may be considered to provide internal, potentially sensitive network information, and thus packets that contain these options preferably are not allowed into the internal network. At step 74, if a handling option of 7, 68, 131, or 137 is present, then the packet is junked; if these options are not present, then the process proceeds to filter IP packet step 76 (exemplary details of step 76 are explained in greater detail hereinafter). If the packet passes the filtering rules applied in filter IP packet step 76, then the packet is passed, as indicated by step 78. If the packet does not pass the filtering rules applied in filter IP packet step 76, then the packet is junked.

As illustrated in FIG. 4, any signals indicating that the packet should be junked are provided to result aggregator 24, as indicated by line 73. The filtering results are thus routed to result aggregator 24, which records whether any of the packets were junked and thus invalidated. Result aggregator 24 provides one or more signals to the logic of block 60 at a time early enough so that a Frame Check Sequence (FCS) character may be altered to effectively invalidate the packet. Therefore, prior to complete forwarding of the packet, the filtering decision is made and the FCS character is either altered in order to ensure that it is corrupted, if the packet is to be junked, or forwarded unchanged, if the packet is to be passed. In effect, a system in accordance with the present invention acts like a hub or repeater by receiving packet nibbles (2 or 4 bits at a time) on one interface wire and by broadcasting those nibbles on other interfaces. Thus, the data protection system cannot make a decision about a packet before forwarding the nibbles on the non-receiving interfaces since this may result in an inoperable Ethernet network. If the system is enabled to filter a packet, it must still transmit data while receiving data to ensure the Ethernet network functions correctly and efficiently. The data protection system filters packets by transmitting a nibble on the non-receiving interfaces for each collected nibble on the receiving interface, but ensures that the Ethernet packet FCS character is not correct if the packet is suspect. Thus, the sending station may perceive that it successfully transmitted the packet without collision, but in fact all receiving stations will discard the corrupted packet. It should be noted that, in alternative embodiments, in lieu of or in addition to the selective alteration of a FCS or checksum-type value, the data contents of the packet also may be selectively corrupted in order to invalidate packets. In such embodiments, the packet contents are selectively altered to corrupt the packet (e.g., ensure that the checksum is not correct for the forwarded packet data or that the data is otherwise corrupted) if the packet did not pass the filtering rules.

FIG. 4 also illustrates physical switch or toggle 62, the state of which can be used to enable or control packet filtering in accordance with the present invention. The state of switch/toggle 62 is coupled to the data protection system in a manner to enable or disable packet filtering. In the illustrated example, the state of switch/toggle 62 is coupled to the logic of block 60; if, for example, packet filtering is disabled, then block 60 can receive and forward packets while disregarding the output of result aggregator 24 (alternatively, result aggregator 24 can be controlled to always indicate that the packet should not be invalidated, etc.). In other embodiments, the state of such a switch/toggle can control result aggregator 24 or all or part of the particular filtering steps. As will be appreciated in accordance with the present invention, the data protection system may be controlled and configured without requiring the implementation of complex software. The data protection system preferably utilizes toggle buttons or other physical switches to selectively enable various functions, such as Internet client applications, Internet server applications, and filtering features. The system, for example, also may contain a button for retrieving updated core logic or filtering rules from a data source. The data source for such updating of the core logic may include a wide range of forms of digital media, including but not limited to a network server, a floppy disk, hard drive, CD, ZIP disk, and DVD.Configuration, therefore, may be determined by physical interface components attached or linked to the system .

Referring to FIG. 5, additional details of preferred filter IP packet step 76 will now be described. FIG. 5 is a flow chart illustrating the packet filtering functions of the Level 3 filters first illustrated in FIG. 3. At step 81, the Level 3 filtering processes determine the IP datagram characteristics, which preferably include: datagram type (ICMP, IGMP, TCP, UDP, unknown); source and destination IP addresses; fragment offset; and fragment size. Based on the IP datagram characteristics, further filtering operations are performed. Preferred functions for Level 3 filtering will now be described in greater detail.

At step 80, if the IP datagram type is unknown, then the fail signal is set, sending a signal to the result aggregator that the packet should be invalidated. At step 82, if the IP datagram type is Internet Group Management Protocol (IGMP), then the fail signal is set, preventing IGMP packets from passing. At step 84, if the type is Internet Control Message Protocol (ICMP) and the packet is from the external PHY, then the filtering proceeds to step 88. At step 84, if the type is ICMP and the packet is not from the external PHY, then the packet is passed as indicated by step 86. At step 88, if the type is ICMP, and the packet is from the external PHY and does not contain a fragment offset of 0, then the fail signal is set, preventing fragmented ICMP packets from passing, as indicated by step 90; otherwise, the filtering proceeds to step 92. At step 92, if the type is ICMP, the packet is from the external PHY and contains a fragment offset of 0, then the packet type is further evaluated for request and exchange data. This data preferably includes one of the following ICMP message types: 5 for redirect; 8 for echo request; 10 for router solicitation; 13 for timestamp request; 15 for information request; or 17 for address mask request. Accordingly, if the packet type satisfies the criteria for step 92, then the fail signal is set as indicated by step 96. Otherwise, the packet is allowed to pass, as indicated by step 94. As will be appreciated, the ICMP filtering branch serves to keep potentially harmful ICMP packets from entering from the external network. (The listed message types represent an exemplary set of ICMP packets that may expose the internal network topology to threats or cause routing table changes.)

If IP datagram characteristics indicate that the packet is a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) packet, then the filtering proceeds to step 98. At step 98, it is determined whether the packet is a fragment 0 packet. If it is not, then the packet is allowed to pass, as indicated by step 100. This filtering process follows the convention of filtering only the first fragments, as subsequent fragments will be discarded if the first one is not allowed to pass; in other words, the data protection system ignores all but the first packet of a TCP or UDP datagram. At step 104, if the packet is TCP or UDP and is a first fragment packet, then it is determined whether a proper protocol header is included in the fragment; if it is not, then the fail signal is set as indicated by step 102 (in the illustrated embodiment all TCP and UDP packets that have improper headers are junked). If the packet is TCP or UDP, is a first fragment, and a proper protocol header is included in the packet, then the filtering proceeds to step 106 (further exemplary details of which will be described in connection with FIG. 6).

FIG. 6 is a flow chart that illustrates a preferred example of how TCP and UDP packets are evaluated in parallel in accordance with the present invention (see, e.g., the multiple rules engines and related discussion in connection with FIG. 2 and the Level 4 filters of FIG. 3). As is known, TCP and UDP are host-to-host protocols located in the Transport Layer of the protocol stack. FIG. 6 illustrates how packet data 108 is unbundled and decoded for packet characteristics at step 110 (e.g., IP addresses, ports, flags, etc.) as well as for packet type and PHY activity at 112 (i.e., whether it is an internally generated packet or an externally generated one). In the preferred embodiments, the packets are evaluated in parallel according to the following rules.

As indicated at step 114, if the internal port number is 68 and the external port number is 67, then the packet is passed, regardless of whether it originated on the internal network or the external network. As indicated at step 116, if the packet type is TCP, the server-mode is enabled (such as may be controlled by a toggle or other physical switch), the external PHY is active, and the internal port number is 80, then the packet is passed to the internal network(s). (The server mode is explained in greater detail in connection with FIG. 7 below). As indicated at step 118, if the packet type is TCP and either the Acknowledge ("ACK") bit or Final ("FIN") bit is set, then the packet is passed, regardless of whether it originated on the internal network or the external network. As indicated at step 120, if the packet type is TCP and an internal PHY is active, then the packet is passed to the external network. As indicated at step 122, if the packet type is UDP, an internal PHY is active, and the external port number is 53, then the packet is passed to the external network and the communication state (e.g., source and destination port numbers) is stored as indicated by comm or communication state store 124. As indicated at step 126, if the packet type is UDP, the external PHY is active and the external port number is 53, then the packet is passed to the internal network(s) if there is a match in the communication state. As indicated at step 128, if the packet type is TCP, an internal PHY is active, the external port number is 21, the Synchronize Sequence Numbers ("SYN") bit is not set but the ACK bit is set, and the packet is a PORT command, then the packet is passed to the external network and the client (internal network) active port is determined and the communication state is stored. As indicated at step 130, if the packet type is TCP, the external PHY is active, the external port number is 20, and the SYN bit is set but the ACK bit is not set, then the packet is passed to the internal network(s) if there is a communication state match. As indicated at step 132, if all checks have been completed, then a complete signal is set, and signals indicative of whether the packet passes to internal or external network(s) as previously described are bitwise logically ORed to generate pass internal and pass external signals, as illustrated.

In preferred embodiments, if the completion signal is not generated by the time that the packet has been completely received, then the packet is junked. It should be noted that the use of such a completion signal and packet junking can be extended to the diagrams and description, etc. of other figures, such as FIGS. 2, 3, 4, 5, 7 and 8. If the filtering process has not been completed by the time that the packet has been completely received, then the packet is preferably junked.

Referring now to FIG. 7, Level 4 filtering in accordance with the present invention will be further described. The embodiment of FIG. 7 is a table-based filter, which uses an approach similar to that described in connection with FIG. 2. This approach preferably utilizes a programmable logic device (PLD) that includes low latency, high-speed ROM and RAM blocks.

As previously described, Level 4 filtering is based on TCP and UDP packet characteristics, the determination of which is illustrated in FIG. 7 by block 133. TCP and UDP characteristics, as noted elsewhere herein, may include not only source and destination port numbers, but also the state of the SYN, ACK, FIN and/or RESET flags in the case of TCP packets. The TCP/UDP characteristics are determined by the TCP/UDP header information. The TCP/UDP characteristics and active PHY information are used in the generation of a lookup code, which in the embodiment of FIG. 7 is coupled to rules dispatcher 134. Rules dispatcher 134 uses a lookup code to determine the filtering rules to be applied to a packet and then places the identifiers of the rules to be run in queues 138-1 to 138-N for each of the rules engines 140-1 to 140-N. Mapping table 136 is coupled to and receives address data from rules dispatcher 134.

Mapping table 136 preferably is a ROM block that identifies the rules associated with each lookup code and the rules engine for which each rule is to be dispatched. The mapping data for the rules and rules engines are returned to rules dispatcher 134.

The identifiers of the rules to be run are dispatched by rules dispatcher 134 to the appropriate queues 138-1 to 138-N, which are preferably FIFO-type structures that hold the rule identifiers for corresponding rules engines 140-1 to 140-N. Queues 138-1 to 138-N not only enable rules dispatcher 134 to assign rules at maximum speed, but also allow each rules engine to retrieve rules as each one is evaluated. The rules engines 140-1 to 140-N are a plurality of filtering engines/logic that use a rule table to read a definition specifying whether a rule applies to a packet and whether the packet passes or fails the rule test. Rules tables 142-1 to 142-N preferably are ROM blocks that contain a definition of a set of filtering rules that are controllably run by the rules engines 140-1 to 140-N. Rules tables 142-1 to 142-N may contain different rules as may be appropriate to provide all of the rules necessary to adequately filter packets within the timing constraints imposed by the real time filtering of the present invention, and the speed of the hardware used to implement the data protection system.

In addition, as illustrated in FIG. 7, rules engines 140-1 to 140-N may receive as inputs signals indicative of a stored communication state, IP datagram characteristics, or physical switch/toggle states. As indicated by block 148, toggles may be utilized for a variety of features, such as enabling web client, web servers or other user-defined features. With at least some of the executed rules based on the stored communication state, stateful rules are implemented with the illustrated embodiment. A communication state table or cache is provided. A cache of communication state information between different hosts provides a set of bits that represent rule defined state information. For example, source and destination port information may be stored in the cache and used for state-dependent filtering.

In the illustrated embodiment, communication state information from rules engines 140-1 to 140-N may be provided to result aggregator 144, which in turn may store the communication state information to the communication state cache or storage area. Result signals, representing pass or fail of the packet based on the applied rules, also are provided to result aggregator 144. Result aggregator 144 combines the pass/fail results signals and provides a pass or junk signal or signals, which may be provided to the repeater core or to another result aggregator.

FIG. 8 illustrates an alternative preferred embodiment, in which the Level 4 filtering is implemented with a register-based filtering methodology. As with the Level 4 filtering of FIG. 7, both stateful filters 154 and non-stateful filters 153 may be implemented. As with the embodiment of FIG. 7, Level 4 filtering requires that TCP and UDP packet characteristics be determined, as illustrated by box 150. In addition to the Level 3 packet characteristics, Level 4 filters in accordance with this embodiment also require the source and destination port numbers and the TCP header values for the SYN, RST, FIN flags and the ACK value. This information preferably is used by both non-stateful and stateful filters 153 and 154. The implementation of the non-stateful filters is executed with a state machine or other logic preferably in the PLD that compares characteristics to the allowed non-stateful rules and makes a judgement as to whether the packet should be passed or failed. The non-stateful rules engine/logic uses a set of static rules to decide if a packet is allowed to pass through the firewall. These rules preferably are specified using a combination of control inputs, active PHY, and network packet characteristics.

Stateful filters are implemented to handle communication channel interactions that span multiple transmissions between hosts. The interactions typically occur at the Application Layer of the protocol stack, where examples may include FTP, RealAudio, and DHCP. These interactions may also take place at lower levels in the protocol stack, such as ARP and ICMP request/response.

In this embodiment, stateful filters 154 use protocol front-end and protocol back-end logic, along with a plurality of state registers to implement state-dependent filters. Each protocol that requires stateful packet filtering preferably has protocol handlers in the form of front-end and back-end logic, which decide when to issue a pass signal for a packet or store the identifying characteristics of a bitstream for later reference. Front-end logic 160-1 to 160-N monitors the network traffic to identify when the current communication state needs to be stored, deleted or updated. Front-end logic 160-1 to 160-N informs a corresponding back-end logic 158-1 to 158-N that a register will be allocated for storage for a bitstream. All store and delete state register requests are sent to back-end logic 158-1 to 158-N so it may update its internal information. Register controller 155 controls the actual selection of registers in state registers 156 and informs the corresponding back-end logic 158-1 to 158-N. Back-end logic 158-1 to 158-N monitors which state registers are dedicated to its protocol and issues a pass signal for packets that match an existing bitstream, as indicated by the appropriate packet characteristics and a matching state register. It should be noted that in alternate embodiments, different organizations of the functions of the programmable logic may be implemented in accordance with the present invention, incorporating various types of protocol handlers and state registers, as may be necessary.

Register controller 155 consolidates multiple store and clear signals from the various front-end logic 160-1 to 160-N and directs them to the appropriate registers in state registers 156. Register controller 155 also informs the various back-end logic 158-1 to 158-N which registers of state registers 156 are to be used for storage. The registers of state registers 156, under control of register controller 155, store the communication state of a bitstream; for example, a particular register records information about the two communication ends of the bitstream and also monitors each network packet to see if it matches the stored end-point characteristics. State registers 156 then sets a signal when its state matches the current packet characteristics. A "garbage collection" function also is implemented (as further illustrated in FIG. 13 below) to help free up state registers when the protocol information during the three-way handshake is not accessed within specific time frames.

As is known in the art, many protocols provide a way of identifying the end of a communication session. Accordingly, in preferred embodiments the data protection system detects when a stateful stream ends and frees up the associated state registers. Since clients and servers do not always cleanly terminate a communication session, the system preferably implements session time-outs to free state registers after a period of bitstream activity and to prevent indefinite state register exhaustion. If the network experiences a high rate of bitstreams requiring stateful inspections, the system's resources, which are allocated to tracking application data, can become exhausted. In this case, the system preferably resorts to allowing network traffic based on a set of static rules to pass through the non-stateful rules designed specifically for each protocol. This stateful to non-stateful transition is called "stateful relaxation." To maintain maximum security, a protocol handler that cannot gain access to an open state register will free up all of its state registers to help prevent other protocol handlers from entering into a relaxation state. The system will then wait for a state register to open, start a timer, and record protocol communication data in the state registers, while relying on the static rules. When the timer expires, the state filter will cease relying upon the static rules and approve packets solely on state register information.

FIG. 8 also illustrates toggle 152, which, in the additional illustrated example, selectively enables FTP (File Transfer Protocol) communications based on the switch state. Protocol back-end logic 158-1 to 158-N, as appropriate, utilize such toggle state information to selectively generate the pass/fail signals for the applicable protocols. For example, when the toggle switch is enabled, which is the default mode in most FTP client applications, it may send a signal to the internal FTP server to open a TCP connection to the client. Front-end logic 160-1 monitors the network traffic for data from the internal network, PORT command, source port number (greater than 1024) and destination port number (equal to 21). When this information is matched, front-end logic 160-1 requests state register controller 155 to store both the PORT command IP address and the port number as the destination end and the destination IP address, as well as store port 20 as the source end of a future communication packet. (In other embodiments, additional checks may be conducted to ensure the active connection IP address is the same as the current source IP address.) When back-end logic 158-1 recognizes the storage request, it waits for the allocated state register in state registers 156 to be sent by register controller 155. For example, when the state register number is set as register #1, then it records that register #1 is dedicated to allowing active FTP connections through the data protection system. Back-end logic 158-1 then waits for register #1 to signify that the current packet matches its stored state. When back-end logic 158-1 recognizes that the three-way TCP handshake has been completed for the new connection, it will notify front-end logic 160-1 to delete the state register. If the state register is junked, then back-end logic 158-1 records that register #1 is no longer dedicated to active FTP connections, allowing register controller 155 to allocate that register to a different protocol or network connection in the future.

FIG. 9 illustrates a preferred physical implementation of one embodiment of the present invention. In this embodiment, one external network connection and one internal network connection are provided. It will be appreciated that the components of FIG. 9 can be altered to implement, for example, bastion network connections and multiple internal network connections, etc.

The Internet connection, for example, via a cable modem, DSL router or other network interface, preferably is coupled with a physical cable to connector 168, which may be an RJ-45 connector. The signals received via connector 168 are coupled to and from PHY 170, which provides the physical interface for the data signals received from, or coupled to, the external network. Signals are coupled between PHY 170 and PLD 162, and signals are coupled between PLD 162 and PHY 172, which couples signals between connector 174 (which again may be an RJ-45 connector). The connection to the internal network may be made through connector 174.

In the preferred embodiment, PLD 162 implements the various levels of filtering as previously described. PLD 162 provides logic/hardware based, parallel filtering rules logic/engines, which make a decision about whether the packet should be allowed to pass or fail prior to the time that the packet is passed on by the repeater core portion of PLD 162 (as described elsewhere herein). The logic of PLD 162 to implement the filtering rules is programmed/loaded by controller 164, which may be a RISC CPU such as a MIPS, ARM, SuperH-type RISC microprocessor or the like. The PLD code preferably is stored in memory 166, which preferably is a re-programmable, non-volatile memory, such as FLASH or EEPROM. In this manner, the PLD code may be updated by reprogramming memory 166, and the updated PLD code may then be programmed/loaded in to PLD 162 under control of processor 164.

FIG. 9 also illustrates the use of LEDs 177, 178 and 179 to provide visual feedback of the data protection system status. In accordance with the present invention, the use of such displays or light sources may be used to convey various types of information to the user. For example, LEDs 177 and 179 may be provided to indicate that PHYs 170 and 172 are detecting an active network connection (and thus provide an indication that the network connections are present and functioning properly). LED 178 preferably provides alarm type information. For example, LED 178 may be provided in the form of a multi-color LED, which may provide a first colored light (e.g., yellow) if the data protection system has rejected one or more packets (thereby indicating that the system may be detecting an attack), and which may provide a second colored light (e.g., red) if the data protection system is continually rejecting packets or rejecting packets at a high rate (thereby indicating that the system is likely under attack). Such visual indicators, which may be coupled with audio feedback as described elsewhere herein, serve to inform the user that the user's computer or network may be under attack, thereby enabling the user to take further action, such as disconnecting from the network.

It should be noted that such visual feedback may be implemented in a variety of forms. In addition to multi-color or multiple LEDs or other lights sources or displays, a single LED could be provided, with the LED blinking at a rate that indicates the level of severity as predicted by the data protection system. For example, if no packets have been rejected, then the LED may be in an off or safe (e.g., green) state. If packets have been rejected but not on a continual or high rate basis, then the LED (e.g., red) may be controlled to blink on and off at a first, preferably lower speed rate. If packets are being rejected on a continual or high rate basis (or otherwise in a manner that that system believes is suspect), then the LED may be controlled to blink on and off at a second, preferably higher speed rate. Thus, the LED blink rate desirably may be controlled to blink at a rate that corresponds to the level of severity of the security threat that is determined by the data protection system. Optionally coupled with audio feedback, such visual indicators may provide the user with alarm and status information in a simple and intuitive manner.

As further illustrated in the preferred embodiments of FIG. 9, a variety of physical switches or toggles 176, 180, 181 and 182 may be coupled to PLD 162 or controller 164. As illustrated by update button 176, toggles may be used to control the updating of the PLD code (for instance, to reconfigure or update the system, providing updated filtering algorithms). As illustrated by buttons 180 and 181, toggles may be used to selectively activate/deactivate filtering steps depending on whether a protected computer is enabled to operate in either a server mode or client mode (the state of such toggles preferably being used to control filtering decisions made within the filtering logic). As illustrated by reset button 182, toggles may also be used to control the reset of the data protection system (for example, to cause the PLD code to be re-loaded, as when the system enters an inoperable state caused by power supply irregularities or other unusual circumstances). The use of such physical switches/toggles allows the data protection system to be controlled in a straightforward manner, simplifying the user operability of embodiments of the present invention.

With reference to FIG. 9, additional details of preferred update program and protocols will now be described. The data protection system may be controlled to operate in an update mode by pressing update button or toggle 176, which preferably is provided on an external case (further described in FIG. 10 below). In accordance with preferred embodiments, during the interval when the update button is pressed by the user and the update either completes or is canceled by the user, the data protection system will not forward any packets (i.e., filtering is not active, so packet transmission is blocked). The user may then run an update program (which may be a browser-based or stand-alone application) from an internal host computer.

In the illustrated embodiment, it is assumed that the user previously downloaded a system update or is downloading an update through a browser. The update program preferably breaks the update into 1K size packets and forwards them, using a limited broadcast destination address (for example, 255.255.255.255). The source and destination ports are set to a predetermined value, such as 1 (1-4 are currently unassigned according to RFC 1010), and an IP option is set in the IP header. The program data preferably is preceded by the system update header that has the following structure in the illustrated embodiment: ID (1)/count (1)/bit length (2). The numbers in parentheses represent the field size in bytes. The ID for the entire transaction remains unchanged, except for the count field increments for each packet. In a preferred embodiment, the data protection system may receive the packets in order and perform several checks, such as ensuring the ID and count fields are correct, verifying the UDP checksum, and storing the configuration data in non-volatile memory. Preferably, these checks may be controlled by controller 164. Thereafter, the updated PLD code may be loaded into the PLD, with the filtering operations being based on this updated code.

As a result of the parallel filter rules evaluation as previously described, packets do not need to be buffered, except, for example, to create octets that facilitate determining protocol elements. (As is known, data needs to be combined into 8-bit, 16-bit, or 32-bit words because header and packet data often exist in these sizes or straddle a 4-bit nibble boundary.) Instead of buffering each packet, the data protection system generates another distinct data packet or chunk. This process of packet generation occurs while a plurality of filtering rules are applied in real time and in parallel, producing improved data protection systems and methods.

FIG. 10 illustrates a preferred embodiment of an exemplary design of an external case of a data protection system in accordance with the present invention (it being noted that the particular switches, lights, etc., and their physical arrangements being exemplary). For example, external case 184 may be a molded plastic box in the shape of a "U" or folded tube as illustrated. The exemplary features of this external case may include ports, buttons (or toggle switches), LEDs, a clock, a removable logo disk, and a power supply connector. Home (internal) port 186, Internet (external) port 188, and power supply connector 190 are preferably located on the same side of external case 184 with power supply connector 190 set between the two ports. Home port 186 connects to the internal network via cable 192; Internet port 188 connects to the external network via cable 194. Power supply connector 190 is coupled to an external DC power supply via cable 193. The PHY of each port preferably is coupled to a link LED, such as previously described: home port 186 is coupled to internal link LED 196; and Internet port 188 is coupled to external link LED 198. The link LEDs are thus coupled to the internal and external PHYs, respectively, and serve to indicate whether the PHYs have detected a network connection.

In the preferred embodiment, on the internal network side of the U-shaped case, server mode button 200 is provided to allow the user to selectively enable filtering depending on whether the internal computer is allowed to operate in a server mode (thus, the state of server mode button 200 may be used to selectively control filtering decisions based on whether internal computers will be operating in a server mode, etc.). Server mode button 200 preferably includes server mode LED 202. When illuminated (e.g., green), server mode LED 202 indicates that the internal computers are enabled to operate in a server mode and the filtering decisions will be controlled accordingly. Server mode button 200 and server mode LED 202 are coupled to PLD 162, as described in FIG. 9. In the illustrated embodiment, parallel to server mode button 200 on the external side of the case is alert button 204, which contains alert LED 206. Alert LED 206 is coupled to alarm controller 53, which preferably is implemented as a part of PLD 162 (as illustrated in FIGS. 3 and 9, respectively). Alert LED 206 may contain a single or multi-colored LED, which, when illuminated, indicates the data protection system is under attack and is rejecting suspect packets. The data protection system preferably registers the frequency of attacks and sends signals to alert LED 206 based on such information. In a preferred embodiment, alert LED 206 may contain a LED (e.g., red), which remains consistently illuminated during irregular attacks or blinks at regular intervals under heavy attack. In another preferred embodiment, alert LED 206 may contain a multi-colored LED, which similarly indicates when the system is under attack and is rejecting packets. However, with a multi-colored LED, the increase in frequency or intervals of attacks may be indicated by a change in color: for example, green (indicating no registered attacks by suspect packets) to yellow (indicating a few irregular attacks) to red (indicating more frequent attacks) to blinking red (indicating a heavy attack). The alert alarm may be reset by depresseing alert button 204.

In a preferred embodiment, speaker 55 or some form of audio transducer may be coupled to alarm controller 53 to also indicate the presence or severity of attacks (as described in connection with FIG. 3). For example, when the data protection system is under heavy attack and alert LED 206 is blinking (e.g., red), an alarm signal may be transmitted to speaker 55 to emit audio information to indicate a suspected severe attack or emergency. Alarm-type information may also be coupled to the internal network (such as via a UDP packet, as described elsewhere herein), and thus transmit alarm information over the network to a software interface on the desktop. In other embodiments of the data protection system, an array of different features, including buttons, LEDs, alarms, and graphical user interfaces, may be utilized to indicate the class, frequency and severity of attacks on the system.

Adjacent to alert button 204 on the external network side of the case preferably is protection button 208, which is coupled to protection-on LED 212 and protection-off LED 214. When protection button 208 is set in the "on" position, protection-on LED 212 preferably illuminates red and the filtering system is enabled; when protection button 208 is set in the "off" position, protection-off LED 214 preferably illuminates yellow and the filtering system is disabled. As will be appreciated, the particular colors utilized are exemplary.

Still referring to FIG. 10, power LED 210 is coupled in a manner to indicate power is being provided via power supply connector 190. When power LED 210 is illuminated (e.g., green), it indicates the power supply is providing power to the data protection system. It should be noted that in the illustrated embodiment, the present invention does not require an on/off switch for the power supply because the system is designed to be enabled once a DC power supply is provided. As previously described, reset button 182 is coupled to controller 164 and may be used to initiate loading or re-loading of the PLD code.

Adjacent to reset button 182 is update button 176, which is coupled to update-enabled LED 218 and update-disabled LED 220, as well as PLD 162 (as illustrated in FIG. 9). As previously described, an update program preferably is utilized to update the logic programming and rules tables. Preferably, after pressing update button 176, the data protection system is automatically restarted, causing the new PLD code to load. The load version bit preferably will be set in the flash configuration header, which causes the system to load using the new program file. In a preferred embodiment, update-enabled LED 218 will illuminate in green to indicate the data protection system is ready to receive the new updated programming. After the update begins, the system may continually flash update-enabled LED 218 until the successful completion of the update; LED 218 is extinguished upon successful completion of this process. However, if an update is incomplete and fails to occur, update-failed LED 220 may illuminate in red and blink. The user extinguishes LED 220 by pressing the update button a second time. If possible, the data protection system may generate a UDP packet to inform the internal client of the reason for the failure. As an additional example, if the system contains an LCD, it may display an error code. The data protection system will continue to filter packets after update-failure LED 220 is extinguished. LED 216 is preferably provided to be illuminated when the system is operating and filtering packets in the manner described.In addition to the various toggles in a preferred embodiment of the present invention, additional types of components may be used to enter filtering criteria and/or selectively enable or control the filtering, such as a LCD display coupled with input buttons, a touch screen, an audio input for speech recognition, and/or a clock. Thus, filtering decisions may be made based on such switch inputs, audio commands, time of day or date, etc.

As further illustrated in FIG. 10, a removable logo disk 222 may be located on a preferred embodiment of the case. This removable disk may include a company logo, registered trademark, and/or other copyrighted material that may be valuable for branding and marketing the data protection system under a separate wholesaler. The disk is thus removable and replaceable for a variety of branding purposes.

In an alternate embodiment, security levels switch 223 may be implemented to prevent stateful relaxation, in which a stateful to non-stateful transition may occur during state register exhaustion. As illustrated in FIG. 8, security levels switch 223 may preferably include a variety of features that prevent stateful relaxation, such as timers, protocol-specific filters, and other rules-based filters. For example, switch 223 may be configured for three positions: one which allows FTP protocols, but does not allow DNS protocols; another which allows DNS protocols, but does not allow FTP; and a third which may serve as an emergency back-up feature and block all network traffic.

In other embodiments, different designs may be used in accordance with the present invention, incorporating various buttons, switches, LEDs, ports, cables, slots, connectors, plug-ins, speakers, and other audio transducers, which in turn may be embodied in a variety of external case shapes, as may be necessary. As will be appreciated, the filtering criteria may be dependent upon physical switch position, packet characteristics, clock time, and/or user-specified criteria, all of which may be entered through one or more physical input device(s). Such a physical input device, for example, may be comprised of one or more switches (such as a toggle switch, button switch, or multi-state switch), an audio input device, or display input device. The user-specified criteria may be transferred from the configuration software to the system using a network protocol, infrared port, or cable attachment.

FIGS. 11and 12 are flow diagrams illustrating examples of "SYN flood" protection in accordance with preferred embodiments of the present invention. Such SYN flood protection is optionally provided as an additional computer protection mechanism in accordance with certain preferred embodiments.

As is known in the art, SYN flood is a common type of "Denial of Service" attack, in which a target host is flooded with TCP connection requests. In the process of exchanging data in a three-way handshake, source addresses and source TCP ports of various connection request packets are random or missing. In a three-way handshake, the system registers a request from an IP address, then sends a response to that address based on its source, and waits for the reply from that address.

As illustrated in FIG. 11, the data protection system waits for a packet from external PHY 14 (as illustrated in FIG. 2) at step 224. When the system receives a packet from the external PHY, it compares the IP address and ports to the flood list entries at step 226, then proceeds to step 228. At step 228, the system determines whether the packet type is TCP, the ACK bit is set, and the packet matches an entry in the flood list. If these criteria are met, then the system proceeds to step 230, where the packet is removed from the flood list. If the packet is removed from the flood list, then the system returns to step 224 and waits for the next packet from the external PHY. Otherwise, if the criteria at step 228 are not met, then the system proceeds to step 232, where the system determines whether the packet type is TCP, the SYN bit is set and the ACK bit is not set. If the criteria at step 232 are met, then the system proceeds to step 234; otherwise, the system returns to step 224. At step 234, the system determines if the flood list is full and if the client has reached the maximum connection requests. If the flood list is not full, then the system returns to step 224 to wait for more packets from the external PHY. However, if the flood list is full at step 234, then the system proceeds to step 236, where the packet is junked and the system returns to step 224.

As illustrated in FIG. 12, the data protection system also waits for a packet from internal PHY 18 (as illustrated in FIG. 2) at step 238. When the system receives a packet from the internal PHY, it accesses the flood list location and writes the bits into the list, swapping ACK bits as well as MAC, IP and port addresses. The system then proceeds to step 242, where it determines if the packet type is TCP and the SYN and ACK bits are set. If the criteria at step 242 are met, then the system proceeds to step 244; if not, then the system returns to step 238 and waits for another packet from the internal PHY. At step 244, the SYN flag is unset and number 1 is added to the new ACK number. The system then proceeds to step 246, where it determines if the flood list is full. If the flood list at step 246 is full, then the Reset flag is set, the checksums for TCP, IP and Ethernet protocols are recalculated, and the Reset packet is transmitted. The system then returns to step 238. However, if the flood list at step 246 is not full, then the system proceeds to step 248, where the checksums for TCP, IP and Ethernet protocols are recalculated and the ACK packet is transmitted. The system then proceeds to step 252, where the recalculated packet is added to the flood list and the system returns to step 238, where it waits for another packet from the internal network.

In accordance with the present invention, SYN flood protection as described does not require either an IP or MAC address. The data protection system uses the destination MAC address as the source Ethernet address when framing the response packet that completes the TCP three-way handshake. In all cases, when forming the new packet, the source and destination header information is swapped, so that the source IP address and port become the destination IP address and port. It should be appreciated that SYN flood protection, as preferably implemented by the system, does not buffer the incoming packet, but builds the TCP response packet in real-time. The new TCP packet is placed in a queue for transmission at the earliest time possible based on the rules dictated by the link level protocol..

As illustrated in FIG. 13, in order to keep the flood lists from filling up with stale entries, the data protection system must free up state registers when the protocol information is not accessed within specific time frames, such as when a three-way handshake is initiated by a client, but the transaction is not closed. After the system receives a packet, it for one second at step 254, then proceeds to step 256, where the packet is checked against each flood list entry and passed to step 258. At step 258, the system checks for stale entries (or garbage collection) in the flood lists and proceeds to step 260, where it determines if time has expired. If time has expired at step 260, then the packet proceeds to step 262; if not, then the system returns to step 256 to check each flood entry list again. At step 262, the system unsets the ACK bit and sets the Reset flag, adds 1 to the sequence number, recalculating the checksums, and then recalculates the checksums for TCP, IP, and Ethernet protocols. The system proceeds to step 264, where the Reset packet is transmitted; it then proceeds to step 266 and removes the packet from the flood list. The system then proceeds to step 256. lt should be noted that if time expires for the request, then the system sends the Reset flag, terminating the connection.

Although the invention has been described in conjunction with specific preferred and other embodiments, it is evident that many substitutions, alternatives and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, the invention is intended to embrace all of the alternatives and variations that fall within the spirit and scope of the appended claims. For example, it should be understood that, in accordance with the various alternative embodiments described herein, various systems, and uses and methods based on such systems, may be obtained. The various refinements and alternative and additional features also described may be combined to provide additional advantageous combinations and the like in accordance with the present invention. As will also be understood by those skilled in the art based on the foregoing description, various aspects of the preferred embodiments may be used in various subcombinations to achieve at least certain of the benefits and attributes described herein, and such subcombinations also are within the scope of the present invention. All such refinements, enhancements and further uses of the present invention are within the scope of the present invention.

What is claimed is:

1. A method for communicating data between an external computing system and an internal computing system over a packet-based network, comprising the steps of:

receiving a communication packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

2. The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or

audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. The method of claim 25, wherein the at least one light source comprises an LED.

30. The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. A system for filtering packets of data between at least an external network and an internal network, comprising:

a first interface circuit for coupling data to and from the external network;

a second interface circuit for coupling data to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria.

32. The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the packet has been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. The system of claim 55, wherein the at least one light source comprises an LED.

60. The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

Abstract

Methods and systems for firewall/data protection that filters data packets in real time and without packet buffering are disclosed. A data packet filtering hub, which may be implemented as part of a switch or router, receives a packet on one link, reshapes the electrical signal, and transmits it to one or more other links. During this process, a number of filters checks are performed in parallel, resulting in a decision about whether each packet should or should not be invalidated by the time that the last bit is transmitted. To execute this task, the filtering hub performs rules-based filtering on several levels simultaneously, preferably with a programmable logic or other hardware device. Various methods for packet filtering in real time and without buffering with programmable logic are disclosed. The system may include constituent elements of a stateful packet filtering hub, such as microprocessors, controllers, and integrated circuits. The system may be reset, enabled, disabled, configured, and/or reconfigured with toggles or other physical switches. Audio and visual feedback may be provided regarding the operation and status of the system.

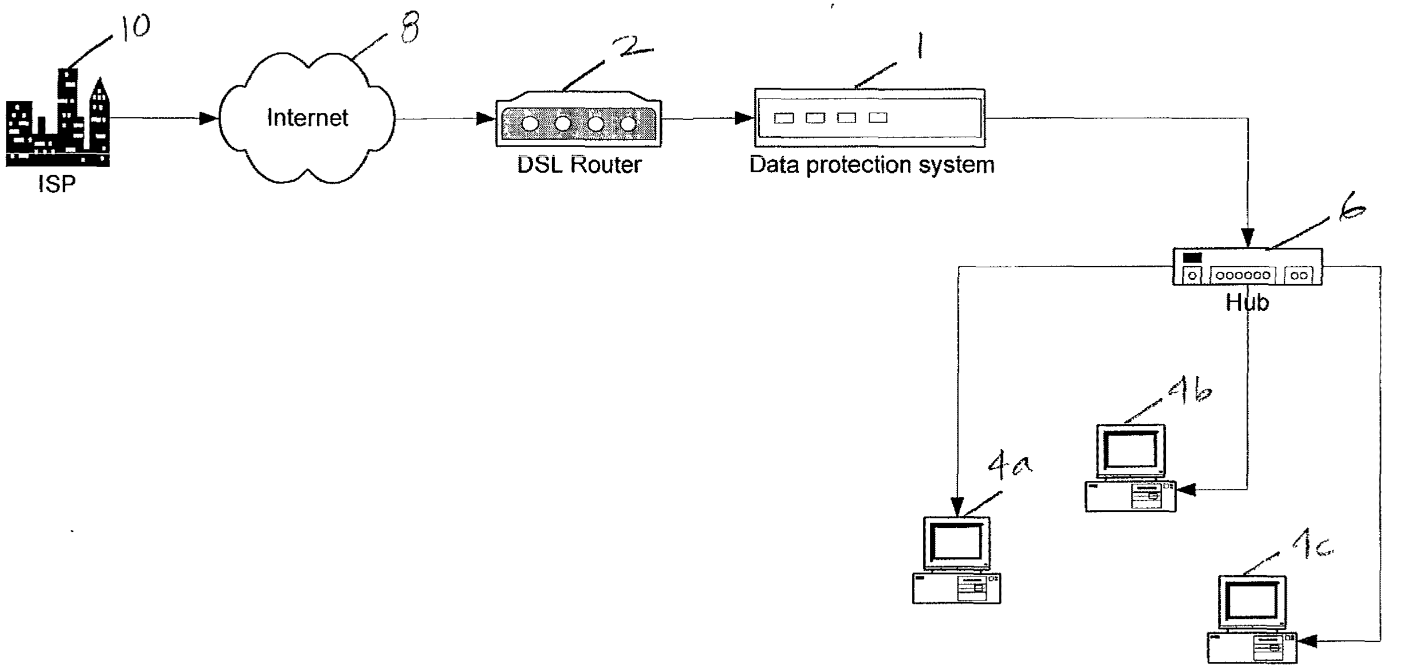

10
ISP
8
Internet
2
DSL Router
1
Data protection system
6
Hub
4b
4a
4c

FIG. 1A

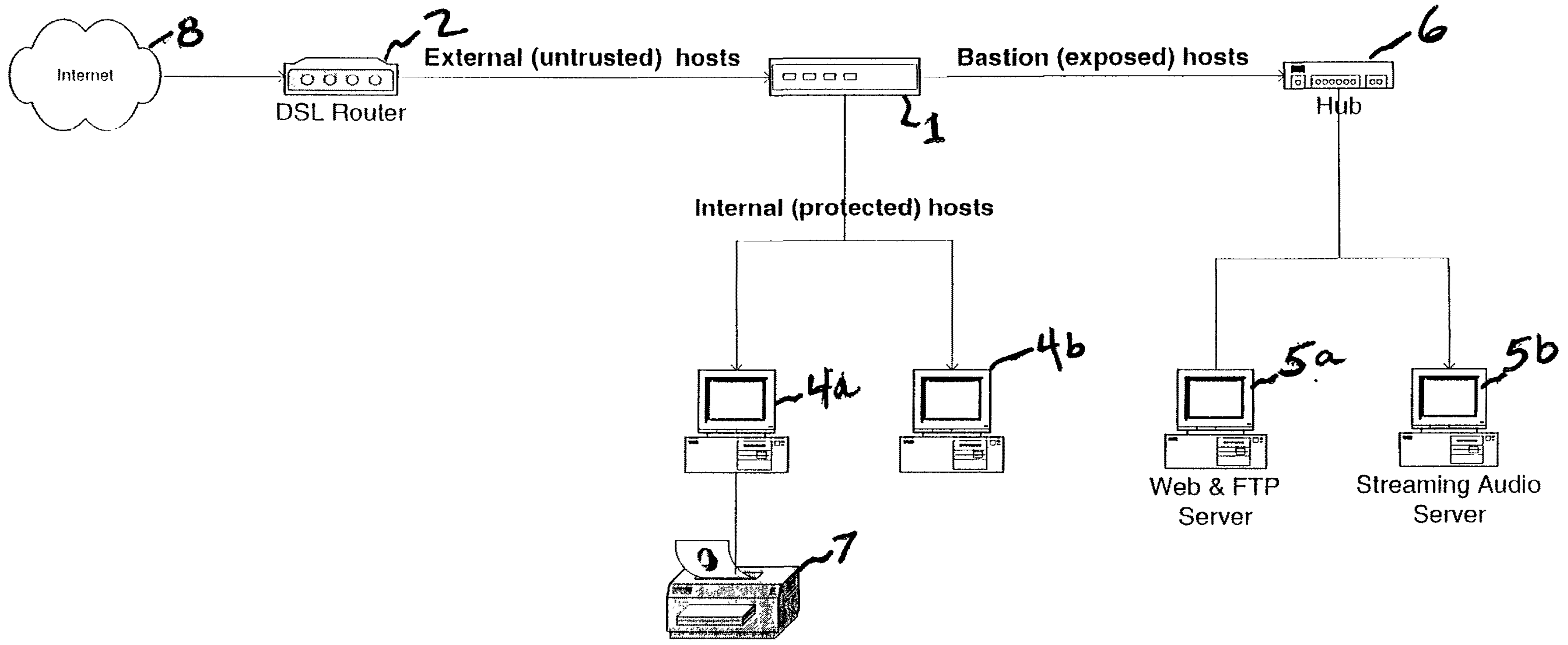

Internet
8
2
DSL Router
External (untrusted) hosts
1
Bastion (exposed) hosts
6
Hub
Internal (protected) hosts
4a
4b
7
5a
Web & FTP Server
5b
Streaming Audio Server

FIG. 1B

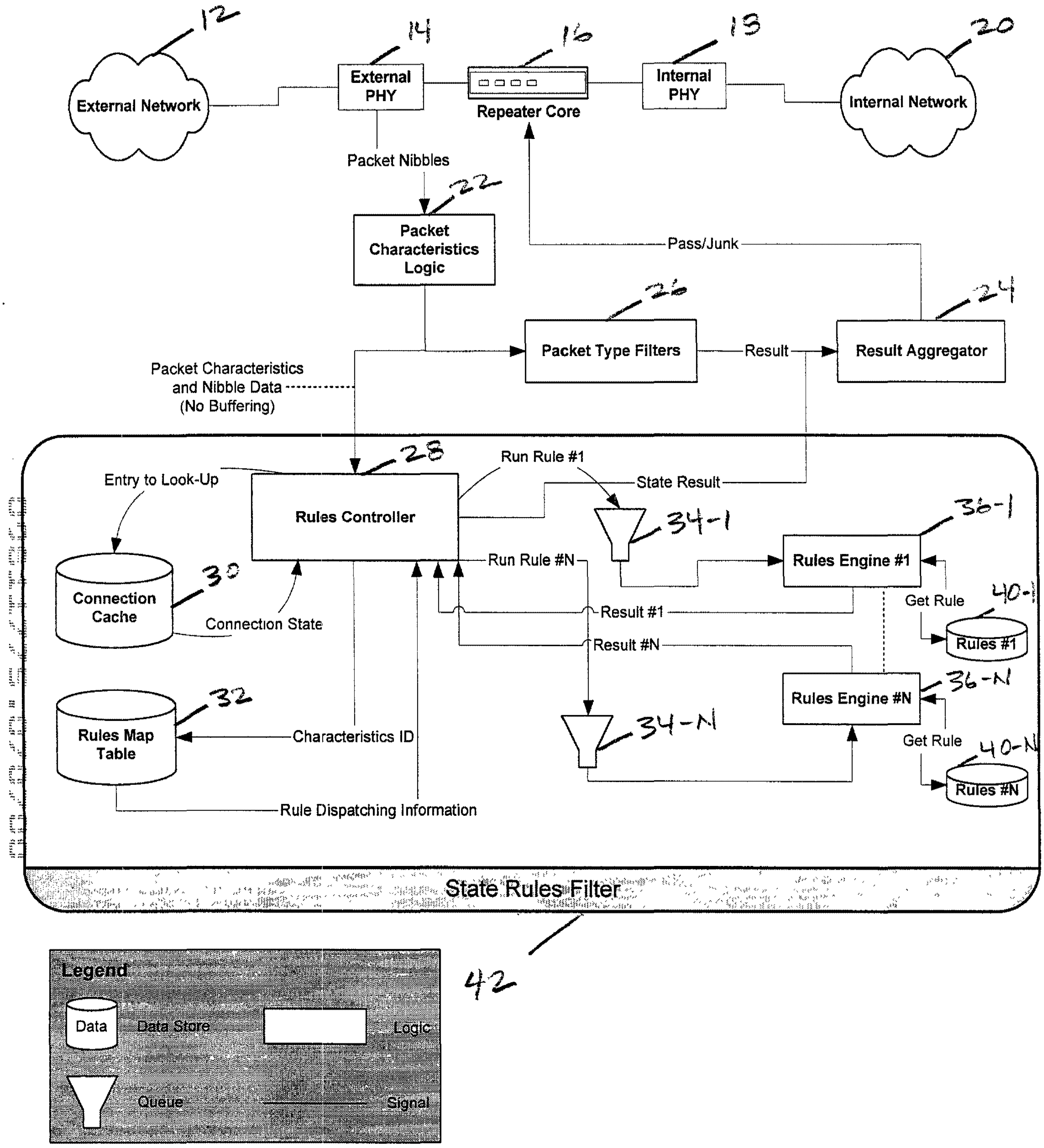

12
External Network
14
External PHY
16
Repeater Core
18
Internal PHY
20
Internal Network
Packet Nibbles
22
Packet Characteristics Logic
Pass/Junk
26
Packet Type Filters
Result
24
Result Aggregator
Packet Characteristics and Nibble Data (No Buffering)
28
Rules Controller
Run Rule #1
State Result
Entry to Look-Up
34-1
36-1
Rules Engine #1
Connection Cache
30
Connection State
Run Rule #N
Result #1
Get Rule
40-1
Rules #1
Result #N
Rules Map Table
32
Characteristics ID
34-N
Rules Engine #N
36-N
Get Rule
40-N
Rules #N
Rule Dispatching Information
State Rules Filter
42
Legend
Data
Data Store
Logic
Queue
Signal

FIG. 2

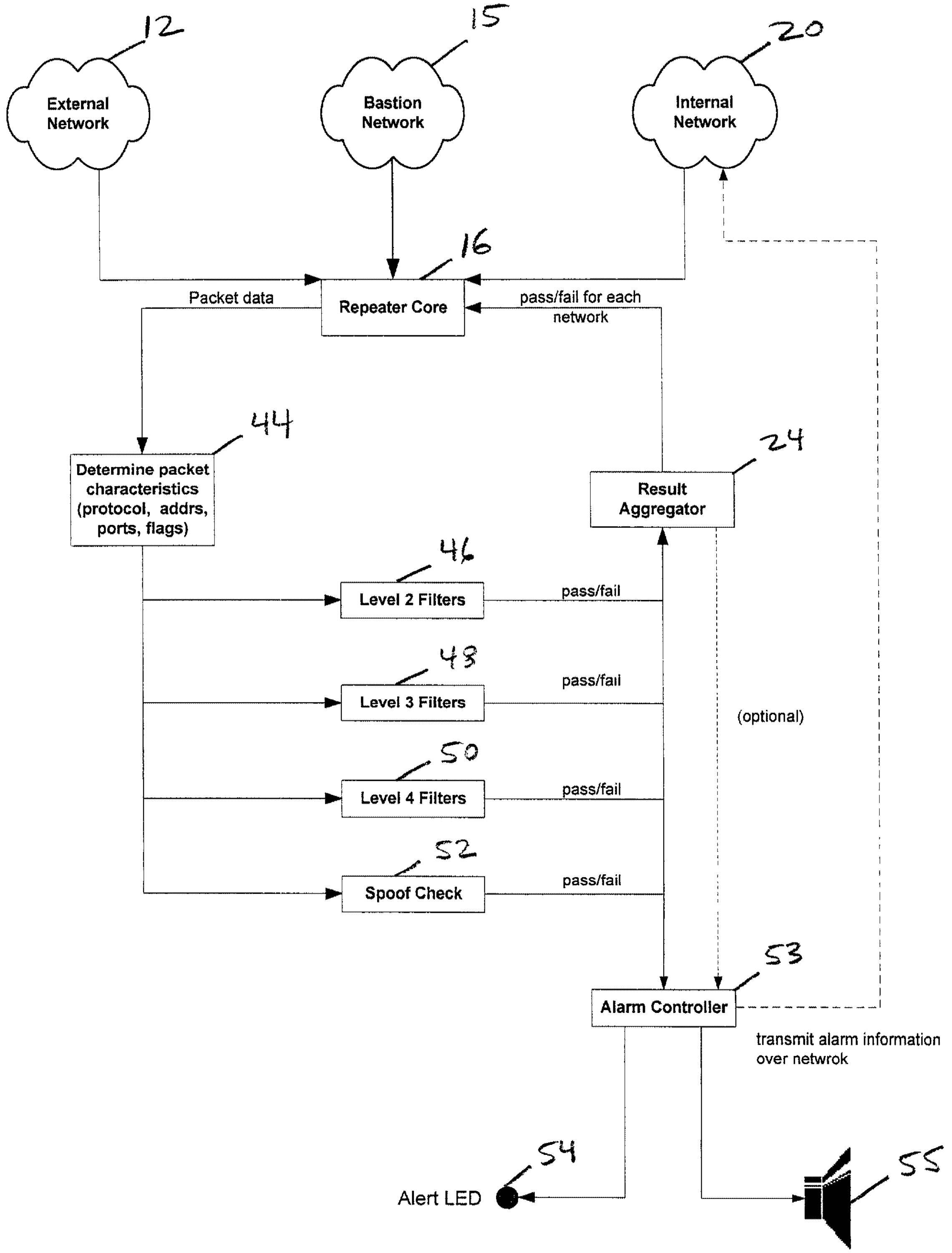
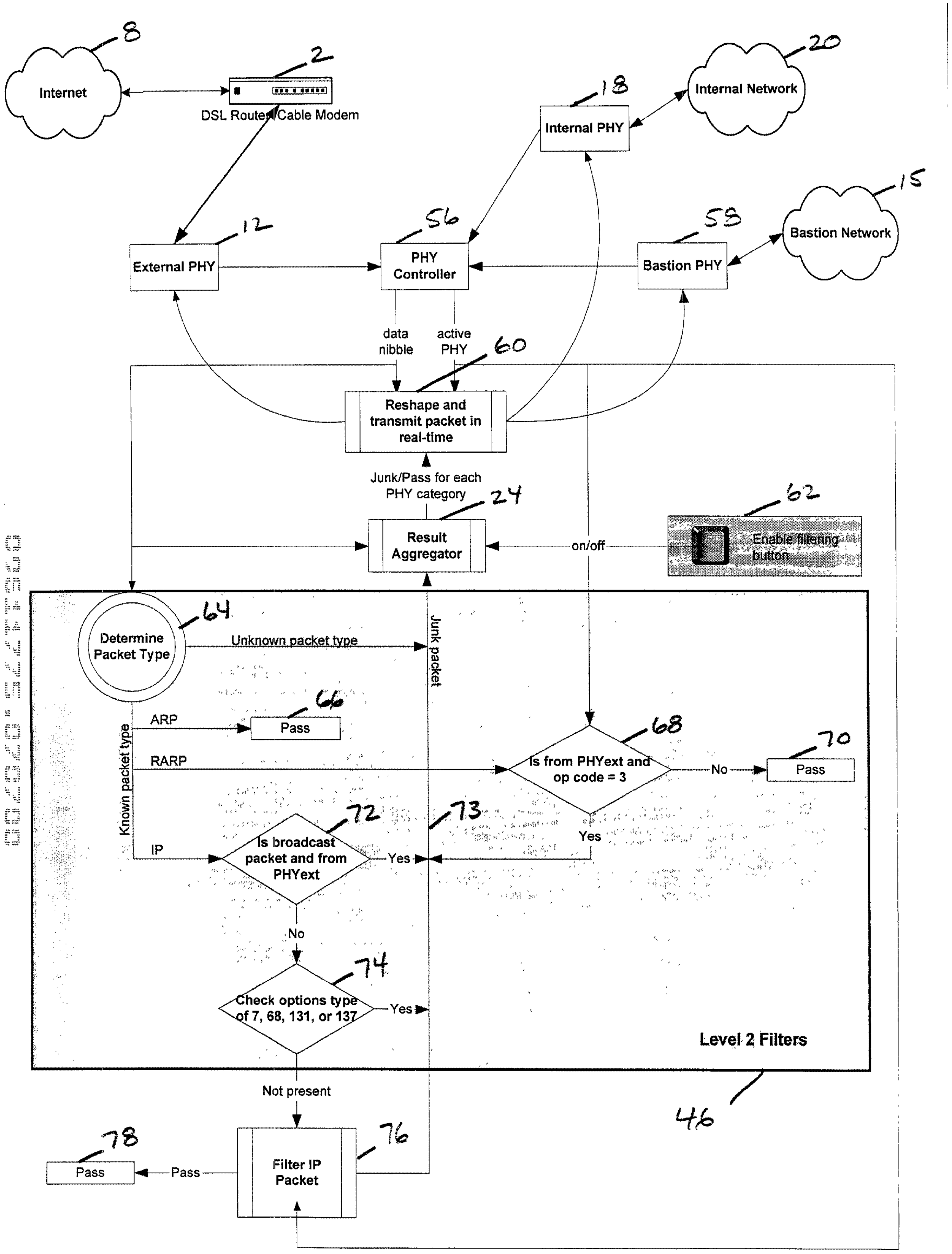

12
External Network
15
Bastion Network
20
Internal Network
16
Packet data
Repeater Core
pass/fail for each network
44
Determine packet characteristics (protocol, addrs, ports, flags)
24
Result Aggregator
46
Level 2 Filters
pass/fail
48
Level 3 Filters
pass/fail
(optional)
50
Level 4 Filters
pass/fail
52
Spoof Check
pass/fail
53
Alarm Controller
transmit alarm information over netwrok
54
Alert LED
55

FIG. 3



8
Internet
2
DSL Router/Cable Modem
18
Internal PHY
20
Internal Network
15
Bastion Network
12
External PHY
56
PHY Controller
58
Bastion PHY
data nibble
active PHY
60
Reshape and transmit packet in real-time
Junk/Pass for each PHY category
24
Result Aggregator
on/off
62
Enable filtering button
64
Determine Packet Type
Unknown packet type
Junk packet
Known packet type
ARP
66
Pass
RARP
68
Is from PHYext and op code = 3
No
70
Pass
Yes
73
IP
72
Is broadcast packet and from PHYext
Yes
No
74
Check options type of 7, 68, 131, or 137
Yes
Level 2 Filters
46
Not present
78
Pass
Pass
76
Filter IP Packet

FIG. 4

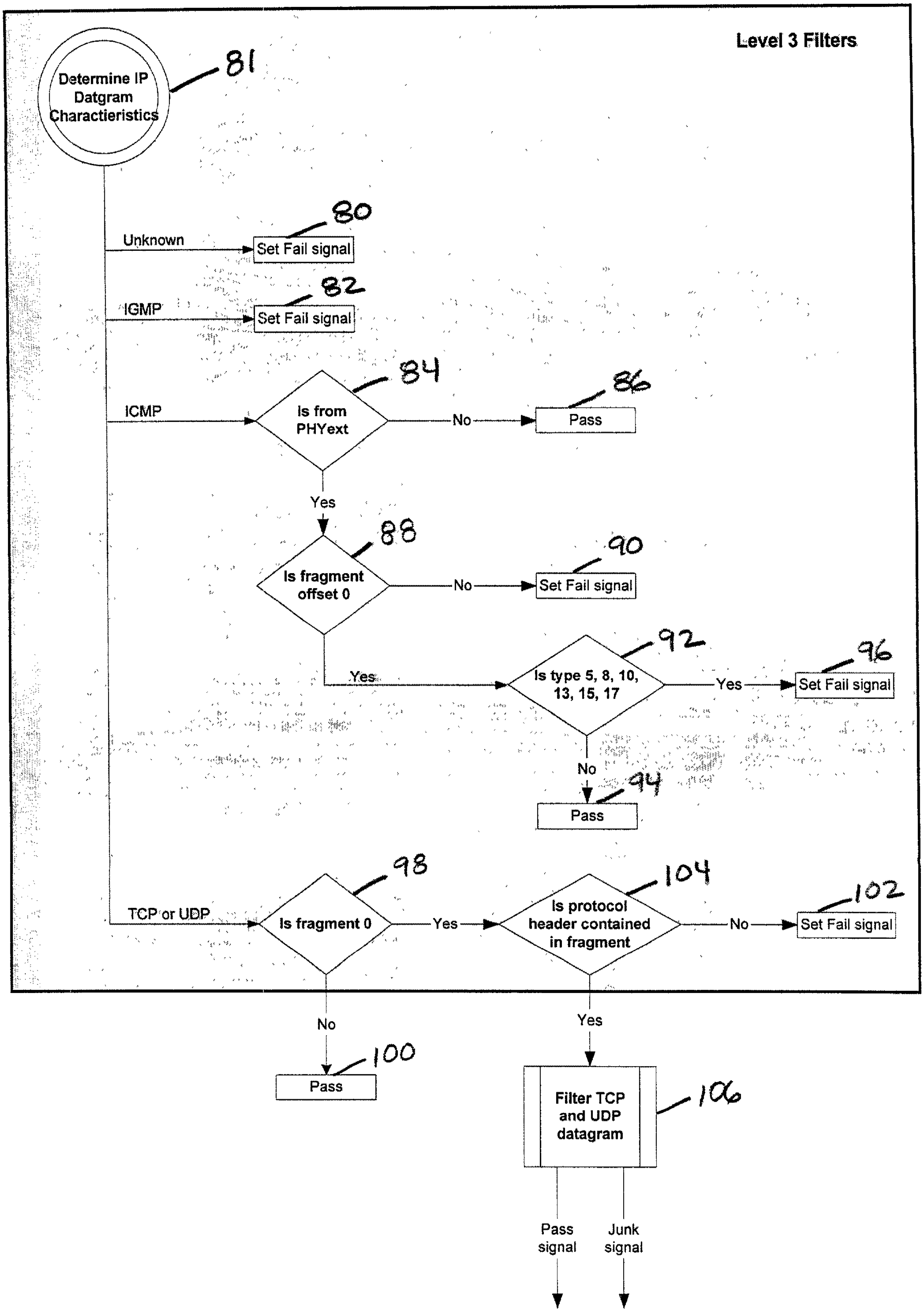

Level 3 Filters
Determine IP Datgram Charactieristics
81
Unknown
Set Fail signal
80
IGMP
Set Fail signal
82
ICMP
Is from PHYext
84
No
Pass
86
Yes
Is fragment offset 0
88
No
Set Fail signal
90
Yes
Is type 5, 8, 10, 13, 15, 17
92
Yes
Set Fail signal
96
No
Pass
94
TCP or UDP
Is fragment 0
98
Yes
Is protocol header contained in fragment
104
No
Set Fail signal
102
No
Pass
100
Yes
Filter TCP and UDP datagram
106
Pass signal
Junk signal

FIG. 5

TCP and UDP packets are evaluated for pass or fail in parallel (other protocols also handled simultaneously)

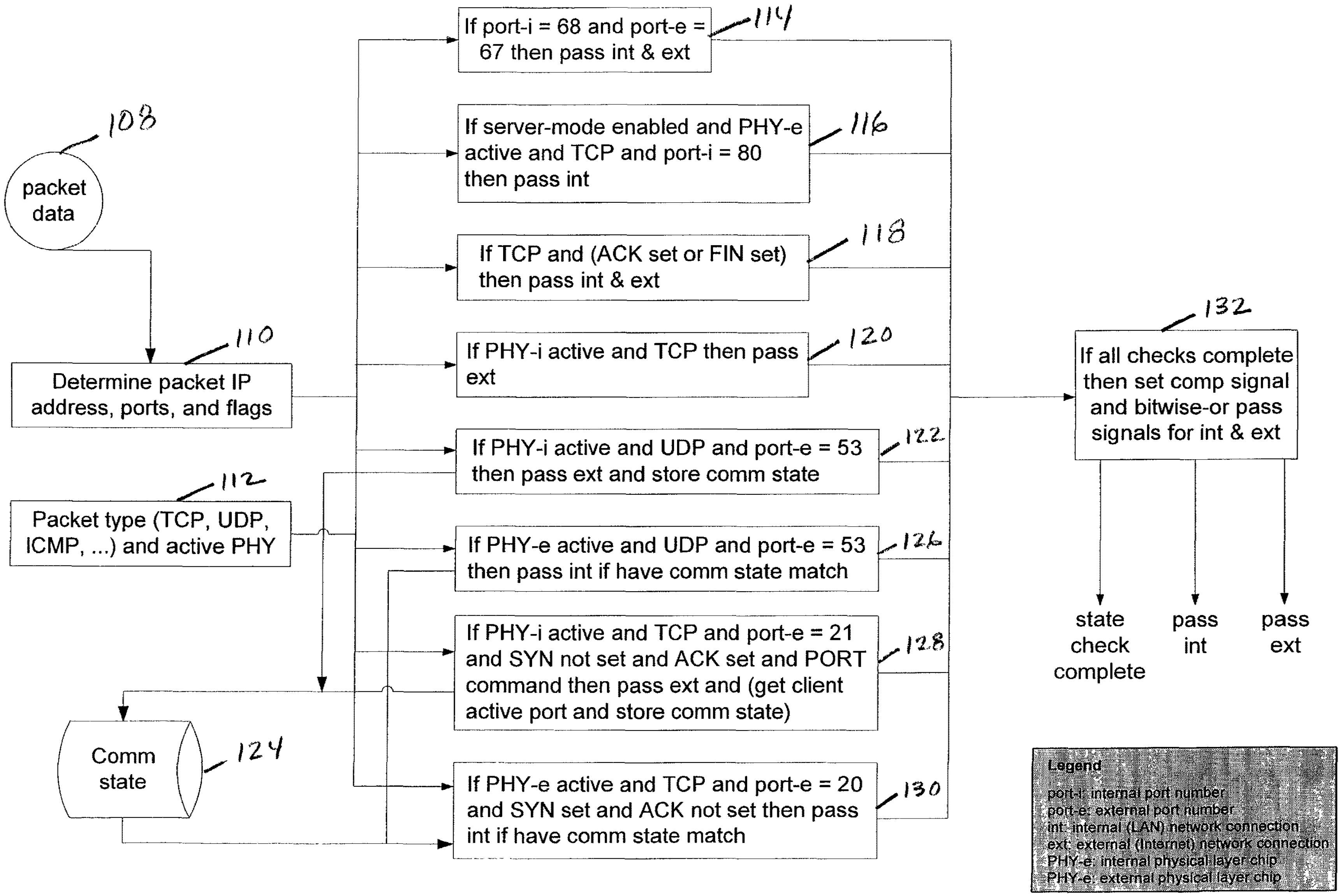

108
packet data
110
Determine packet IP address, ports, and flags
112
Packet type (TCP, UDP, ICMP, ...) and active PHY
124
Comm state
114
If port-i = 68 and port-e = 67 then pass int & ext
116
If server-mode enabled and PHY-e active and TCP and port-i = 80 then pass int
118
If TCP and (ACK set or FIN set) then pass int & ext
120
If PHY-i active and TCP then pass ext
122
If PHY-i active and UDP and port-e = 53 then pass ext and store comm state
126
If PHY-e active and UDP and port-e = 53 then pass int if have comm state match
128
If PHY-i active and TCP and port-e = 21 and SYN not set and ACK set and PORT command then pass ext and (get client active port and store comm state)
130
If PHY-e active and TCP and port-e = 20 and SYN set and ACK not set then pass int if have comm state match
132
If all checks complete then set comp signal and bitwise-or pass signals for int & ext
state check complete
pass int
pass ext
Legend
port-i: internal port number
port-e: external port number
int: internal (LAN) network connection
ext: external (Internet) network connection
PHY-e: internal physical layer chip
PHY-e: external physical layer chip

FIG. 6

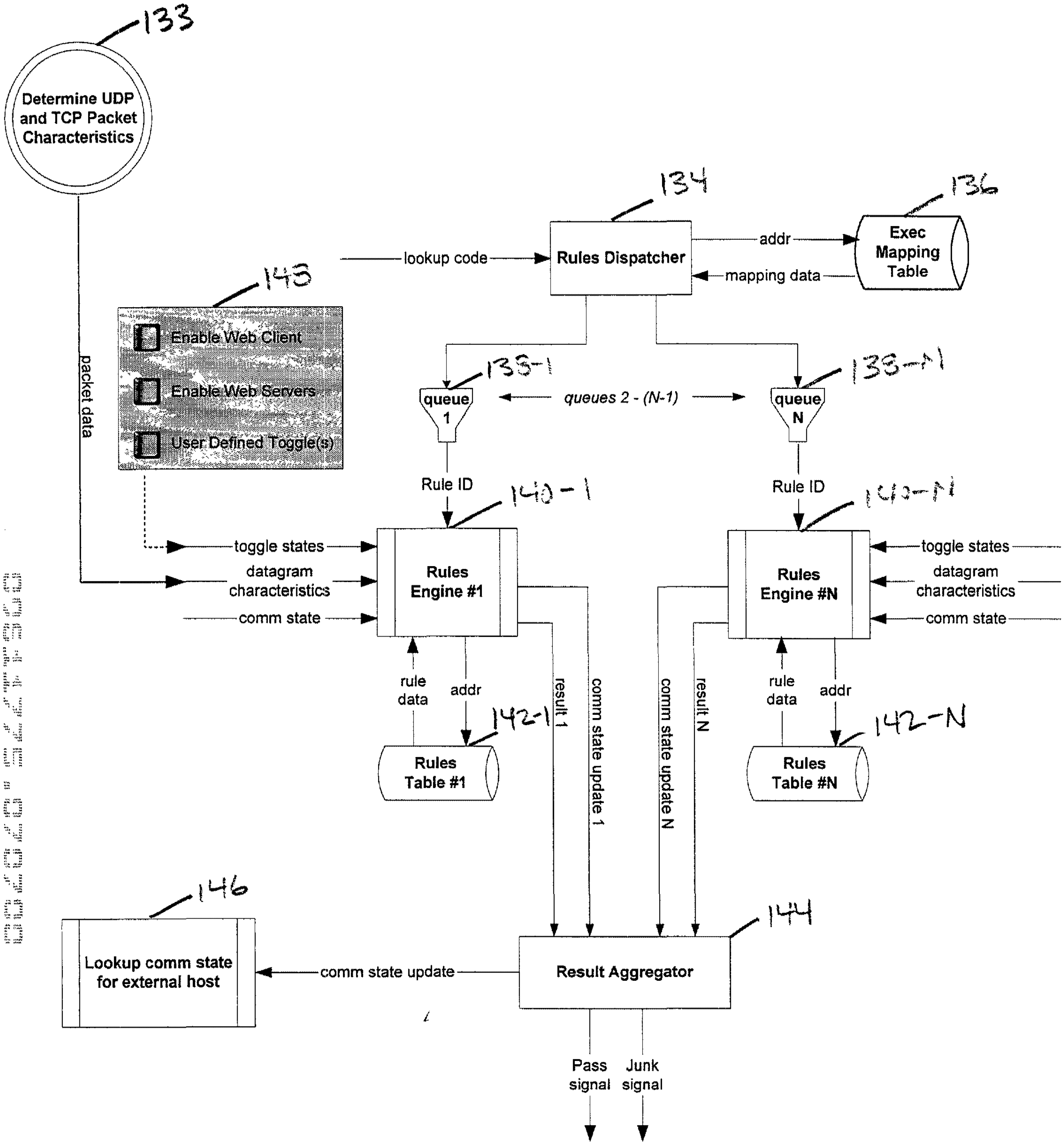

133
Determine UDP and TCP Packet Characteristics
packet data
134
Rules Dispatcher
lookup code
addr
mapping data
136
Exec Mapping Table
145
Enable Web Client
Enable Web Servers
User Defined Toggle(s)
138-1
138-N
queue 1
queues 2 - (N-1)
queue N
Rule ID
Rule ID
140-1
140-N
toggle states
datagram characteristics
comm state
Rules Engine #1
Rules Engine #N
toggle states
datagram characteristics
comm state
rule data
addr
rule data
addr
142-1
142-N
Rules Table #1
Rules Table #N
result 1
comm state update 1
comm state update N
result N
146
Lookup comm state for external host
comm state update
144
Result Aggregator
Pass signal
Junk signal

FIG. 7

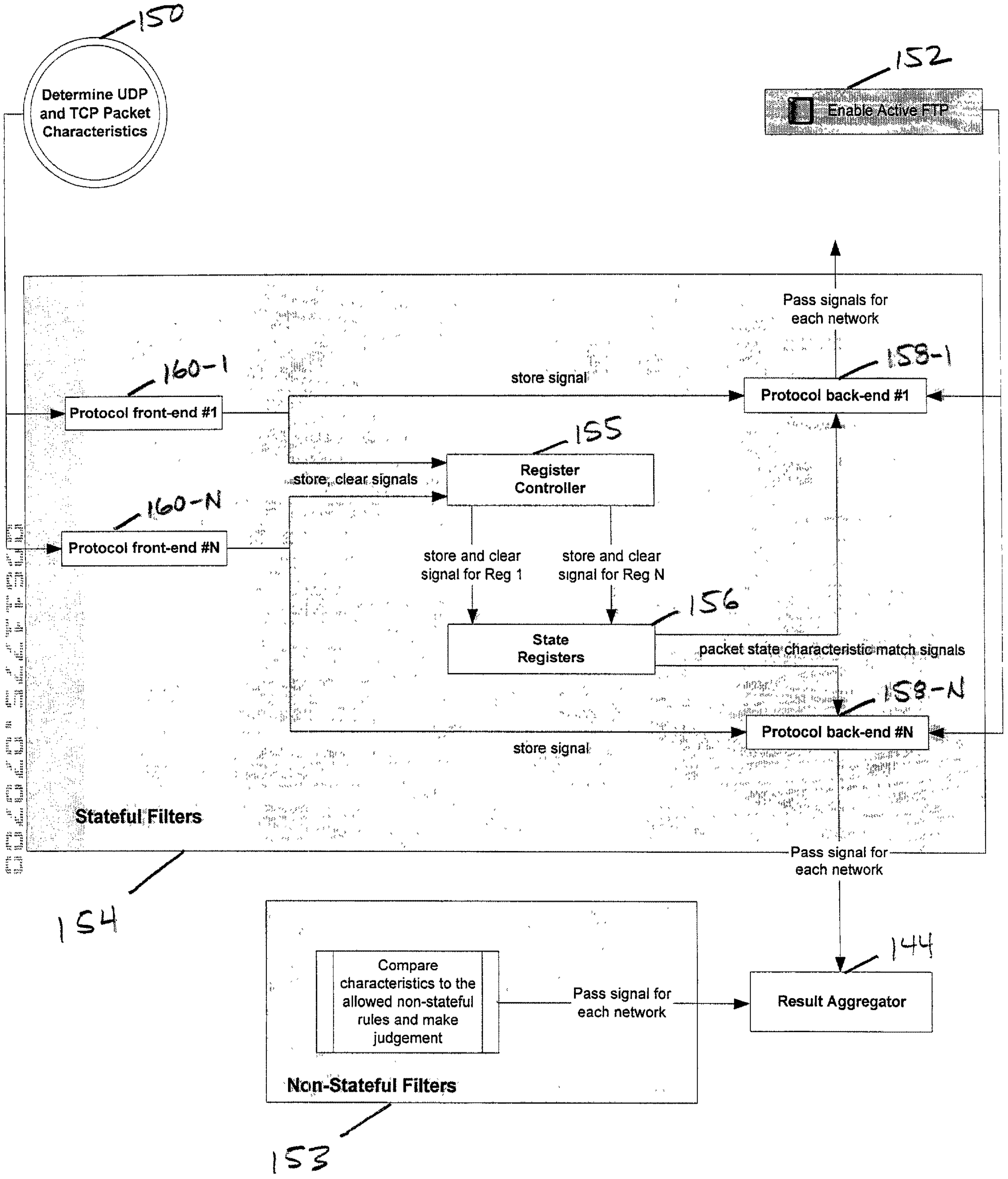

150
Determine UDP and TCP Packet Characteristics
152
Enable Active FTP
Pass signals for each network
158-1
store signal
160-1
Protocol front-end #1
Protocol back-end #1
155
Register Controller
store, clear signals
160-N
Protocol front-end #N
store and clear signal for Reg 1
store and clear signal for Reg N
156
State Registers
packet state characteristic match signals
158-N
Protocol back-end #N
store signal
Stateful Filters
Pass signal for each network
154
144
Compare characteristics to the allowed non-stateful rules and make judgement
Pass signal for each network
Result Aggregator
Non-Stateful Filters
153

FIG. 8

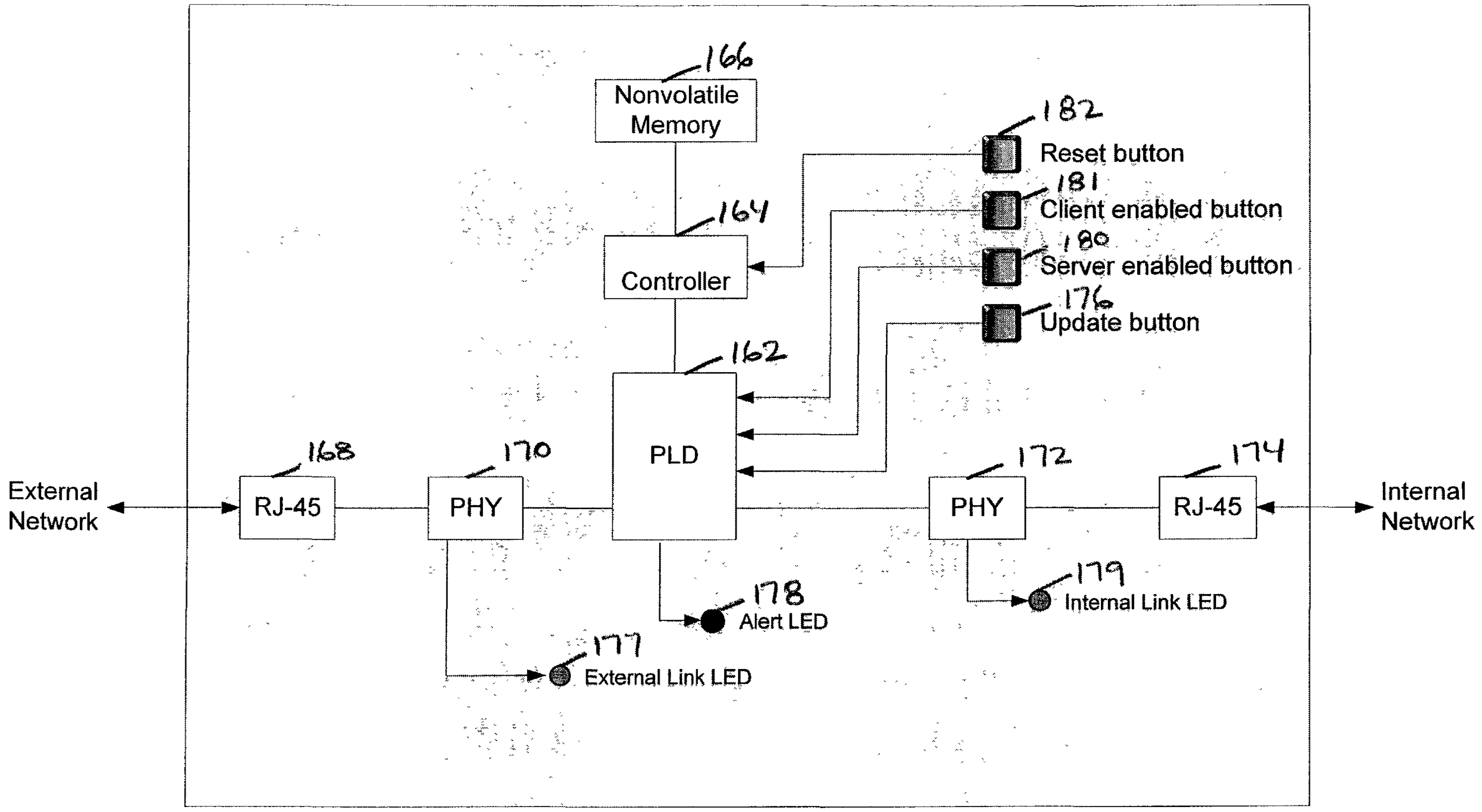

166
Nonvolatile Memory
164
Controller
162
PLD
182
Reset button
181
Client enabled button
180
Server enabled button
176
Update button
External Network
168
RJ-45
170
PHY
172
PHY
174
RJ-45
Internal Network
177
External Link LED
178
Alert LED
179
Internal Link LED

FIG. 9

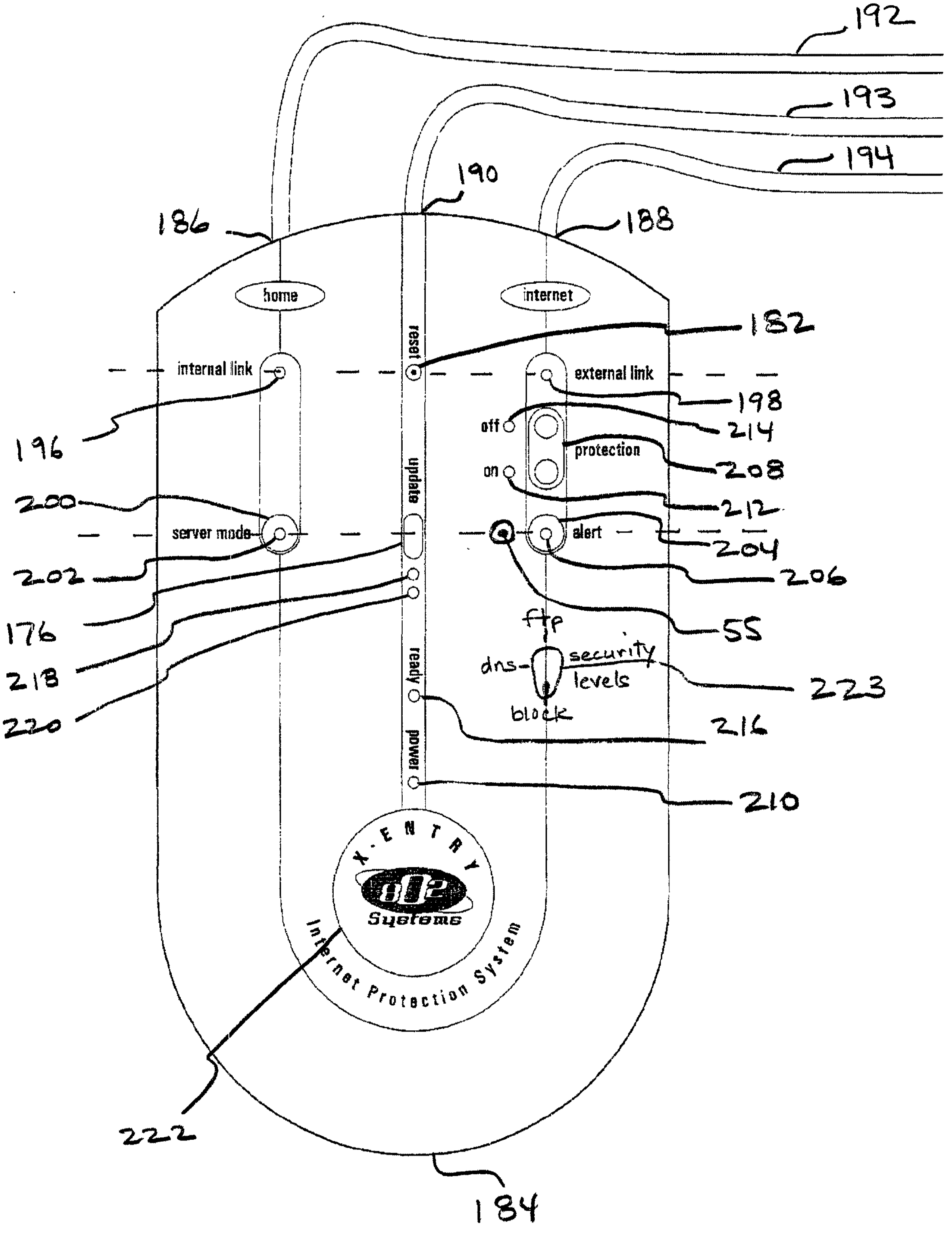

192
193
194
190
186
188
home
internet
reset
182
internal link
external link
198
196
off
214
protection
208
on
update
200
212
server mode
alert
204
202
206
176
55
ftp
218
dns
security levels
223
ready
220
block
216
power
210
X-ENTRY
812
Systems
Internet Protection System
222
184

FIG. 10

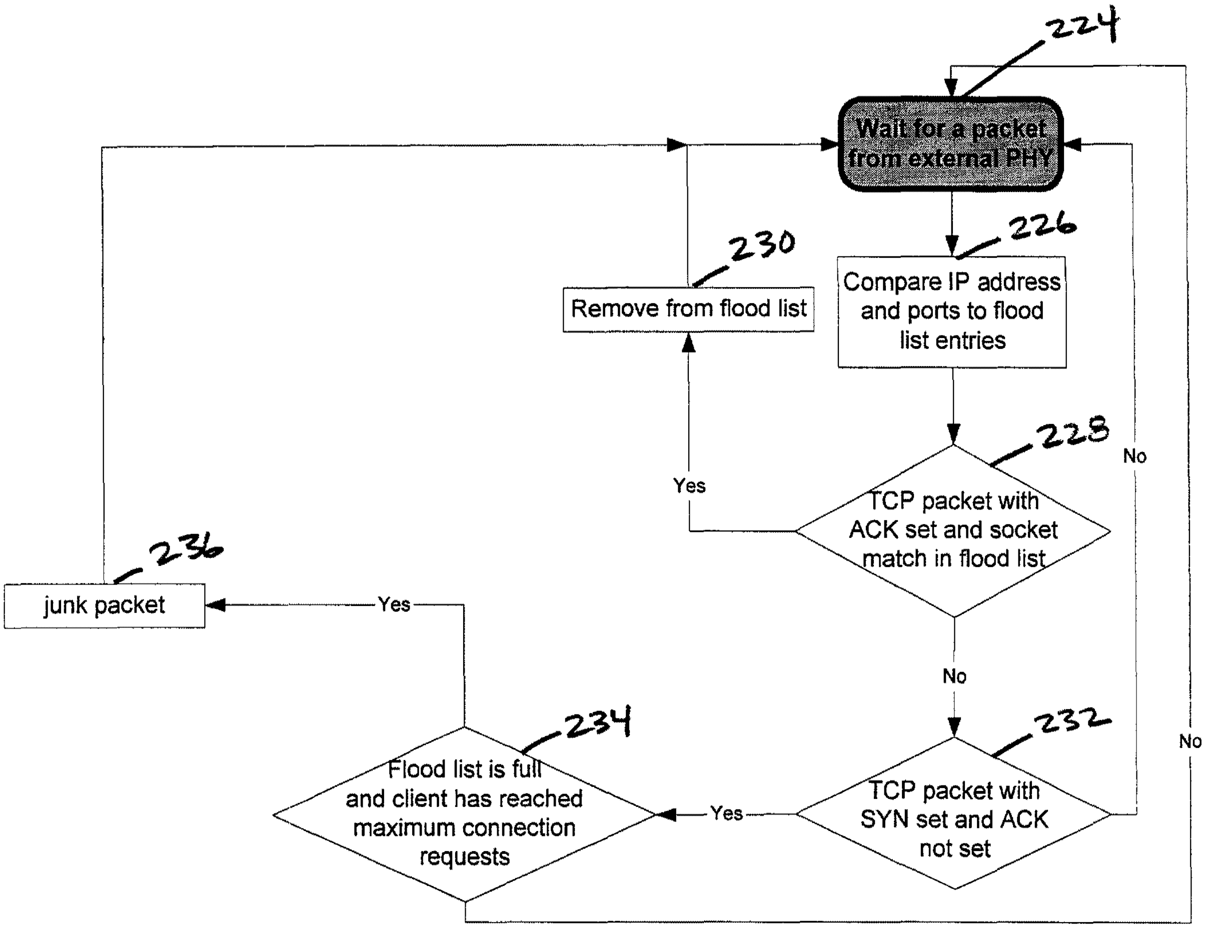

224
Wait for a packet from external PHY
226
Compare IP address and ports to flood list entries
230
Remove from flood list
228
TCP packet with ACK set and socket match in flood list
Yes
No
No
236
junk packet
Yes
234
Flood list is full and client has reached maximum connection requests
232
TCP packet with SYN set and ACK not set
Yes
No

FIG. 11

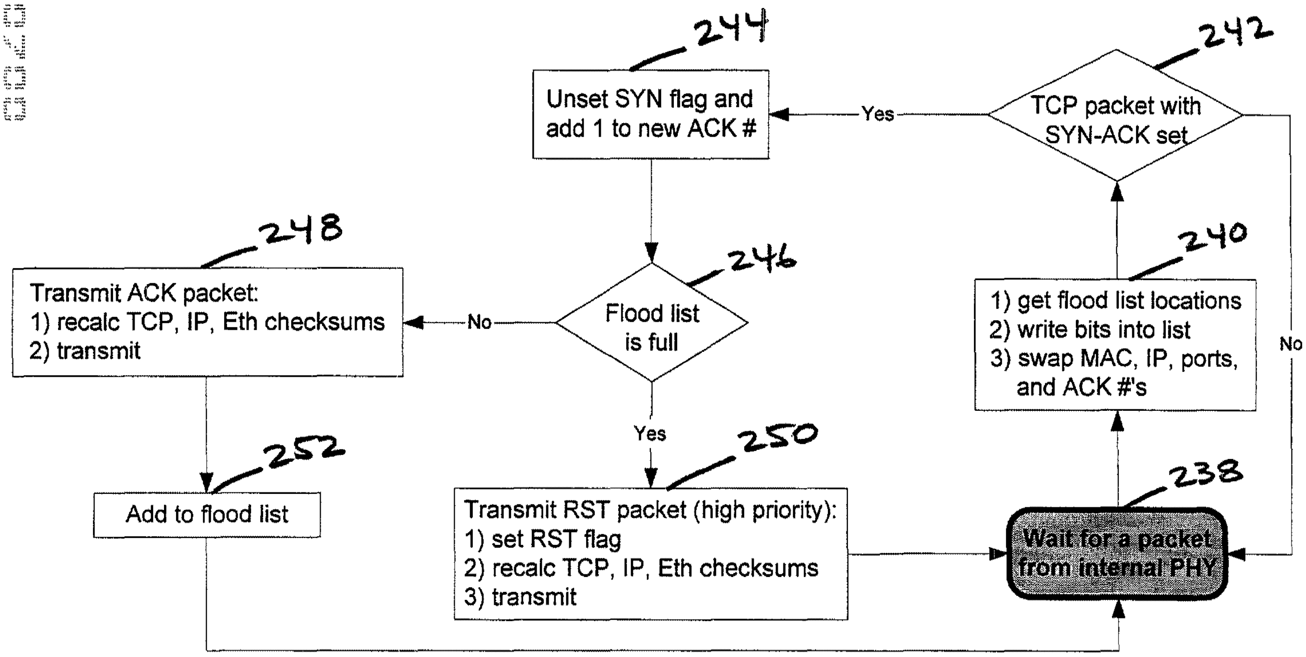

244
Unset SYN flag and add 1 to new ACK #
Yes
242
TCP packet with SYN-ACK set
No
248
Transmit ACK packet:
1) recalc TCP, IP, Eth checksums
2) transmit
No
246
Flood list is full
240
1) get flood list locations
2) write bits into list
3) swap MAC, IP, ports, and ACK #'s
Yes
252
Add to flood list
250
Transmit RST packet (high priority):
1) set RST flag
2) recalc TCP, IP, Eth checksums
3) transmit
238
Wait for a packet from internal PHY

FIG. 12

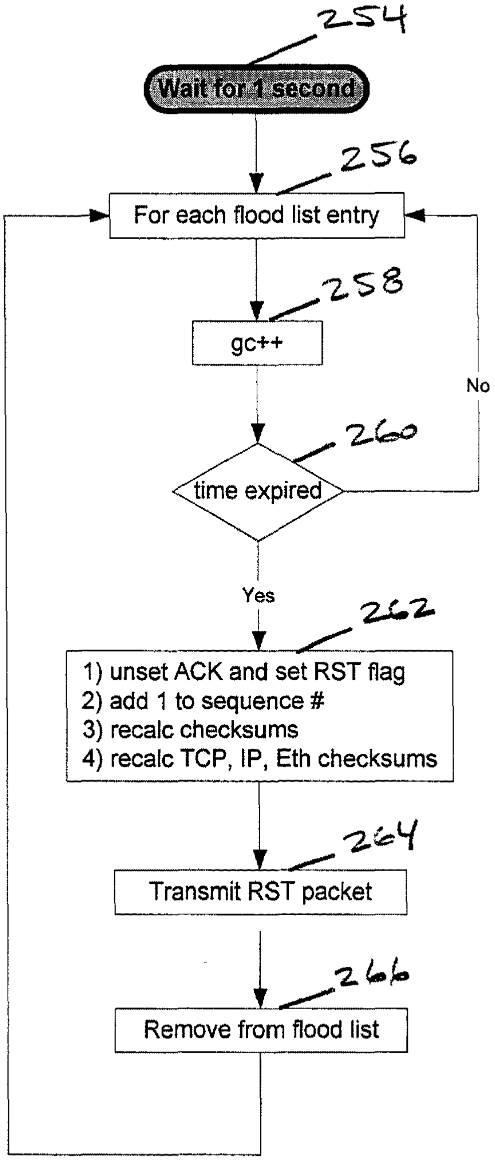

254
Wait for 1 second
256
For each flood list entry
258
gc++
260
time expired
No
Yes
262
1) unset ACK and set RST flag
2) add 1 to sequence #
3) recalc checksums
4) recalc TCP, IP, Eth checksums
264
Transmit RST packet
266
Remove from flood list

FIG. 13

Attorney Docket No.: 802-001

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

INVENTOR AND SPECIFICATION IDENTIFICATION

My residence, post office address and citizenship are as stated below next to my name, I believe that I am the original, first and sole inventor (*if only one name is listed below*) or an original, first and joint inventor (*if plural names are listed below*) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS
TITLE OF INVENTION

the specification of which:

X is attached hereto.

___ was filed on __________ as Application Serial No. ____________
and was amended on ____________________ (*if applicable*).

___ was described and claimed in PCT International Application No.____________filed on
______________and amended under PCT Article 19 on __________ (*if any*).

REVIEW OF PAPERS AND ACKNOWLEDGMENT OF DUTY OF CANDOR

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I do not know and do not believe that the invention claimed in the above-identified specification was ever known or used in the United States of America before my or our invention thereof, or patented or described in any printed publication in any country before my or our invention thereof or more than one year prior to this application, and that the same was not in public use or on sale in the United States of America more than one year prior to this application.

I acknowledge the duty to disclose to the Patent and Trademark Office information which I know is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56.

FOREIGN APPLICATIONS AND PRIORITY CLAIM

The invention claimed in the above-described specification has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application. I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed.

COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 37 USC 119
			__Yes __No
			__Yes __No
			__Yes __No
			__Yes __No

DOMESTIC PRIORITY CLAIM

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States patent application(s) listed below and, insofar as this application discloses or claims subject matter in addition to that disclosed in the below listed priority applications, I acknowledge the duty to disclose to the Patent and Trademark Office all information known by me to be material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date(s) of the below-listed prior application(s) and the national or PCT international filing date of this application.

______________________ ______________ ______________________________
(APPLICATION SERIAL NO.) (FILING DATE) (STATUS PATENTED, PENDING, ABANDONED)

______________________ ______________ ______________________________
(APPLICATION SERIAL NO.) (FILING DATE) (STATUS· PATENTED, PENDING, ABANDONED)

POWER OF ATTORNEY

I hereby appoint Alan R. Loudermilk (Reg. No. 32,788), who is registered to practice before the Patent and Trademark Office, as my attorney with full power of substitution and revocation, to prosecute this application, to make alterations or amendments therein, to receive the patent and transact all business in the Patent and Trademark Office connected therewith.

All **CORRESPONDENCE** should be addressed to:

Loudermilk & Associates
10950 N. Blaney Avenue Suite B
Cupertino, CA 95014

All **TELEPHONE INQUIRIES** may be directed to Alan R. Loudermilk at (408) 342-1866.

I hereby declare I have read this Declaration, and that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

HAND PRINT DATE BEFORE SIGNING

Full name of sole or
first joint inventor ANDREW K. KRUMEL Citizenship USA

Inventor's signature ______ Date 7/7/00

Residence 3635 Pleasant Knoll Drive, San Jose, CA 95148

Post Office Address 3635 Pleasant Knoll Drive, San Jose, CA 95148

Full name of second
joint inventor ______ Citizenship ______

Inventor's signature ______ Date ______

Residence ______

Post Office Address ______

Full name of third
joint inventor ______ Citizenship ______

Inventor's signature ______ Date ______

Residence ______

Post Office Address ______

Full name of fourth
joint inventor ______ Citizenship ______

Inventor's signature ______ Date ______

Residence ______

Post Office Address ______

Full name of fifth
joint inventor ______ Citizenship ______

Inventor's signature ______ Date ______

Residence ______

Post Office Address ______

_____ If this line is checked, the signature page is continued on the attached Addendum.

(Declaration and Power of Attorney - Page 3 of 3)

09/611775 PTO
07/07/00

JC1
JC835

ISSUE CLASSIFICATION	
Class	Subclass

PATENT NUMBER

U.S. **UTILITY** Patent Application

O.I.P.E.	PATENT DATE
SCANNED LR Q.A. AA	

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/611775		713	201	2131	

APPLICANTS Andrew Krumel

Sinitoski

TITLE Real time firewall/data protection systems and methods

PTO-2040
12/99

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
INTERNATIONAL CLASSIFICATION								
				□ Continued on Issue Slip Inside File Jacket				

□ **TERMINAL DISCLAIMER**

□ The term of this patent subsequent to ____________ (date) has been disclaimed.

□ The term of this patent shall not extend beyond the expiration date of U.S Patent. No. ____________

□ The terminal ____ months of this patent have been disclaimed.

DRAWINGS

Sheets Drwg.	Figs. Drwg.	Print Fig.

(Assistant Examiner) (Date)

(Primary Examiner) (Date)

(Legal Instruments Examiner) (Date)

CLAIMS ALLOWED

Total Claims	Print Claim for O.G.

NOTICE OF ALLOWANCE MAILED

ISSUE FEE

Amount Due	Date Paid

ISSUE BATCH NUMBER

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 6/99)

FILED WITH: □ DISK (CRF) □ FICHE □ CD-ROM
(Attached in pocket on right inside flap)

BEST AVAILABLE COPY

(FACE)

SEARCHED

Class	Sub.	Date	Exmr.
713 709	201 (text only) 229, 249, 225 (text only)	2/13/04	
370	356, 389, 392, 395.21, 395.32, 401 (text only)	2/17/04	

SEARCH NOTES
(INCLUDING SEARCH STRATEGY)

	Date	Exmr.
See attached IEEE, ACM, Gogle & NPL search notes.	2/13/04	
See attached EAST search notes	2/18/04	
Consulted with Andrew Caldwell	2/18/04	

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.

BEST AVAILABLE COPY

(RIGHT OUTSIDE)

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION	VT	69607	7/17/00
O.I.P.E. CLASSIFIER		39	7/21
FORMALITY REVIEW	Sk	835	8/23/00
RESPONSE FORMALITY REVIEW			

INDEX OF CLAIMS

✓	.... Rejected	N	.... Non-elected
=	.... Allowed	I	.... Interference
—	(Through numeral)... Canceled	A	.... Appeal
÷	.... Restricted	O	.... Objected

Claim		Date									
Final	Original	2/18/04									
	1	✓									
	2	✓									
	3	✓									
	4	✓									
	5	✓									
	6	✓									
	7	✓									
	8	✓									
	9	✓									
	10	✓									
	11	✓									
	12	✓									
	13	✓									
	14	✓									
	15	✓									
	16	✓									
	17	✓									
	18	✓									
	19	✓									
	20	✓									
	21	✓									
	22	✓									
	23	✓									
	24	✓									
	25	✓									
	26	✓									
	27	✓									
	28	✓									
	29	✓									
	30	✓									
	31	✓									
	32	✓									
	33	✓									
	34	✓									
	35	✓									
	36	✓									
	37	✓									
	38	✓									
	39	0									
	40	✓									
	41	✓									
	42	✓									
	43	✓									
	44	✓									
	45	✓									
	46	✓									
	47	✓									
	48	✓									
	49	✓									
	50	✓									

Claim		Date									
Final	Original	2/18/04									
	51	✓									
	52	✓									
	53	✓									
	54	✓									
	55	✓									
	56	✓									
	57	✓									
	58	✓									
	59	✓									
	60	✓									
	61	✓									
	62	✓									
	63	✓									
	64	✓									
	65	✓									
	66	✓									
	67										
	68										
	69										
	70										
	71										
	72										
	73										
	74										
	75										
	76										
	77										
	78										
	79										
	80										
	81										
	82										
	83										
	84										
	85										
	86										
	87										
	88										
	89										
	90										
	91										
	92										
	93										
	94										
	95										
	96										
	97										
	98										
	99										
	100										

Claim		Date									
Final	Original										
	101										
	102										
	103										
	104										
	105										
	106										
	107										
	108										
	109										
	110										
	111										
	112										
	113										
	114										
	115										
	116										
	117										
	118										
	119										
	120										
	121										
	122										
	123										
	124										
	125										
	126										
	127										
	128										
	129										
	130										
	131										
	132										
	133										
	134										
	135										
	136										
	137										
	138										
	139										
	140										
	141										
	142										
	143										
	144										
	145										
	146										
	147										
	148										
	149										
	150										

If more than 150 claims or 10 actions
staple additional sheet here

(LEFT INSIDE)

07-10-00

A

FORM PTO-1082

Attorney Docket No.: 802-001

BOX PATENT APPLICATION
THE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of
Inventor(s): Andrew K. Krumel
For (Title): REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

Enclosed are:

[X] 13 sheets of informal drawings.

[X] An assignment of the invention to 802 Systems, Inc.

[ ] A certified copy of a ______________________ application.

[X] A declaration and power of attorney.

[X] A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.

[ ] Applicant(s) claim convention priority under 35 U.S.C. 119 based on ______________ application
Serial No. ______________ filed ______________.

[ ] ______________________________

	(Col. 1)	(Col. 2)	SMALL ENTITY			OTHER THAN A SMALL ENTITY	
FOR:	NO. FILED	NO. EXTRA	RATE	FEE	OR	RATE	FEE
BASIC FEE				$380.00	OR		760.00
TOTAL CLAIMS	66 -20 =	46	x 9=	$414.00	OR	x 18=	$
INDEP. CLAIMS	2 - 3 =	0	x 39=	$	OR	x 78=	$
MULTIPLE DEPENDENT CLAIM(S) PRESENTED =			+ 130=	$	OR	+ 260=	$
			TOTAL	$ 794.00	OR	TOTAL	$

* If the difference in Col. 1 is less than zero, enter "0" in Col. 2.

[X] Please charge Deposit Account No. 50-0251 or backup account 12-2175 in the amount of $794.00. An additional copy of this sheet is enclosed.

[ ] A check in the amount of $__________ to cover the filing fee is enclosed.

[X] The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175. An additional copy of this sheet is enclosed.
- [X] Any additional filing fees required under 37 CFR 1.16.
- [X] Any patent application processing fees under 37 CFR 1.17.

[X] The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175. An additional copy of this sheet is enclosed.
- [X] Any patent application processing fees under 37 CFR 1.17.
- [ ] The issue fee set in 37 CFR 1.18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b).
- [ ] Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Dated July 7, 2000 A Loudermilk Reg. No. 32,788
Attorney of Record: Alan R. Loudermilk

CERTIFICATE OF "EXPRESS MAIL" UNDER 37 CFR 1.10

"EXPRESS MAIL" Mailing Label Number **EL401103166US** Date of Deposit July , 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Maria Margaretich
(Typed or Printed Name of Person Mailing Paper or Fee)

(Signature of Person Mailing Paper or Fee)

LOUDERMILK & ASSOCIATES • 10950 N. Blaney Ave. • Suite B • Cupertino, CA 95014 • (408) 342-1866

APPLICATION TRANSMITTAL

07-10-00

A

FORM PTO-1082

Attorney Docket No.: 802-001

BOX PATENT APPLICATION
THE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of
Inventor(s): Andrew K. Krumel
For (Title): REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

Enclosed are:

[X] 13 sheets of informal drawings.

[X] An assignment of the invention to 802 Systems, Inc.

[ ] A certified copy of a ______ application.

[X] A declaration and power of attorney.

[X] A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.

[ ] Applicant(s) claim convention priority under 35 U.S.C. 119 based on ______ application
Serial No. ______ filed ______.

[ ] ______

	(Col. 1)	(Col. 2)	SMALL ENTITY			OTHER THAN A SMALL ENTITY	
FOR:	NO. FILED	NO. EXTRA	RATE	FEE	OR	RATE	FEE
BASIC FEE				$380.00	OR		760.00
TOTAL CLAIMS	66 -20 =	46	x 9=	$414.00	OR	x 18=	$
INDEP. CLAIMS	2 - 3 =	0	x 39=	$	OR	x 78=	$
MULTIPLE DEPENDENT CLAIM(S) PRESENTED =			+ 130=	$	OR	+ 260=	$
			TOTAL	$ 794.00	OR	TOTAL	$

* If the difference in Col. 1 is less than zero, enter "0" in Col. 2.

[X] Please charge Deposit Account No. 50-0251 or backup account 12-2175 in the amount of $794.00. An additional copy of this sheet is enclosed.

[ ] A check in the amount of $______ to cover the filing fee is enclosed.

[X] The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175. An additional copy of this sheet is enclosed.

- [X] Any additional filing fees required under 37 CFR 1.16.
- [X] Any patent application processing fees under 37 CFR 1.17.

[X] The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to Deposit Account No. 50-0251 or backup account 12-2175. An additional copy of this sheet is enclosed.

- [X] Any patent application processing fees under 37 CFR 1.17.
- [ ] The issue fee set in 37 CFR 1.18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b).
- [ ] Any filing fees under 37 CFR 1.16 for presentation of extra claims.

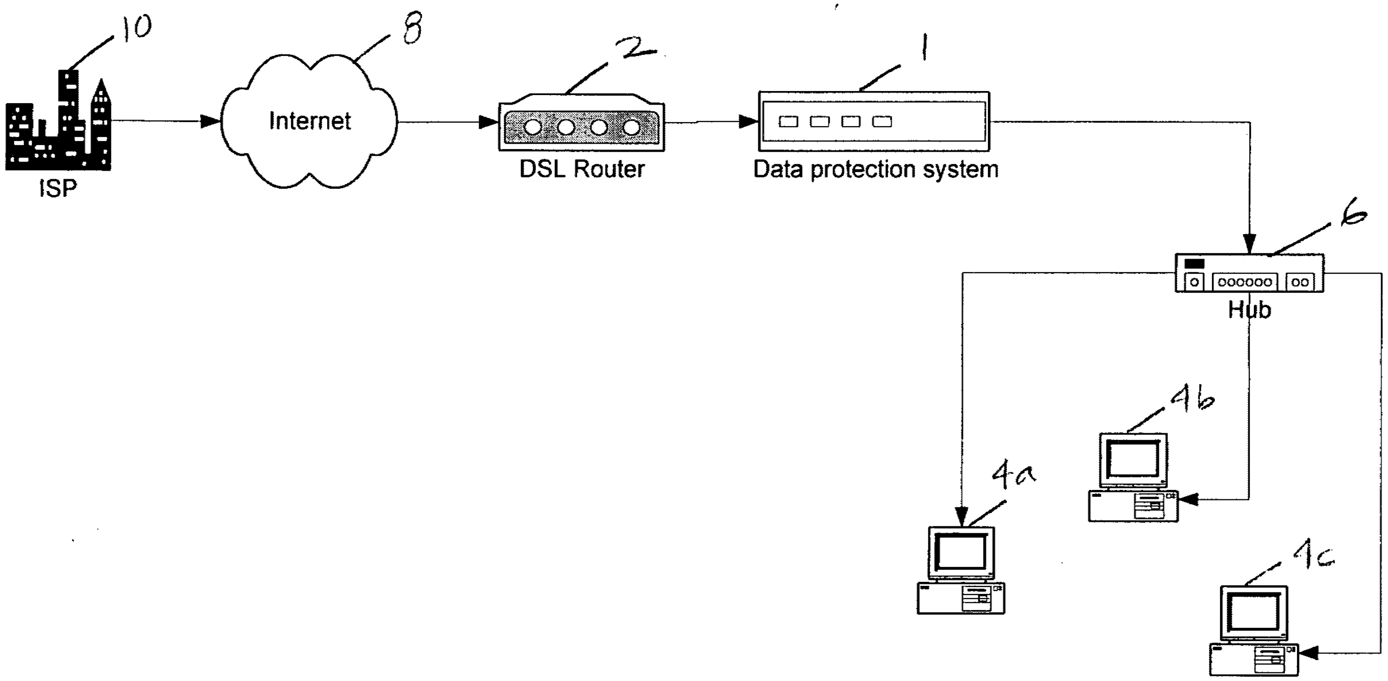
Dated July 7, 2000 ______ Reg. No. 32,788
Attorney of Record: Alan R. Loudermilk

CERTIFICATE OF "EXPRESS MAIL" UNDER 37 CFR 1.10

"EXPRESS MAIL" Mailing Label Number **EL401103166US** Date of Deposit July , 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Maria Margaretich
(Typed or Printed Name of Person Mailing Paper or Fee)

(Signature of Person Mailing Paper or Fee)



10
ISP
8
Internet
2
DSL Router
1
Data protection system
6
Hub
4a
4b
4c

FIG. 1A

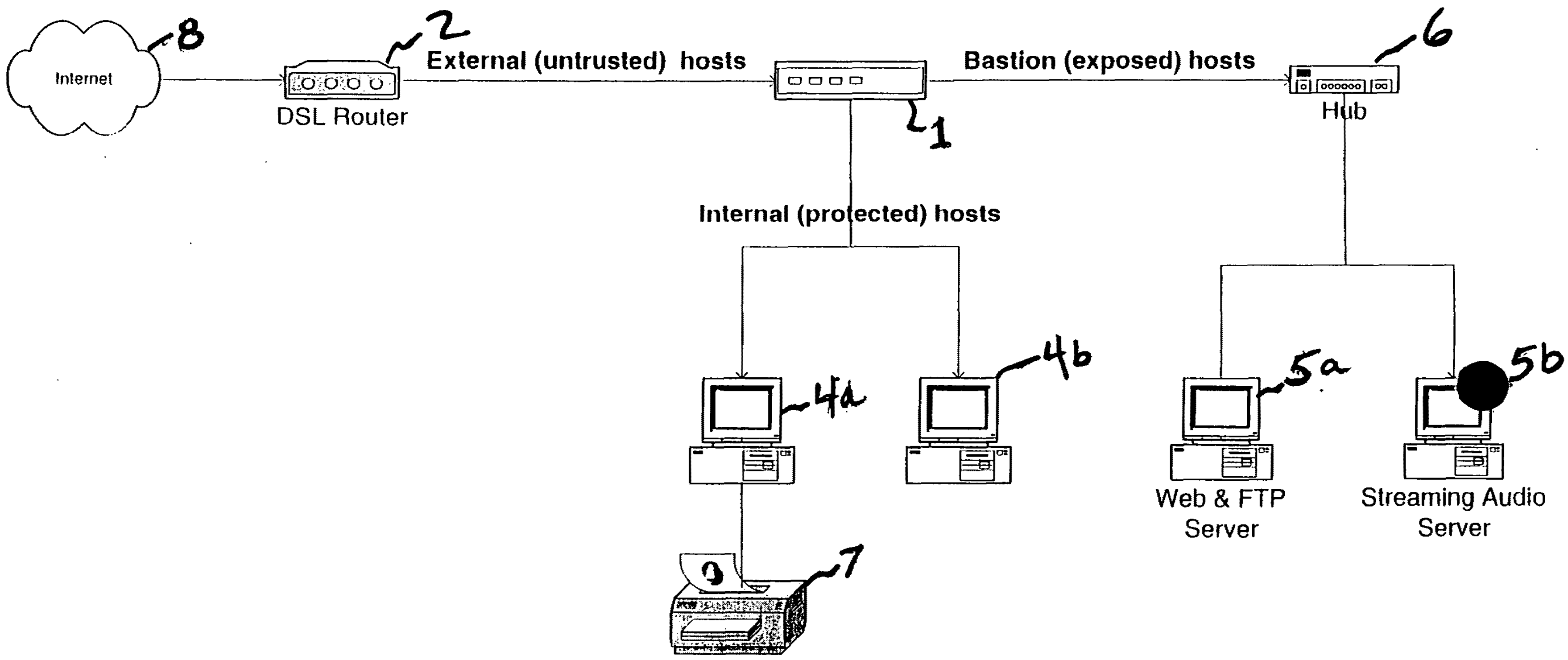

Internet
8
DSL Router
2
External (untrusted) hosts
1
Bastion (exposed) hosts
Hub
6
Internal (protected) hosts
4a
4b
7
5a
Web & FTP Server
5b
Streaming Audio Server

FIG. 1B

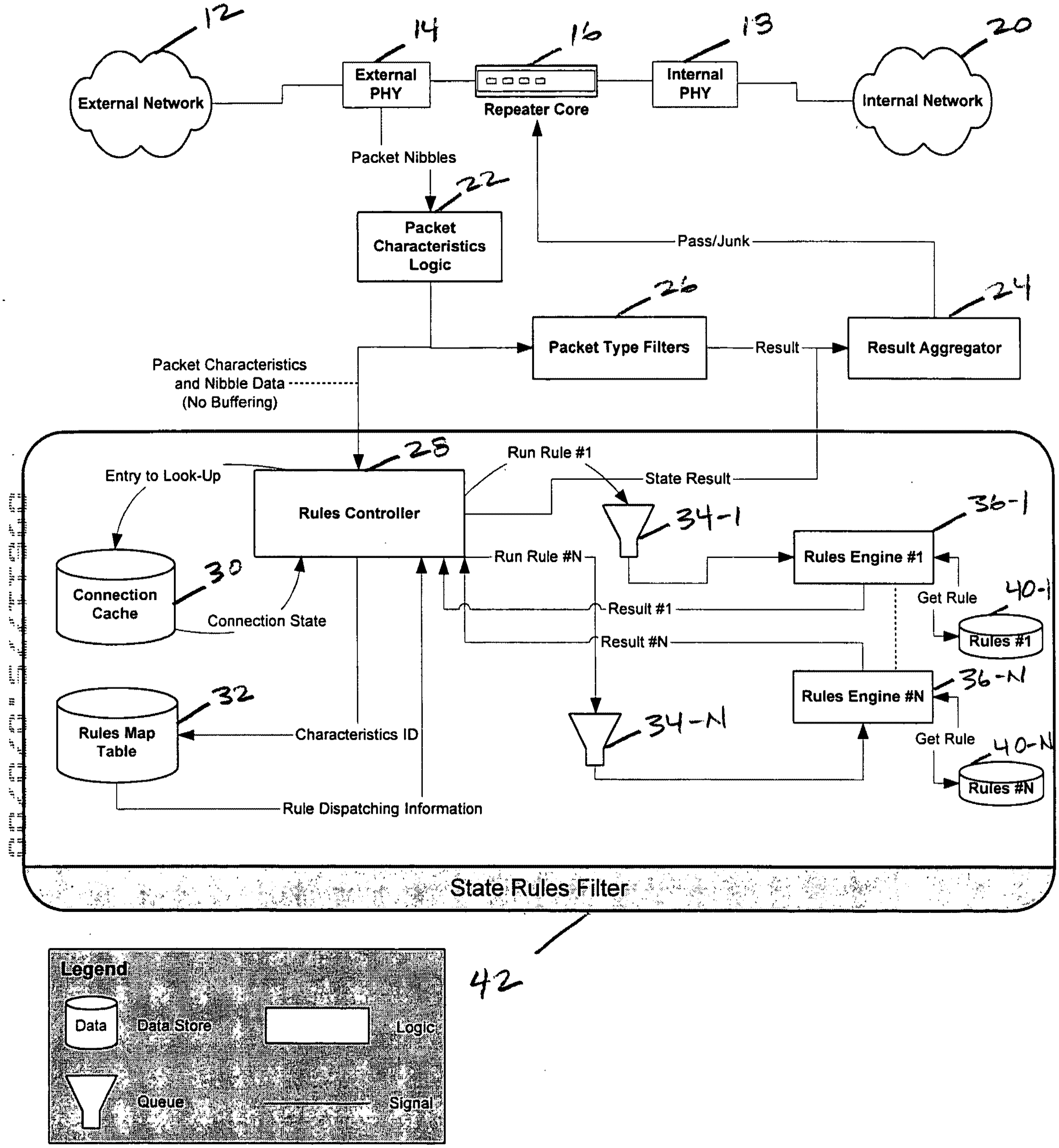

12
14
16
18
20
External Network
External PHY
Repeater Core
Internal PHY
Internal Network
Packet Nibbles
22
Packet Characteristics Logic
Pass/Junk
26
24
Packet Type Filters
Result
Result Aggregator
Packet Characteristics and Nibble Data (No Buffering)
28
Entry to Look-Up
Rules Controller
Run Rule #1
State Result
34-1
36-1
Rules Engine #1
Connection Cache
30
Connection State
Run Rule #N
Get Rule
40-1
Result #1
Rules #1
Result #N
36-N
Rules Engine #N
32
Rules Map Table
Characteristics ID
34-N
Get Rule
40-N
Rules #N
Rule Dispatching Information
State Rules Filter
42
Legend
Data
Data Store
Logic
Queue
Signal

FIG. 2

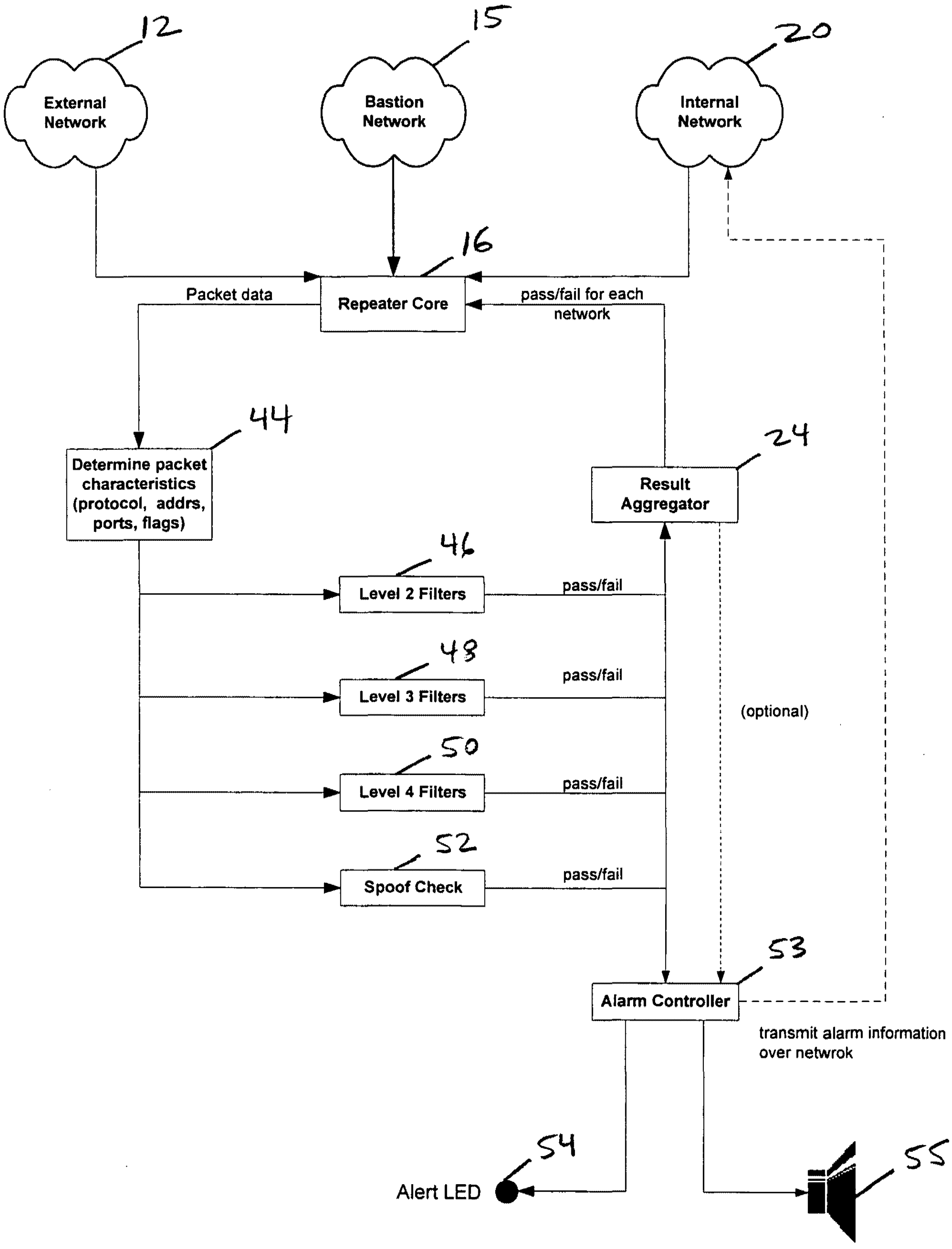

12
15
20
External Network
Bastion Network
Internal Network
16
Packet data
Repeater Core
pass/fail for each network
44
Determine packet characteristics (protocol, addrs, ports, flags)
24
Result Aggregator
46
Level 2 Filters
pass/fail
48
Level 3 Filters
pass/fail
(optional)
50
Level 4 Filters
pass/fail
52
Spoof Check
pass/fail
53
Alarm Controller
transmit alarm information over netwrok
54
Alert LED
55

FIG. 3

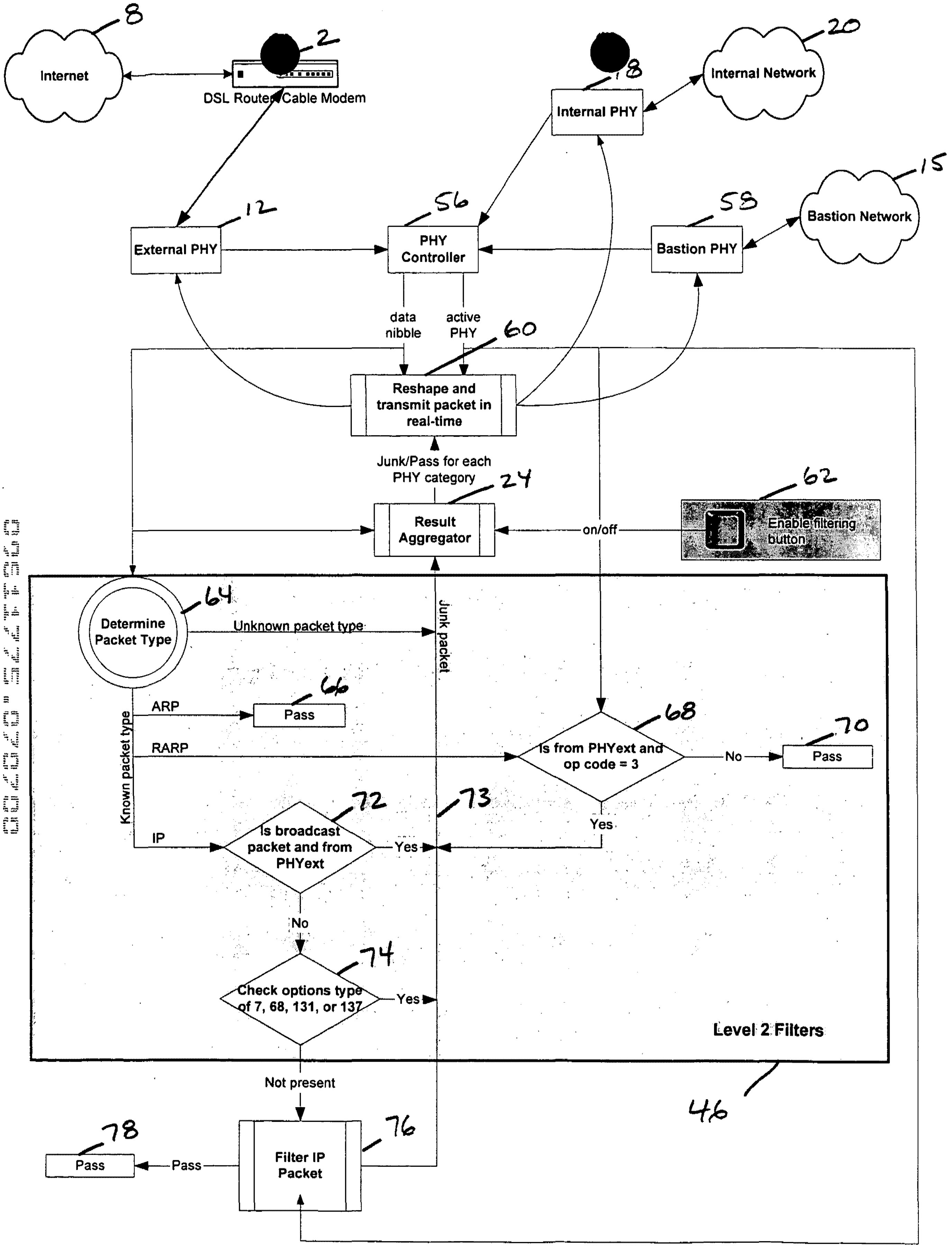

8
Internet
2
DSL Router/Cable Modem
18
Internal PHY
20
Internal Network
15
Bastion Network
12
External PHY
56
PHY Controller
58
Bastion PHY
data nibble
active PHY
60
Reshape and transmit packet in real-time
Junk/Pass for each PHY category
24
Result Aggregator
on/off
62
Enable filtering button
64
Determine Packet Type
Unknown packet type
Junk packet
Known packet type
ARP
66
Pass
RARP
68
Is from PHYext and op code = 3
No
70
Pass
Yes
IP
72
Is broadcast packet and from PHYext
Yes
73
No
74
Check options type of 7, 68, 131, or 137
Yes
Level 2 Filters
46
Not present
78
Pass
Pass
76
Filter IP Packet

FIG. 4

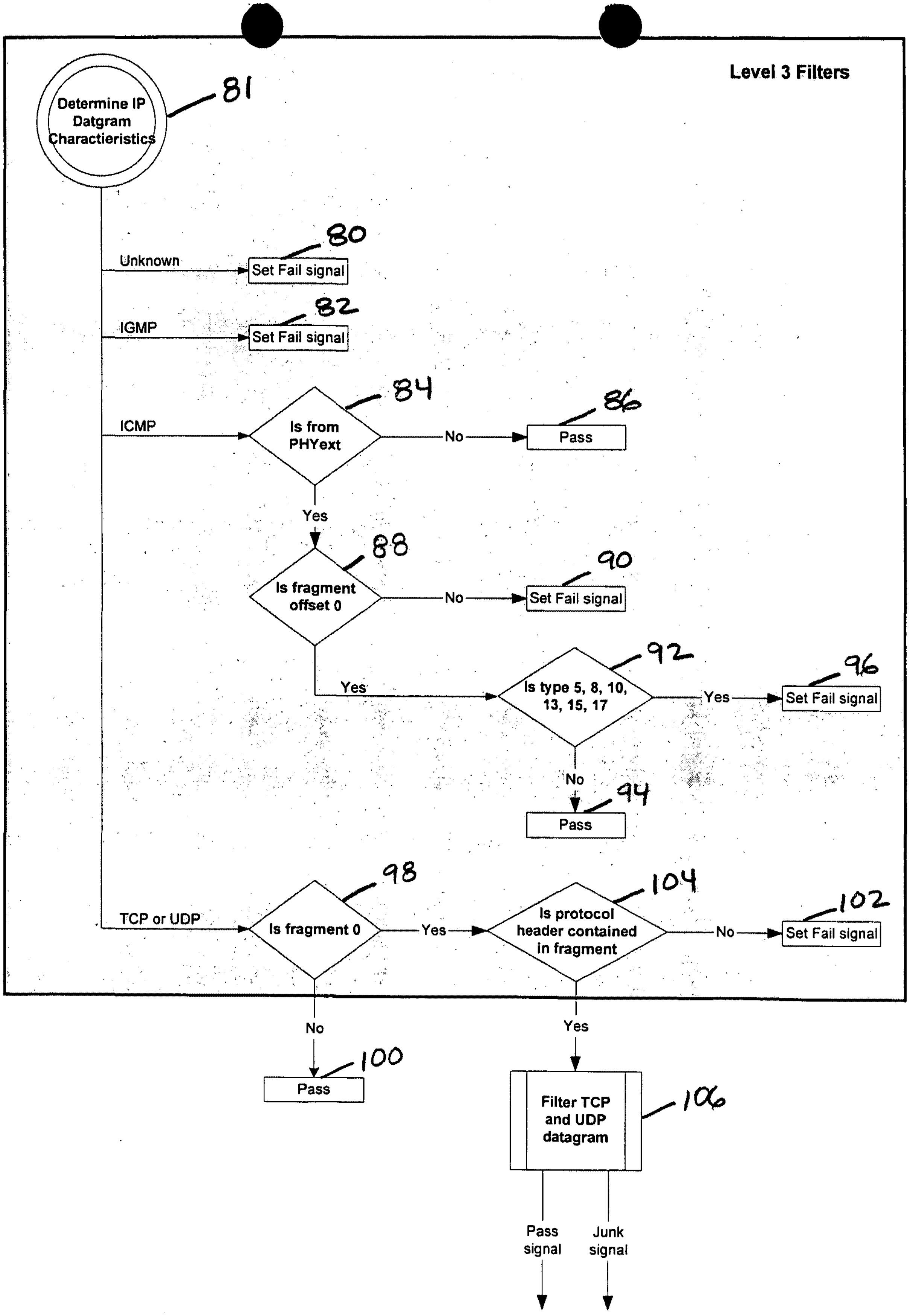

Level 3 Filters
Determine IP Datgram Charactieristics
81
Unknown
Set Fail signal
80
IGMP
Set Fail signal
82
ICMP
Is from PHYext
84
No
Pass
86
Yes
Is fragment offset 0
88
No
Set Fail signal
90
Yes
Is type 5, 8, 10, 13, 15, 17
92
Yes
Set Fail signal
96
No
Pass
94
TCP or UDP
Is fragment 0
98
Yes
Is protocol header contained in fragment
104
No
Set Fail signal
102
No
Pass
100
Yes
Filter TCP and UDP datagram
106
Pass signal
Junk signal

FIG. 5

TCP and UDP packets are evaluated for pass or fail in parallel (other protocols also handled simultaneously)

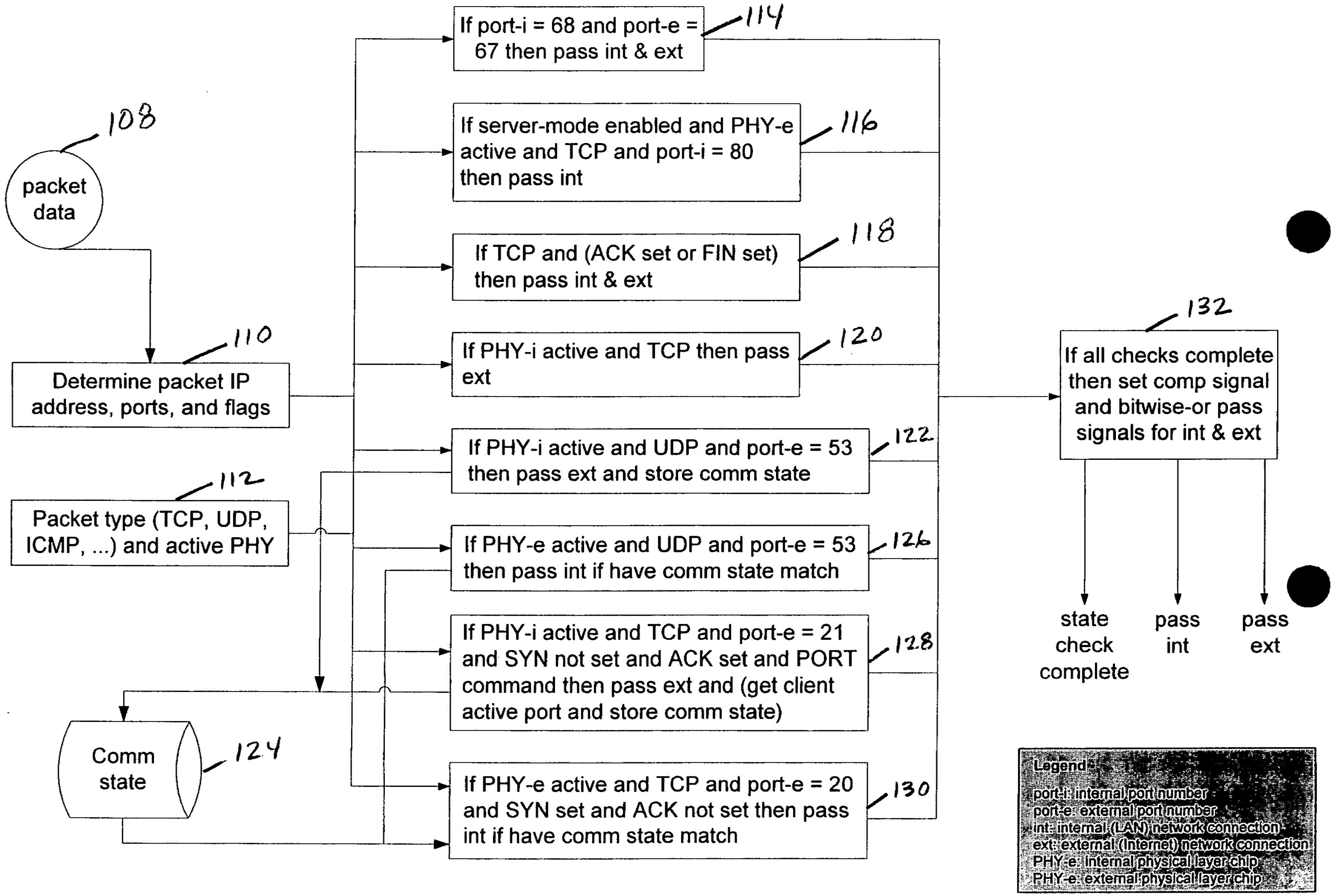

108
packet data
110
Determine packet IP address, ports, and flags
112
Packet type (TCP, UDP, ICMP, ...) and active PHY
124
Comm state
114
If port-i = 68 and port-e = 67 then pass int & ext
116
If server-mode enabled and PHY-e active and TCP and port-i = 80 then pass int
118
If TCP and (ACK set or FIN set) then pass int & ext
120
If PHY-i active and TCP then pass ext
122
If PHY-i active and UDP and port-e = 53 then pass ext and store comm state
126
If PHY-e active and UDP and port-e = 53 then pass int if have comm state match
128
If PHY-i active and TCP and port-e = 21 and SYN not set and ACK set and PORT command then pass ext and (get client active port and store comm state)
130
If PHY-e active and TCP and port-e = 20 and SYN set and ACK not set then pass int if have comm state match
132
If all checks complete then set comp signal and bitwise-or pass signals for int & ext
state check complete
pass int
pass ext
Legend:
port-i: internal port number
port-e: external port number
int: internal (LAN) network connection
ext: external (Internet) network connection
PHY-e: internal physical layer chip
PHY-e: external physical layer chip

FIG. 6

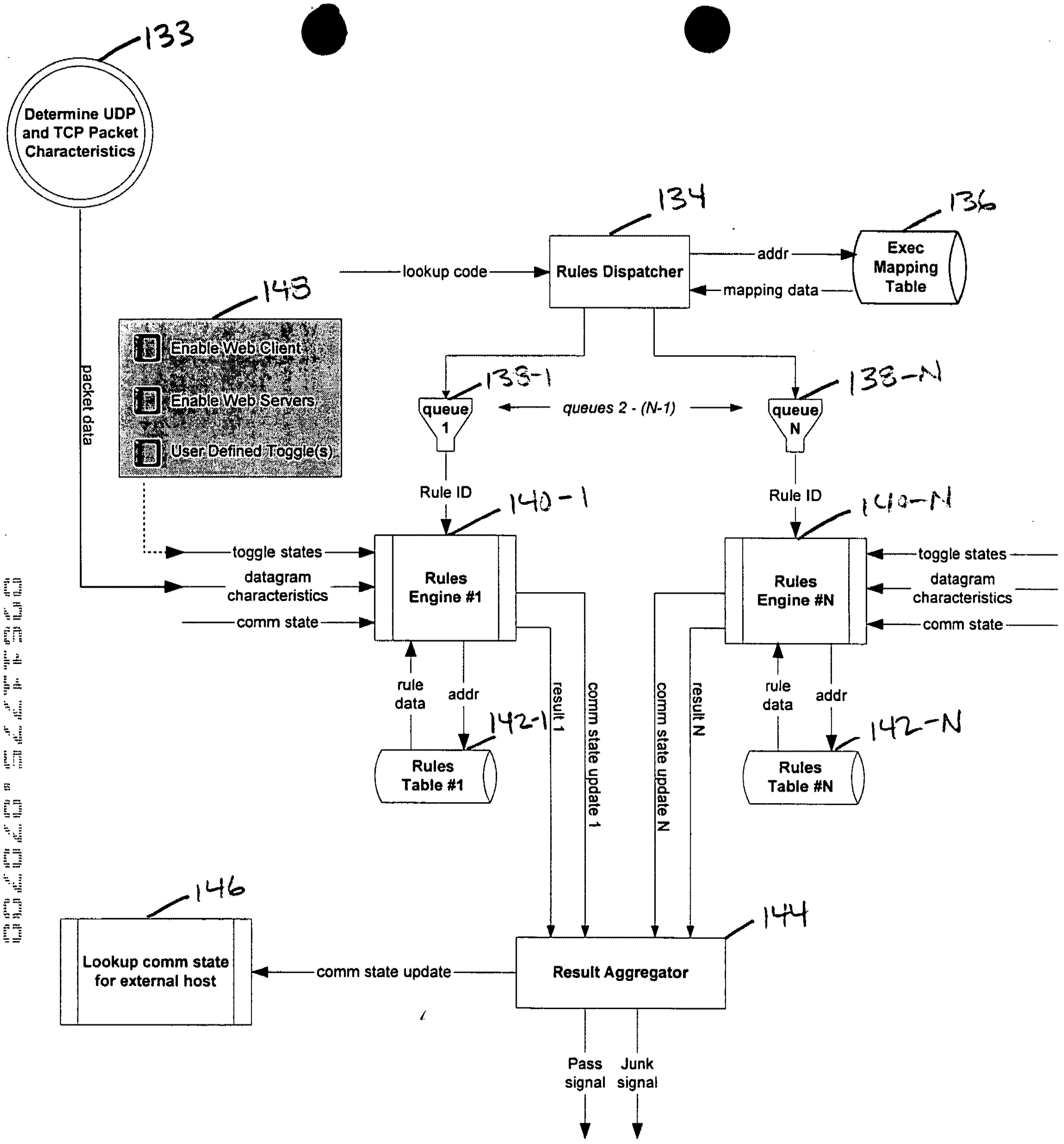

133
Determine UDP and TCP Packet Characteristics
packet data
134
lookup code
Rules Dispatcher
addr
mapping data
136
Exec Mapping Table
143
Enable Web Client
Enable Web Servers
User Defined Toggle(s)
138-1
queue 1
queues 2 - (N-1)
138-N
queue N
Rule ID
140-1
Rule ID
140-N
toggle states
datagram characteristics
comm state
Rules Engine #1
Rules Engine #N
toggle states
datagram characteristics
comm state
rule data
addr
142-1
Rules Table #1
result 1
comm state update 1
comm state update N
result N
rule data
addr
142-N
Rules Table #N
146
Lookup comm state for external host
comm state update
144
Result Aggregator
Pass signal
Junk signal

FIG. 7

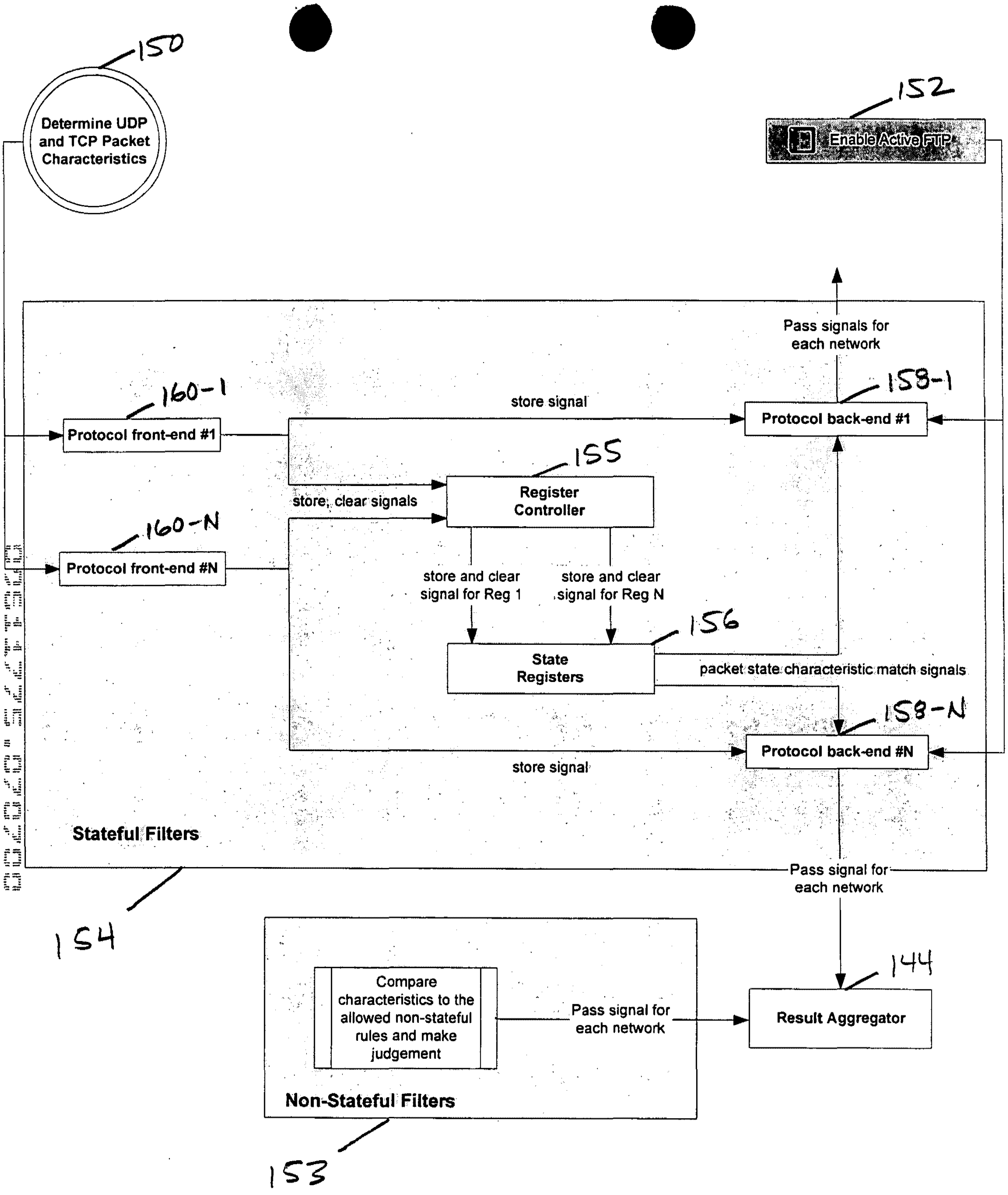

150
Determine UDP and TCP Packet Characteristics
152
Enable Active FTP
Pass signals for each network
160-1
Protocol front-end #1
store signal
158-1
Protocol back-end #1
155
Register Controller
store, clear signals
160-N
Protocol front-end #N
store and clear signal for Reg 1
store and clear signal for Reg N
156
State Registers
packet state characteristic match signals
158-N
Protocol back-end #N
store signal
Stateful Filters
Pass signal for each network
154
144
Compare characteristics to the allowed non-stateful rules and make judgement
Pass signal for each network
Result Aggregator
Non-Stateful Filters
153

FIG. 8

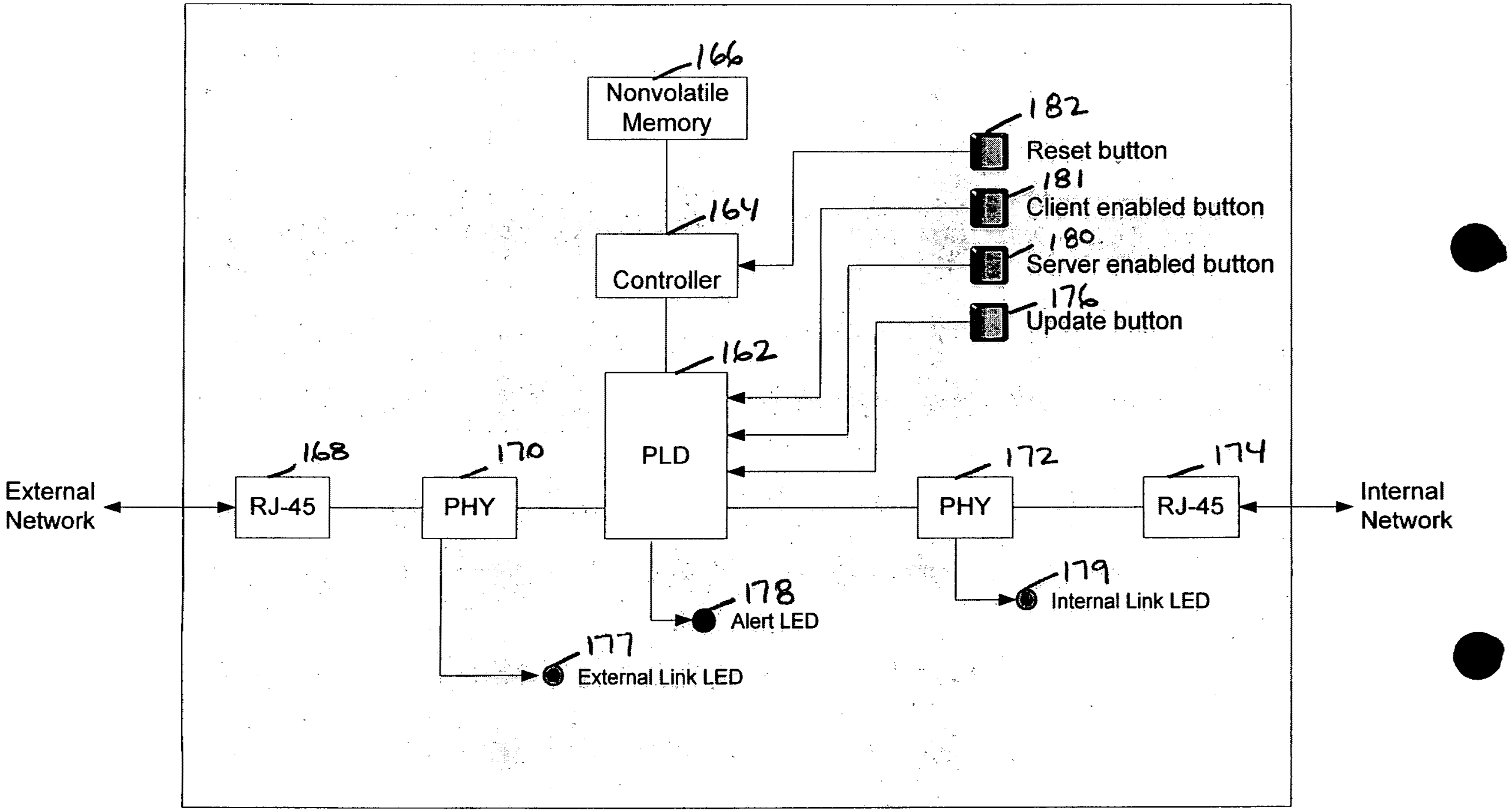

166
Nonvolatile Memory
164
Controller
162
PLD
182
Reset button
181
Client enabled button
180
Server enabled button
176
Update button
External Network
168
RJ-45
170
PHY
172
PHY
174
RJ-45
Internal Network
179
Internal Link LED
178
Alert LED
177
External Link LED

FIG. 9

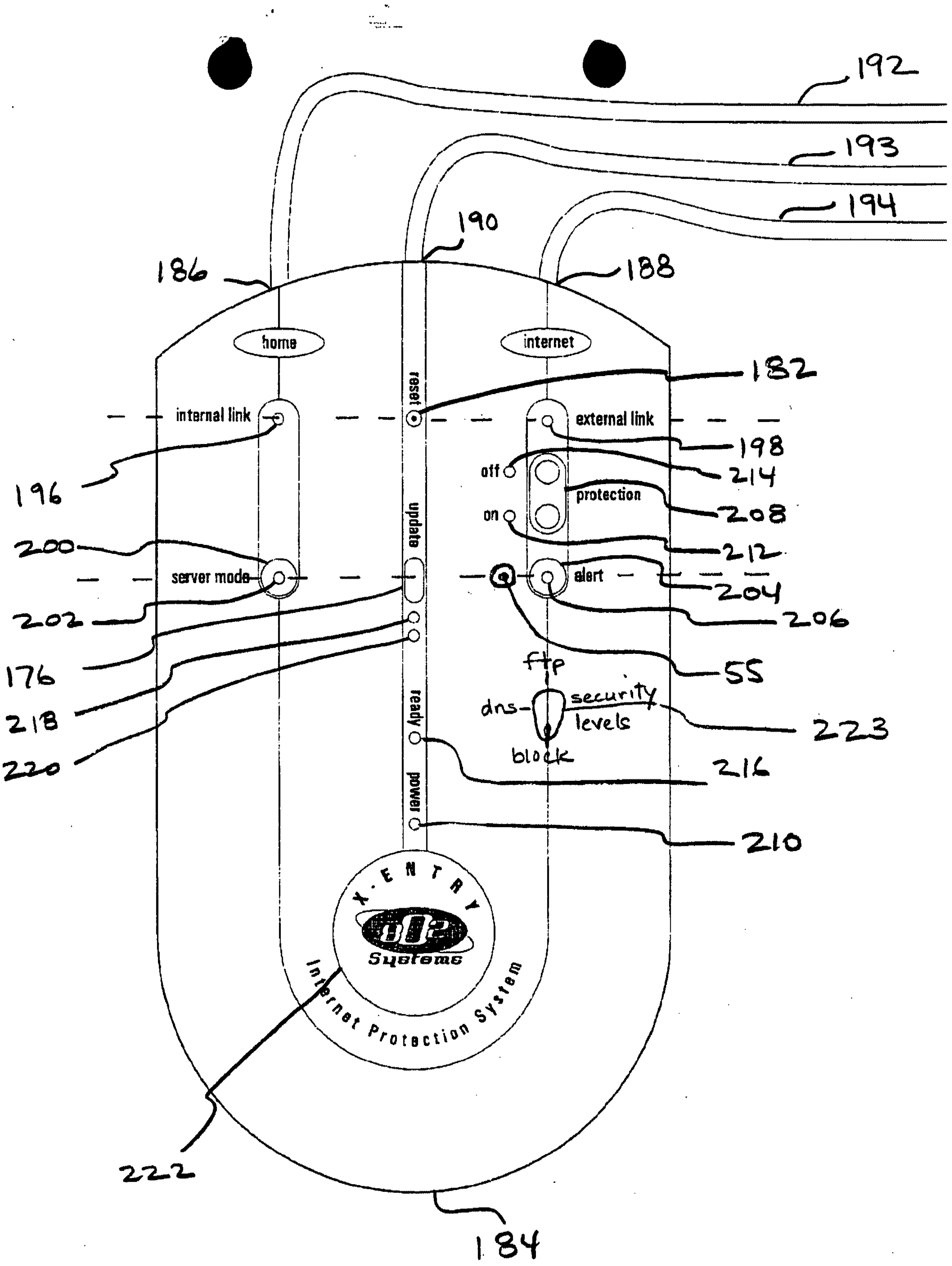

192
193
194
190
186
188
home
internet
reset
182
internal link
external link
198
off
214
protection
208
196
update
on
212
200
server mode
alert
204
202
206
176
55
ftp
218
dns-
security levels
223
220
block
ready
216
power
210
X·ENTRY
Systems
Internet Protection System
222
184

FIG.10

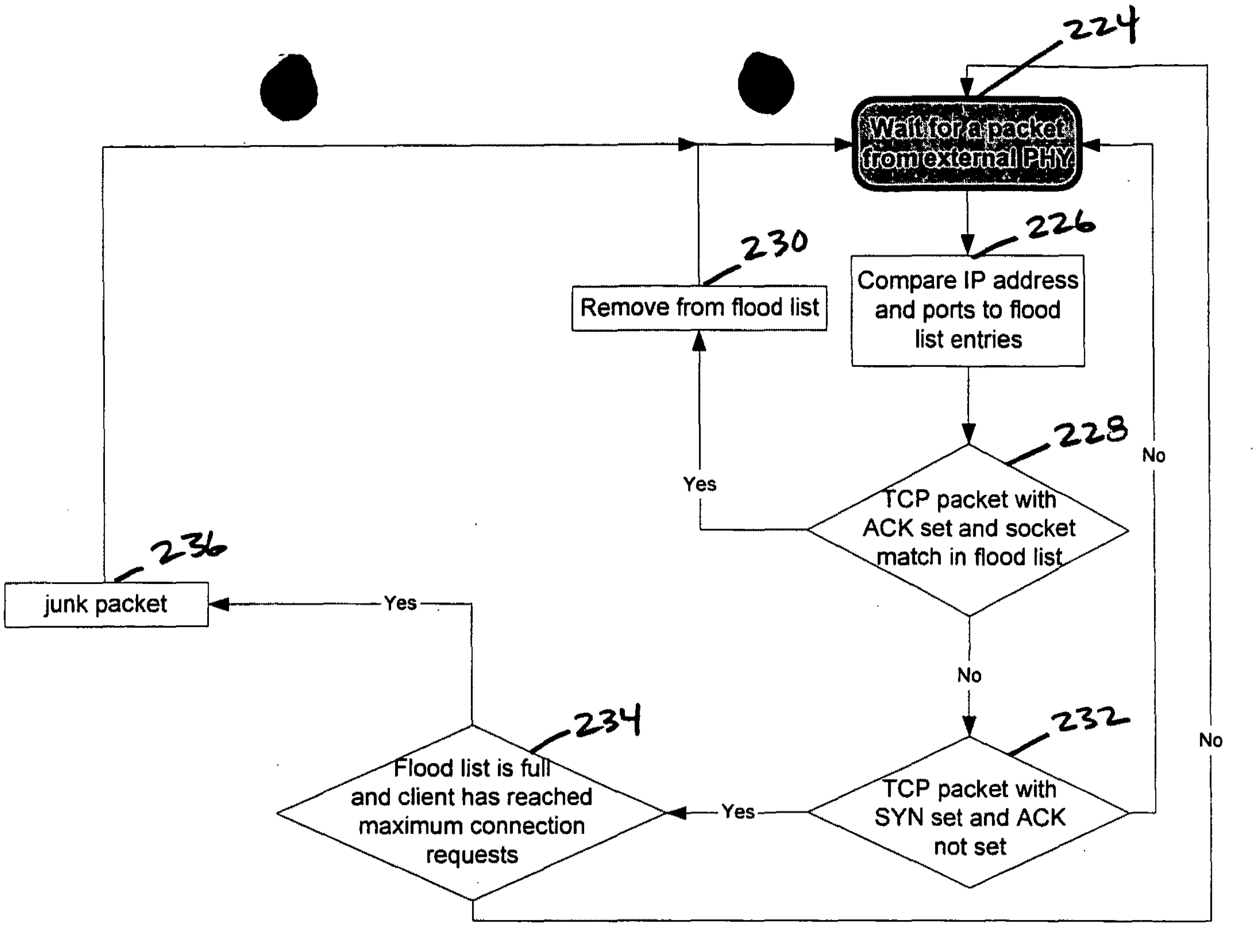

224
Wait for a packet from external PHY
226
Compare IP address and ports to flood list entries
230
Remove from flood list
228
TCP packet with ACK set and socket match in flood list
Yes
No
No
236
junk packet
Yes
234
Flood list is full and client has reached maximum connection requests
232
TCP packet with SYN set and ACK not set
Yes
No
No

FIG. 11

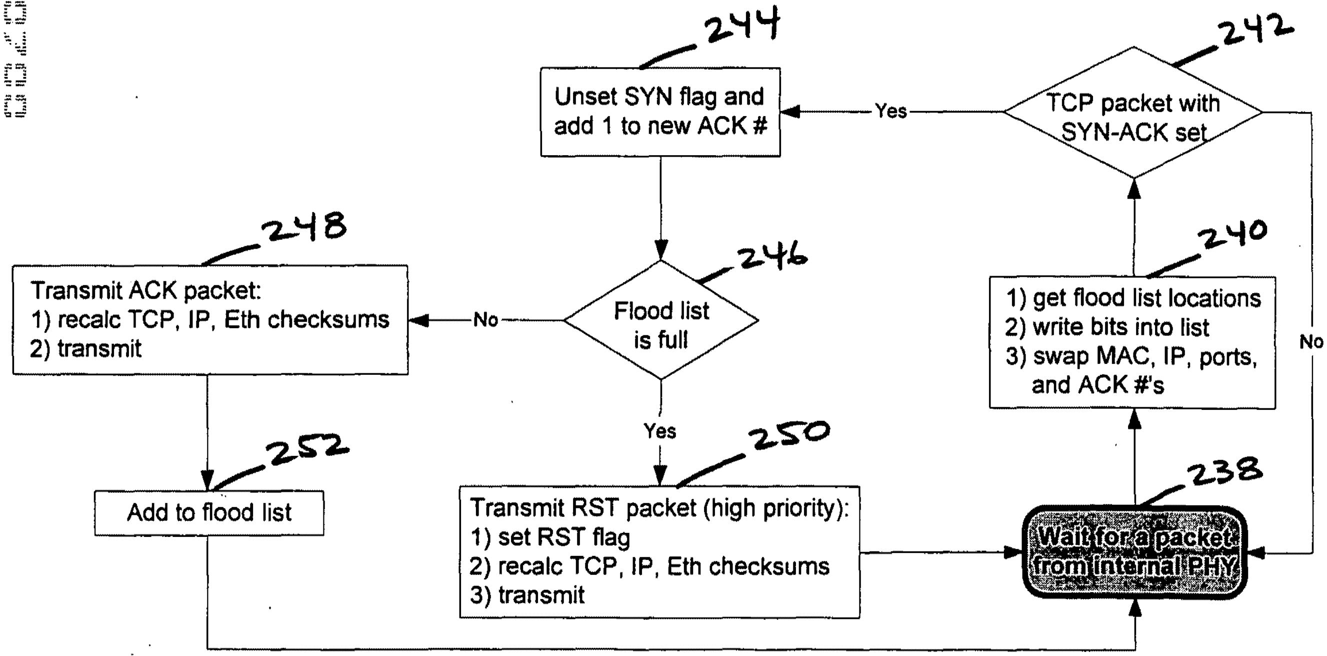

244
Unset SYN flag and add 1 to new ACK #
Yes
242
TCP packet with SYN-ACK set
248
Transmit ACK packet:
1) recalc TCP, IP, Eth checksums
2) transmit
No
246
Flood list is full
240
1) get flood list locations
2) write bits into list
3) swap MAC, IP, ports, and ACK #'s
No
Yes
252
250
Add to flood list
Transmit RST packet (high priority):
1) set RST flag
2) recalc TCP, IP, Eth checksums
3) transmit
238
Wait for a packet from internal PHY

FIG. 12

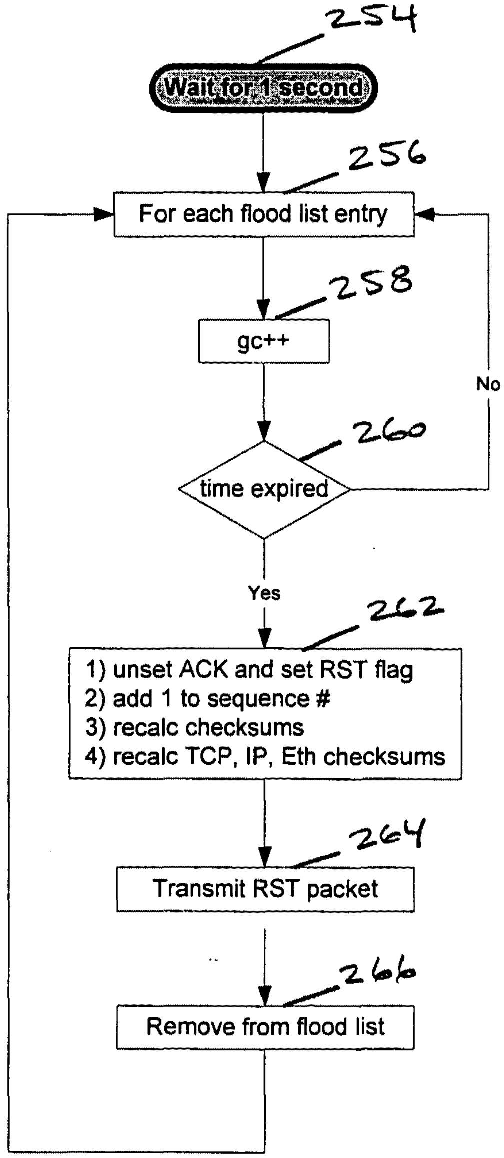

254
Wait for 1 second
256
For each flood list entry
258
gc++
No
260
time expired
Yes
262
1) unset ACK and set RST flag
2) add 1 to sequence #
3) recalc checksums
4) recalc TCP, IP, Eth checksums
264
Transmit RST packet
266
Remove from flood list

FIG. 13

REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

Field of the Invention

The present invention relates to computer security and data protection systems and methods, and more particularly to firewall and data protection systems and methods for filtering packets, such as from the Internet, in real time and without packet buffering.

Background of the Invention

The use of the Internet has exploded in recent years. Small and large companies as well as individual users are spending more time with their computers connected to the Internet. With the advent of Internet technologies, such as cable modems, digital subscriber lines, and other "broadband" access devices, users are connecting their computers to the Internet for extended periods of time.

Such extended or "persistent" connection to the Internet brings many advantages to users in immediate access to the content on the Internet through the use of email, search engines, and the like. Unfortunately, however, persistent access to the Internet exposes connected computers to potential security threats, where intruders and "hackers" may compromise proprietary systems, engage in information theft, or take control of the connected computers remotely. With more sophisticated tools at their disposal, hackers pose security and privacy risks to systems with persistent access to the Internet. Such security risks are even present for computers connected to the Internet for limited periods of time (such as through dial-up, modem connections), though to a lesser degree than the extended access computers.

There are currently many different types of firewall systems available on the market, including proxy servers, application gateways, stateful inspection firewalls, and packet filtering firewalls, each of which provides a variety of strategies and services for data protection. Conventional packet filters typically are computers, routers, or ASICs based on general purpose CPUs. They perform their filtering duties by receiving a packet, buffering the data until a determination can be made, and forwarding the packet as applicable for the particular system.

For example, a dual-homed, Linux-based filter with two network cards might receive a packet completely, evaluate whether it meets specific criteria, and transmit the packet on the other network card. In another example, a router designed for switch mode routing might begin buffering a packet until a decision is made, then forward the packet on the applicable interface while still receiving the packet. With most packet filters, software is used and data is buffered.

Sophisticated computer users working for medium- to large-sized companies have a variety of relatively expensive protection devices and tools at their disposal. Such devices and tools typically screen data packets received from the Internet with sophisticated software-based filtering techniques. Using relatively complex tools for software analysis, each packet is stored in a buffer and examined sequentially with software-based rules, which results in each packet being either accepted (and passed to the computer) or rejected (and disposed of by the software). This software often requires substantial computer knowledge and experience. Users of such devices and tools typically have an expertise in network administration or a similar field, so they can configure, optimize, and even build the complex filtering and security options provided by the software.

While such devices and tools can be quite effective in providing "firewall" protection for sophisticated users of large office systems, they pose several barriers to unsophisticated users of small office and home systems in the growing SOHO market. Current large office systems are expensive, difficult to set up, and require technical skills. What is needed for SOHO systems is a relatively inexpensive, uncomplicated, "plug and play" type of Internet protection system that can be easily connected and configured by relatively unsophisticated users.

<u>Summary of the Invention</u>

In accordance with the present invention, devices, methods and systems are provided for the filtering of Internet data packets in real time and without packet buffering. A stateful packet filtering hub is provided in accordance with preferred embodiments of the present invention. The present invention also could be implemented as part of a switch or incorporated into a router.

A packet filter is a device that examines network packet headers and related information, and determines whether the packet is allowed into or out of a network. A stateful packet filter, however, extends this concept to include packet data and previous network activity in order to make more intelligent decisions about whether a packet should be allowed into or out of the network. An Ethernet hub is a network device that links multiple network segments together at the medium level (the medium level is just above the physical level, which connects to the network cable), but typically provides no capability for packet-type filtering. As is known, when a hub receives an Ethernet packet on one connection, it forwards the packet to all other links with minimal delay and is accordingly not suitable as a point for making filtering-type decisions. This minimum delay is important since Ethernet networks only work correctly if packets travel between hosts (computers) in a certain amount of time.

In accordance with the present invention, as the data of a packet comes in from one link (port), the packet's electrical signal is reshaped and then transmitted down other links. During this process, however, a filtering decision is made between the time the first bit is received on the incoming port and the time the last bit is transmitted on the outgoing links. During this short interval, a substantial number of filtering rules or checks are performed, resulting in a determination as to whether the packet should or should not be invalidated by the time that the last bit is transmitted. To execute this task, the present invention performs multiple filtering decisions simultaneously: data is received; data is transmitted; and filtering rules are examined in parallel and in real time. For example, on a 100 Mbit/sec Ethernet network, 4 bits are transmitted every 40 nano seconds (at a clock speed of 25 MHz). The present invention makes a filtering decision by performing the rules evaluations simultaneously at the hardware level, preferably with a programmable logic device.

The present invention may employ a variety of networking devices in order to be practical, reliable and efficient. In addition, preferred embodiments of the present invention may include constituent elements of a stateful packet filtering hub, such as microprocessors, controllers, and integrated circuits, in order to perform the real time, packet-filtering, without requiring buffering as with conventional techniques. The present invention preferably is reset, enabled, disabled, configured and/or reconfigured with relatively simple toggles or other physical switches, thereby removing the requirement for a user to be trained in sophisticated computer and network configuration. In accordance with preferred embodiments of the present invention, the system may be controlled and/or configured with simple switch activation(s).

Accordingly, one object of the present invention is to simplify the configuration requirements and filtering tasks of Internet firewall and data protection systems.

Another object is to provide a device, method and system for Internet firewall and data protection that does not require the use of CPU-based systems, operating systems, device drivers, or memory bus architecture to buffer packets and sequentially carry out the filtering tasks.

A further object of the present invention is to perform the filtering tasks of Internet firewall protection through the use of hardware components.

Another object is to utilize programmable logic for filtering tasks.

Still another object is to provide a device, method, and system to carry out bitstream filtering tasks in real time.

Yet another object is to perform parallel filtering, where packet data reception, filtering, and transmission are conducted simultaneously.

A further object of the present invention is to perform the filtering tasks relatively faster than current state-of-the-art, software-based firewall/data protection systems.

Another object is to provide a device, method and system for firewall protection without the use of a buffer or temporary storage area for packet data.

Still another object of the present invention is to design a device, method and system that does not require software networking configurations in order to be operational.

A further object of the present invention is to provide a device, method and system for Internet firewall and data security protection that supports partitioning a network between client and server systems.

It is a yet another object of the present invention to provide a device, method and system for Internet firewall and data protection that supports multiple networking ports.

Another object is to maintain stateful filtering support for standard data transmission protocols on a per port basis.

Still another object of is to configure network functionality using predefined toggles or other types of physical switches.

A further object of the present invention is to conduct packet filtering without requiring a MAC address or IP address to perform packet filtering.

Yet another object of the present invention is to facilitate the shortest time to carry out bitstream filtering tasks.

Finally, it is another object of the present invention to be able to perform filtering rules out of order and without the current state-of-the-art convention of prioritizing the filtering rules serially.

Brief Description of the Drawings

The present invention may be more fully understood by a description of certain preferred embodiments in conjunction with the attached drawings in which:

FIGS. 1A and 1B are application level diagrams illustrating exemplary data protection systems in accordance with the present invention;

FIG. 2 is a flow diagram illustrating the components and operations of a preferred embodiment of the present invention;

FIG. 3 is a flow chart illustrating the basic functions of a repeater core and four filter levels in accordance with preferred embodiments of the present invention;

FIG. 4 is a diagram illustrating filtering functions of Level 2 filters in relation to the flow of packet data from internal and external networks in accordance with preferred embodiments of the present invention;

FIG. 5 is a flow chart illustrating packet filtering functions of Level 3 filters in accordance with preferred embodiments of the present invention;

FIG. 6 illustrates the rules by which TCP and UDP packets are evaluated in parallel in accordance with preferred embodiments of the present invention;

FIG. 7 is a diagram illustrating parallel rule evaluation for TCP and UDP packets in accordance with preferred embodiments of the present invention;

FIG. 8 is a flow chart illustrating packet filtering functions of Level 4 filters in accordance with preferred embodiments of the present invention;

FIG. 9 is a block diagram of the hardware components of a preferred embodiment of the present invention;

FIG. 10 is an illustration of an exemplary design of an external case in accordance with preferred embodiments of the present invention;

FIGS. 11 and 12 are flow diagrams illustrating SYN flood protection in accordance with preferred embodiments of the present invention; and

FIG. 13 is a flow chart illustrating the process of "garbage collection" in flood lists in accordance with preferred embodiments of the present invention.

Detailed Description of the Preferred Embodiments

The present invention will be described in greater detail with reference to certain preferred and alternative embodiments. As described below, refinements and substitutions of the various embodiments are possible based on the principles and teachings herein.

FIG. 1A and FIG. 1B illustrate the physical positioning of a stateful packet filtering hub in accordance with the present invention in two exemplary network configurations. The packet filtering hub of the illustrated embodiments preferably serves as an Internet firewall/data protection system (hereafter "data protection system").

With reference to FIG. 1A, in the illustrated embodiment data protection system 1 is coupled through a port to router 2 (or cable modem or other preferably broadband, persistent network connection access device), which is linked through a broadband connection to other computer systems and networks, exemplified by Internet 8 and Internet Service Provider (ISP) 10. Packets of data are transmitted from an ISP, such as ISP 10, via Internet 8 to router 2. The packets are transmitted to data protection system 1, which analyzes the packets in "real time" and without buffering of the packets, while at the same time beginning the process of transmitting the packet to the internal network(s) in compliance with the timing requirements imposed by the Ethernet or other network standards/protocols. If a packet of data satisfies the criteria of the rules-based filtering performed within data protection system 1, which is executed in a manner to be completed by the time the entire packet has been received by data protection system 1, then it is allowed to pass to hub 6 as a valid packet, which may then relay the cleared packet to computers 4a, 4b, 4c, etc. on the internal network. If a packet of data fails to meet the filtering criteria, then it is not allowed to pass as a valid packet and is "junked." Junking is defined as changing bits or truncating data, depending on the type of link, in a manner such that the packet is corrupted or otherwise will be detected by the receiving computers as invalid or unacceptable, etc. Without the intermediate positioning of data protection system 1, the packets would be transmitted directly to unprotected hub 6, thereby exposing computers 4a, 4b and 4c to security risks. It should also be noted that hub 6 is optional in accordance with the present invention; in other embodiments, data protection system 1 may be directly connected to a single computer or may have multiple ports that connect to multiple computers. Similar filtering is performed on packets that are to be transmitted from computers 4a, 4b, and 4c to Internet 8.

With reference to FIG 1B, in this illustrated embodiment data protection system 1 is coupled via one port to DSL router 2 (again, the network access device is not limited to a DSL router, etc.), which provides the broadband connection to Internet 8. As with the embodiment of FIG. 1A, data protection system 1 also is coupled to a number of computers 4a, 4b, etc., on the internal network, and serves to provide filtering for packets between computers 4a and 4b and Internet 8 in the manner described in connection with FIG. 1A. In this embodiment, data protection system 1 is also connected via another port to hub 6, which serves as the main point of contact for incoming connections from the Internet for bastion hosts 5a and 5b, etc. In accordance with this embodiment, packets are transmitted to router 2 and then to data protection system 1. If the packets are approved by data protection system 1 (i.e., passing the filtering rules/checks performed with data protection system 1 while the packet is being received and transmitted), then the packets are allowed to pass as valid packets to computers 4a, 4b and hub 6. (The rules-based filtering process of preferred embodiments of the present invention will be described in more detail hereinafter.) Hub 6 may relay the packets to other internal host computers 5a, 5b, etc., on the local area network (LAN). These computers may include, for example, a Web and FTP server 5a, or a streaming audio server 5b, etc. Thus, in accordance with the illustrated embodiment, packets that passed the filtering rules/checks are passed as valid packets to computers, such as protected internal host computer 4a, which as illustrated may be connected to printer 7. In this particular embodiment, a bastion port is provided that may be used to service more than one bastion host. In other embodiments, different network configurations may be utilized in accordance with the present invention.

FIG. 2 illustrates the general components and operations of certain preferred embodiments of the present invention. Connection to external network 12 is made by physical interface 14..Physical interface (or PHY) 14 preferably is implemented with commercially available, physical layer interface circuits, as are known in the art (such physical layer interface circuits may be off-the-shelf components, as specified in the Ethernet IEEE standard 802.3u.). At a minimum, the data protection system must contain two PHY interfaces, one for the Internet or other external network connection, and one (or more) for the internal network. It should be noted that, in preferred embodiments, PHY controllers are utilized, which implicitly assumes Ethernet-type connections. In other embodiments in accordance with the present invention, other types of PHY interfaces and controllers are utilized for different networking standards.

Repeater core 16 functions as an Ethernet repeater (as defined by the network protocols of the IEEE standard 802.3) and serves to receive packets from external PHY 14, reshape the electrical signals thereof, and transmit the packets to internal PHY 18, which is coupled to internal network 20. While the packet is being received, reshaped, and transmitted between PHYs 14 and 18, however, it is simultaneously being evaluated in parallel with filtering rules to determine if it should be allowed to pass as a valid packet (as will be described in greater detail elsewhere herein). As with the discussion regarding the PHY interfaces and controllers, changes in networking standards may alter the components functionality (such as the characteristics of repeater core 16), but not the basic parallel, real time packet filtering in accordance with the present invention. (In an alternate embodiment, for example, the data protection system may use switch logic or router logic; in full duplex, the same principles apply.)

The parallel filtering preferably consists of packet characteristics logic 22, packet type filters 26, and state rules filters 42. Packet characteristics logic 22 determines characteristics based on packet data (preferably in the form of 4-bit nibbles from PHY 14), whereas packet type filters 26 make filtering decisions generally based on packet type. State rules filters 42 perform rules- based filtering on several levels simultaneously. The results of filtering by packet type filters 26 and state rules filters 42 are combined by aggregator 24, which may be considered a type of logical operation of pass/fail signals (described in greater detail elsewhere herein). In preferred embodiments, if any one or more of the performed filtering rules indicates that the packet should be failed (or not allowed to pass as a valid packet), then the output of aggregator 24 is a fail; otherwise, the packet is allowed and the output of aggregator 24 is a pass. Thus, as packet data is being received and transmitted from PHY 14 to PHY 18 via repeater core 16, it is being evaluated in parallel via packet type filters 26 and state rules filters 42 (depending in part on packet characteristics determined by logic 22 from the data received from PHY 14). In accordance with the present invention, the results of filtering by packet type filters 26 and state rules filters 42 are provided to aggregator 24 by the time that the entire packet reaches repeater core 16, so that, based on the output of aggregator 24, the packet will either be allowed to pass as a valid packet or will be failed and junked as a suspect (or otherwise invalidated) packet.

Packet characteristics logic 22 receives packet data from PHY 14 and examines the packet data to determine characteristics, such as the packet type, datagram boundaries, packet start, packet end, data offset counts, protocols, flags, and receiving port. The packet type may include, for example, what are known in the art as IP, TCP, UDP, ARP, ICMP, or IPX/SPX. Such packet characteristics data is provided to packet type filters 26. Packet type filters 26 preferably make a decision about whether the packet should be passed or failed, with the result being transmitted to aggregator 24. In accordance with preferred embodiments, packet type filters 26 do not require the use of what may be considered an extensible rules system. The filters of packet type filters 26 preferably are expressed as fixed state machines or may be expressed using more flexible rules syntax. What is important is that packet type filtering is performed by filters 26 in the shortest time interval possible and in parallel with the packet data being received and transmitted to internal PHY 18, so that a pass/fail determination may be made prior to the time when the entire packet has been received by repeater core 16.

State rules filters 42 receive packet characteristics data from logic 22 and, based on this data as well as cached/stored connection and communication state information, executes a plurality of rules under the control of rules controller 28, preferably using a plurality of rules engines 36-1 to 36-N, so that a desired set of filtering decisions are promptly made and a pass/fail determination occurs before the entire packet has been received by repeater core 16. State rules filters 42 preserve a cache of information 30 about past network activity (such as IP addresses for established connections, port utilization, and the like), which is used to maintain network connection state information about which hosts have been exchanging packets and what types of packets they have exchanged, etc. Rules controller 28 preferably accesses rules map table 32 based on packet characteristics information, which returns rules dispatch information to rules controller 28. Thus, based on the connection state information stored in connection cache 30 and the characteristics of the packet being examined, rules controller 28 initiates filtering rules via a plurality of rules engines 36-1 to 36-N that simultaneously apply the desired set of filtering rules in parallel. (Preferably, N is determined by the number of rules that need to be performed in the available time and the speed of the particular logic that is used to implement state rules filters 42.)

As will be appreciated, while the packet pass/fail decision is being made in real time, and thus must be concluded by the time that the entire packet has been received, a large of number of filtering rules must be performed quickly and in parallel. Preferably, rules controller 28 utilizes a plurality of rules engines 36-1 to 36-N, which logically apply specific rules retrieved from corresponding storage areas 40-1 to 40-N. Rules controller 28, based on the connection state and packet characteristics, determines which rules should be run based on which information. The rules to be run are then allocated by rules controller 28 to the available rules engines 36-1 to 36-N. As each rules engine 36-1 to 36-N may be required to execute multiple rules in order to complete the filtering decision process in the required time, corresponding queues 34-1 to 34-N are preferably provided. Thus, rules controller 28 determines the list of rules that should be performed (again, depending on the stored connection state and packet characteristics data) and provides the list of rules (and accompanying information to carry out those rules) to the plurality of rules engines 36-1 to 36-N via queues 34-1 to 34-N. Rules engines 36-1 to 36-N, based on the information from the queues 34-1 to 34-N, look up specific rule information from storage areas 40-1 to 40-N, carry out the rules, and preferably return the results to rules controller 28. As the rules are essentially conditional logic statements that notify the data protection system how to react to a particular set of logical inputs, it has been determined that providing a plurality of rules engines may enable the necessary decision making process to quickly provide the outcome of the rules-based filtering by the time the entire packet has been received.

Still referring to FIG. 2, rules controller 28 preferably uses rules map table 32 to dispatch the rules to rules engines 36-1 and 36-N, so that a filtering decision may be reached in the optimal amount of time. In a preferred operation, each rules engine extracts a rule ID from its queue, looks up the rules definition in its own rules table 40-1 to 40-N, evaluates the rule, returns the result to rules controller 28, and looks for another rule ID in its queue 34-1 to 34-N. The results from packet type filter 26 and rules controller 28 are combined into one result via aggregator 24: pass or fail. If a decision is not reached before the end of the packet is transmitted, then in preferred embodiments the packet will be processed as an invalid packet and junked.

It should be appreciated that the data protection system must make a filtering determination before the current packet is completely transmitted. Since the networking standards impose strict timing thresholds on the transit delay of packets, filtering is performed in real time, in parallel and without buffering the packet. (The transit delay threshold is the time it takes to get from the transmitting station to the receiving station.) Given that a filtering decision must be made in real time (before the last bit is received and forwarded to the applicable interfaces), the filter rules are evaluated in parallel by rules engines that possess independent, direct access to the rules set collected in storage areas 40-1 and 40-N, which are preferably implemented as RAM tables. (In a preferred embodiment of the data protection system, the tables are implemented using on-chip, dual port RAM up to 4K in size. A programmable logic device, such as Xilinx Spartan II XC2S100, has 40K dual port synchronous block RAM. For example, an initial 110-bit segment of the rules controller RAM block may be a range table that delineates where each look up code begins and what the number of entries are.) Rules controller 28 dispatches the rules to each rules engine by placing a rules ID entry in a queue. Because each rules engine is assigned its own queue, a pipeline is created allowing the rules engine to continuously run and operate at maximum efficiency.

To operate efficiently the rules engines must also be capable of evaluating rules in any order. In accordance with the preferred embodiments, each rule has a priority and the highest priority result is accepted. Therefore, the rules must be evaluated in any order yet still obtain the same result, as if the rules were being evaluated serially from highest to lowest priority. This operation is accomplished in preferred embodiments by rules map table 32, which notifies rules controller 28 which rule is assigned to which rules engine. Thus, this decision is statically determined by the rules set and the number of rules engines. It should be noted that the rule set in general is greater than the number of rules engines.

FIG. 3 is a flow chart illustrating further aspects of preferred embodiments of the present invention. As previously described, preferred embodiments of the data protection system utilize programmable logic, or other suitable preferably hardware-based logic, to perform a large number of filter rules in parallel and at high speed. Such embodiments may be considered to provide an external interface, for instance, to the Internet, to external network 12, and one or more internal network interfaces, such as to internal network 20 and/or to bastion network 15 (see, for example, FIGS. 1A and 1B). As repeater core 16 (or the PHYs in FIG. 2) receives and transmits packet data, the packet is simultaneously subjected to a plurality of filtering rules. At step 44, the packet characteristics are determined (which, as previously described, may include protocol, addresses, ports, flags, etc.). The filtering rules are based on the packet characteristics, connection state information (depending upon the particular rules), and/or toggle or other physical switch state information. This filtering process may be represented by filtering steps 46, 48, 50 and 52, which, as depicted in FIG. 3, are performed at least in substantial part in parallel, and thus can make filtering decisions by the time the packet has been completely received.

As illustrated, after the packets are transmitted to repeater core 16, their characteristics are analyzed at step 44. Data packets generally consist of several layers of protocols that combine to make a protocol stack. Preferably, each layer of the stack is decoded and the information is passed to various filter blocks, as exemplified in steps 46, 48, 50 and 52. In accordance with the present invention, this filtering process is executed in parallel and in real time. In other embodiments, a variety of filter blocks or rules-based filters may be employed, incorporating parallel execution, real time filtering, etc., as may be necessary to complete the filtering decision in the required time.

Referring again to preferred embodiments illustrated in FIG. 3, Level 2 filters at step 46 may examine information in the link layer header for all incoming packets and decide whether a packet should be junked based on the packet protocol. Level 3 filters at step 48 may examine information in the networking layer headers. (For the IP protocol, these headers would equate to the ARP, RARP, IP, ICMP, and IGMP protocol headers.) While Level 2 filters preferably distinguish the packet type, Level 3 filters at step 48 and Level 4 filters at step 50 preferably distinguish IP datagram characteristics. Level 4 filters at step 50 preferably operate by examining IP, TCP and UDP headers along with data transmitted between the client and server processes, utilizing two techniques: stateful and non-stateful packet filtering. (Level 2, 3 and 4 filters are described in greater detail elsewhere herein.) Preferably a spoof check filter at step 52 detects whether the packet originated from an authorized IP address or not. To determine whether the packet should be allowed to pass as a valid packet, the filters must implement rules in parallel preferably based on programmable logic and register one of two values: pass or fail. After the values are registered, the outcome is collected in result aggregator 24, which logically combines the results to determine if the packet should be allowed to pass as a valid packet or should be denied as an invalid one. If the packet is passed, then repeater core 16 continues to send correct bits. If the packet is failed, then it is junked.

In accordance with preferred embodiments of the present invention as illustrated in FIG. 3, a spoof check is performed at step 52 on all packets entering a port. To prevent IP spoofing, the spoof check filtering of step 52 monitors IP addresses from the internal network and discards any incoming packets with IP source addresses that match internal IP addresses. A spoof check ensures that a host on one network is not trying to impersonate a computer on another network, such as a computer on the Internet assuming the IP address of a computer connected to an internal port. In accordance with preferred embodiments, spoofed packets are always junked by the data protection system. In such embodiments, the data protection system performs this check by keeping track of the IP addresses of packets arriving on the internal and bastion ports. The source and destination addresses of each packet are checked against the known port addresses to ensure they are valid for the appropriate port.

FIG. 3 also illustrates alarm controller 53, which preferably is coupled to result aggregator 24. Alarm controller 53, which could be a separate logic block or within the result aggregator, receives signals indicating when packets are being rejected, either directly from the logic performing the filtering or from result aggregator 24. As described in greater detail elsewhere herein, alarm controller 53 desirably is utilized to provide visual feedback of the system status or operation (such as whether the data protection system is under attack) via LED(s) 54 (or other light source, LCD or other type of alphanumeric or graphic display, etc.). For instance, a LCD may provide an additional mechanism for entering security configurations, such as specific protocols to allow a reference clock. Alarm controller 53 also may be coupled to an audio feedback device, such as speaker 55, which similarly may be used to provide audio feedback of the system status or operation. For example, if a packet is rejected, a first visual indication may be provided via LED(s) 54 (e.g., yellow light); if packets are being rejected in a manner or at a rate that suggests an internal computer is under attack, then a second visual indication may be provided via LED(s) 54 (e.g., a red light). Similarly, first and second tones or other audible indicators (different tones, volumes, sequences, etc.) may be provided via speaker 55 to indicate the detected condition). In preferred embodiments, such feedback, audio and/or visual, may maintain the alert state until reset by the user, such as by depressing a toggle. Thus, if the internal system has been determined to be under attack while the user is away, this fact will be made known to the user when the user returns and sees and/or hears the visual and/or audio feedback. It also should be noted that alarm controller 53 also may generate a UDP packet (indicated by the dashed line that is coupled to internal network 20) that informs the internal client computer of the attack or suspected attack, thereby providing an additional optional mechanism to inform the user of suspect activity.

FIG. 4 illustrates exemplary packet filtering functions of Level 2-type filtering in relation to the flow of packet data from internal and external networks. External PHY 12 receives packet electrical signals off the physical wire or other medium. Similarly, internal PHYs 18 and 58 receive packet electrical signals from internal network 20 or bastion network 15, respectively. Packet data comes in from one of PHYs 12, 18 or 58 to PHY controller 56. PHY controller 56 in general receives incoming data from network PHYs 12, 18 or 58, detects collisions, indicates the start and end of packet data, and forwards the packet data to other appropriate components of the data protection system (such as described herein). From PHY controller 56, data from the packet being received, along with information indicating which PHYs are active (i.e., on which PHY a packet is being received and to which PHYs the packet is being transmitted, etc.), and the packet is reshaped and transmitted in real-time via block 60 (i.e., the packet is not received into a buffer, after which it is sequentially processed to determine if the packet should be allowed to pass, etc., as in conventional firewalls). In the case of a packet received from Internet 8, the packet is received by PHY controller 56 from external PHY 12, and reshaped and transmitted in real-time to the internal PHY 18 and/or bastion PHY 58.

As will be appreciated, block 60 in essence performs the repeater functionality of passing the incoming data to the non-active PHYs after reformatting the preamble. Block 60 also preferably receives "junk" or "pass" signals from the filtering components and a collision detection signal from PHY controller 56. In preferred embodiments, a "jam" signal is propagated to each PHY upon detection of a collision. A packet is invalidated for all PHYs that belong to a network category that receives a "junk" signal. (For example, if the packet is invalidated for internal networks, then the packet is invalidated for all internal network ports.) Preferably, block 60 also receives a single output signal from result aggregator 24 for each PHY category (i.e., internal or external). As will be explained in greater detail hereinafter, result aggregator 24 generates the signals provided to block 60 depending on "junk" or "pass" signals from each filter component.

In accordance with the present invention, the packet is also simultaneously routed through a plurality of filtering steps. In the exemplary illustration of Level 2 filters in FIG. 4, the packet type is determined at step 64. At step 64, the network packet is examined to determine the enclosed Level 3 datagram type, such as ARP, RARP, IP, or IPX. This information is used to perform Level 2 filtering and to decide how to deconstruct the enclosed datagram to perform

Level 3 filtering. If an unknown packet type is received from the external network, then the packet preferably is junked if filtering is enabled. Unknown packet types received from the internal network preferably are forwarded to other hosts on the internal network and may be forwarded to the bastion port but are not forwarded to the external network.

If it is a known packet type, then it is routed through additional filtering steps based on particular packet protocols. In the illustrated embodiment, at step 66, if the packet is an Address Resolution Protocol (ARP) type packet, then it is passed. At step 68, if the packet is a Reverse Address Resolution Protocol (RARP) type packet and is from external PHY 12 and the op code is 3, then it is junked; otherwise, it is passed as indicated at step 70. As is known in the art, RARP generally is a protocol used by diskless workstations to determine their address; in accordance with preferred embodiments, RARP responses are the only RARP packets allowed to enter internal networks from external hosts. At step 72, if the packet is an Internet Protocol (IP) type packet, is from the external PHY and has been broadcast, then it is junked. (For example, broadcast packets from the external network preferably are not allowed; a broadcast packet is determined by examining the IP address or the physical layer address). Otherwise, the process proceeds to step 74. Step 74 preferably examines the IP header, which contains a protocol fragment where an application can place handling options. Certain options (such as the illustrated list) may be considered to provide internal, potentially sensitive network information, and thus packets that contain these options preferably are not allowed into the internal network. At step 74, if a handling option of 7, 68, 131, or 137 is present, then the packet is junked; if these options are not present, then the process proceeds to filter IP packet step 76 (exemplary details of step 76 are explained in greater detail hereinafter). If the packet passes the filtering rules applied in filter IP packet step 76, then the packet is passed, as indicated by step 78. If the packet does not pass the filtering rules applied in filter IP packet step 76, then the packet is junked.

As illustrated in FIG. 4, any signals indicating that the packet should be junked are provided to result aggregator 24, as indicated by line 73. The filtering results are thus routed to result aggregator 24, which records whether any of the packets were junked and thus invalidated. Result aggregator 24 provides one or more signals to the logic of block 60 at a time early enough so that a Frame Check Sequence (FCS) character may be altered to effectively invalidate the packet. Therefore, prior to complete forwarding of the packet, the filtering decision is made and the FCS character is either altered in order to ensure that it is corrupted, if the packet is to be junked, or forwarded unchanged, if the packet is to be passed. In effect, a system in accordance with the present invention acts like a hub or repeater by receiving packet nibbles (2 or 4 bits at a time) on one interface wire and by broadcasting those nibbles on other interfaces. Thus, the data protection system cannot make a decision about a packet before forwarding the nibbles on the non-receiving interfaces since this may result in an inoperable Ethernet network. If the system is enabled to filter a packet, it must still transmit data while receiving data to ensure the Ethernet network functions correctly and efficiently. The data protection system filters packets by transmitting a nibble on the non-receiving interfaces for each collected nibble on the receiving interface, but ensures that the Ethernet packet FCS character is not correct if the packet is suspect. Thus, the sending station may perceive that it successfully transmitted the packet without collision, but in fact all receiving stations will discard the corrupted packet. It should be noted that, in alternative embodiments, in lieu of or in addition to the selective alteration of a FCS or checksum-type value, the data contents of the packet also may be selectively corrupted in order to invalidate packets. In such embodiments, the packet contents are selectively altered to corrupt the packet (e.g., ensure that the checksum is not correct for the forwarded packet data or that the data is otherwise corrupted) if the packet did not pass the filtering rules.

FIG. 4 also illustrates physical switch or toggle 62, the state of which can be used to enable or control packet filtering in accordance with the present invention. The state of switch/toggle 62 is coupled to the data protection system in a manner to enable or disable packet filtering. In the illustrated example, the state of switch/toggle 62 is coupled to the logic of block 60; if, for example, packet filtering is disabled, then block 60 can receive and forward packets while disregarding the output of result aggregator 24 (alternatively, result aggregator 24 can be controlled to always indicate that the packet should not be invalidated, etc.). In other embodiments, the state of such a switch/toggle can control result aggregator 24 or all or part of the particular filtering steps. As will be appreciated in accordance with the present invention, the data protection system may be controlled and configured without requiring the implementation of complex software. The data protection system preferably utilizes toggle buttons or other physical switches to selectively enable various functions, such as Internet client applications, Internet server applications, and filtering features. The system, for example, also may contain a button for retrieving updated core logic or filtering rules from a data source. The data source for such updating of the core logic may include a wide range of forms of digital media, including but not limited to a network server, a floppy disk, hard drive, CD, ZIP disk, and DVD.Configuration, therefore, may be determined by physical interface components attached or linked to the system .

Referring to FIG. 5, additional details of preferred filter IP packet step 76 will now be described. FIG. 5 is a flow chart illustrating the packet filtering functions of the Level 3 filters first illustrated in FIG. 3. At step 81, the Level 3 filtering processes determine the IP datagram characteristics, which preferably include: datagram type (ICMP, IGMP, TCP, UDP, unknown); source and destination IP addresses; fragment offset; and fragment size. Based on the IP datagram characteristics, further filtering operations are performed. Preferred functions for Level 3 filtering will now be described in greater detail.

At step 80, if the IP datagram type is unknown, then the fail signal is set, sending a signal to the result aggregator that the packet should be invalidated. At step 82, if the IP datagram type is Internet Group Management Protocol (IGMP), then the fail signal is set, preventing IGMP packets from passing. At step 84, if the type is Internet Control Message Protocol (ICMP) and the packet is from the external PHY, then the filtering proceeds to step 88. At step 84, if the type is ICMP and the packet is not from the external PHY, then the packet is passed as indicated by step 86. At step 88, if the type is ICMP, and the packet is from the external PHY and does not contain a fragment offset of 0, then the fail signal is set, preventing fragmented ICMP packets from passing, as indicated by step 90; otherwise, the filtering proceeds to step 92. At step 92, if the type is ICMP, the packet is from the external PHY and contains a fragment offset of 0, then the packet type is further evaluated for request and exchange data. This data preferably includes one of the following ICMP message types: 5 for redirect; 8 for echo request; 10 for router solicitation; 13 for timestamp request; 15 for information request; or 17 for address mask request. Accordingly, if the packet type satisfies the criteria for step 92, then the fail signal is set as indicated by step 96. Otherwise, the packet is allowed to pass, as indicated by step 94. As will be appreciated, the ICMP filtering branch serves to keep potentially harmful ICMP packets from entering from the external network. (The listed message types represent an exemplary set of ICMP packets that may expose the internal network topology to threats or cause routing table changes.)

If IP datagram characteristics indicate that the packet is a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) packet, then the filtering proceeds to step 98. At step 98, it is determined whether the packet is a fragment 0 packet. If it is not, then the packet is allowed to pass, as indicated by step 100. This filtering process follows the convention of filtering only the first fragments, as subsequent fragments will be discarded if the first one is not allowed to pass; in other words, the data protection system ignores all but the first packet of a TCP or UDP datagram. At step 104, if the packet is TCP or UDP and is a first fragment packet, then it is determined whether a proper protocol header is included in the fragment; if it is not, then the fail signal is set as indicated by step 102 (in the illustrated embodiment all TCP and UDP packets that have improper headers are junked). If the packet is TCP or UDP, is a first fragment, and a proper protocol header is included in the packet, then the filtering proceeds to step 106 (further exemplary details of which will be described in connection with FIG. 6).

FIG. 6 is a flow chart that illustrates a preferred example of how TCP and UDP packets are evaluated in parallel in accordance with the present invention (see, e.g., the multiple rules engines and related discussion in connection with FIG. 2 and the Level 4 filters of FIG. 3). As is known, TCP and UDP are host-to-host protocols located in the Transport Layer of the protocol stack. FIG. 6 illustrates how packet data 108 is unbundled and decoded for packet characteristics at step 110 (e.g., IP addresses, ports, flags, etc.) as well as for packet type and PHY activity at 112 (i.e., whether it is an internally generated packet or an externally generated one). In the preferred embodiments, the packets are evaluated in parallel according to the following rules.

As indicated at step 114, if the internal port number is 68 and the external port number is 67, then the packet is passed, regardless of whether it originated on the internal network or the external network. As indicated at step 116, if the packet type is TCP, the server-mode is enabled (such as may be controlled by a toggle or other physical switch), the external PHY is active, and the internal port number is 80, then the packet is passed to the internal network(s). (The server mode is explained in greater detail in connection with FIG. 7 below). As indicated at step 118, if the packet type is TCP and either the Acknowledge ("ACK") bit or Final ("FIN") bit is set, then the packet is passed, regardless of whether it originated on the internal network or the external network. As indicated at step 120, if the packet type is TCP and an internal PHY is active, then the packet is passed to the external network. As indicated at step 122, if the packet type is UDP, an internal PHY is active, and the external port number is 53, then the packet is passed to the external network and the communication state (e.g., source and destination port numbers) is stored as indicated by comm or communication state store 124. As indicated at step 126, if the packet type is UDP, the external PHY is active and the external port number is 53, then the packet is passed to the internal network(s) if there is a match in the communication state. As indicated at step 128, if the packet type is TCP, an internal PHY is active, the external port number is 21, the Synchronize Sequence Numbers ("SYN") bit is not set but the ACK bit is set, and the packet is a PORT command, then the packet is passed to the external network and the client (internal network) active port is determined and the communication state is stored. As indicated at step 130, if the packet type is TCP, the external PHY is active, the external port number is 20, and the SYN bit is set but the ACK bit is not set, then the packet is passed to the internal network(s) if there is a communication state match. As indicated at step 132, if all checks have been completed, then a complete signal is set, and signals indicative of whether the packet passes to internal or external network(s) as previously described are bitwise logically ORed to generate pass internal and pass external signals, as illustrated.

In preferred embodiments, if the completion signal is not generated by the time that the packet has been completely received, then the packet is junked. It should be noted that the use of such a completion signal and packet junking can be extended to the diagrams and description, etc. of other figures, such as FIGS. 2, 3, 4, 5, 7 and 8. If the filtering process has not been completed by the time that the packet has been completely received, then the packet is preferably junked.

Referring now to FIG. 7, Level 4 filtering in accordance with the present invention will be further described. The embodiment of FIG. 7 is a table-based filter, which uses an approach similar to that described in connection with FIG. 2. This approach preferably utilizes a programmable logic device (PLD) that includes low latency, high-speed ROM and RAM blocks.

As previously described, Level 4 filtering is based on TCP and UDP packet characteristics, the determination of which is illustrated in FIG. 7 by block 133. TCP and UDP characteristics, as noted elsewhere herein, may include not only source and destination port numbers, but also the state of the SYN, ACK, FIN and/or RESET flags in the case of TCP packets. The TCP/UDP characteristics are determined by the TCP/UDP header information. The TCP/UDP characteristics and active PHY information are used in the generation of a lookup code, which in the embodiment of FIG. 7 is coupled to rules dispatcher 134. Rules dispatcher 134 uses a lookup code to determine the filtering rules to be applied to a packet and then places the identifiers of the rules to be run in queues 138-1 to 138-N for each of the rules engines 140-1 to 140-N. Mapping table 136 is coupled to and receives address data from rules dispatcher 134.

Mapping table 136 preferably is a ROM block that identifies the rules associated with each lookup code and the rules engine for which each rule is to be dispatched. The mapping data for the rules and rules engines are returned to rules dispatcher 134.

The identifiers of the rules to be run are dispatched by rules dispatcher 134 to the appropriate queues 138-1 to 138-N, which are preferably FIFO-type structures that hold the rule identifiers for corresponding rules engines 140-1 to 140-N. Queues 138-1 to 138-N not only enable rules dispatcher 134 to assign rules at maximum speed, but also allow each rules engine to retrieve rules as each one is evaluated. The rules engines 140-1 to 140-N are a plurality of filtering engines/logic that use a rule table to read a definition specifying whether a rule applies to a packet and whether the packet passes or fails the rule test. Rules tables 142-1 to 142-N preferably are ROM blocks that contain a definition of a set of filtering rules that are controllably run by the rules engines 140-1 to 140-N. Rules tables 142-1 to 142-N may contain different rules as may be appropriate to provide all of the rules necessary to adequately filter packets within the timing constraints imposed by the real time filtering of the present invention, and the speed of the hardware used to implement the data protection system.

In addition, as illustrated in FIG. 7, rules engines 140-1 to 140-N may receive as inputs signals indicative of a stored communication state, IP datagram characteristics, or physical switch/toggle states. As indicated by block 148, toggles may be utilized for a variety of features, such as enabling web client, web servers or other user-defined features. With at least some of the executed rules based on the stored communication state, stateful rules are implemented with the illustrated embodiment. A communication state table or cache is provided. A cache of communication state information between different hosts provides a set of bits that represent rule defined state information. For example, source and destination port information may be stored in the cache and used for state-dependent filtering.

In the illustrated embodiment, communication state information from rules engines 140-1 to 140-N may be provided to result aggregator 144, which in turn may store the communication state information to the communication state cache or storage area. Result signals, representing pass or fail of the packet based on the applied rules, also are provided to result aggregator 144. Result aggregator 144 combines the pass/fail results signals and provides a pass or junk signal or signals, which may be provided to the repeater core or to another result aggregator.

FIG. 8 illustrates an alternative preferred embodiment, in which the Level 4 filtering is implemented with a register-based filtering methodology. As with the Level 4 filtering of FIG. 7, both stateful filters 154 and non-stateful filters 153 may be implemented. As with the embodiment of FIG. 7, Level 4 filtering requires that TCP and UDP packet characteristics be determined, as illustrated by box 150. In addition to the Level 3 packet characteristics, Level 4 filters in accordance with this embodiment also require the source and destination port numbers and the TCP header values for the SYN, RST, FIN flags and the ACK value. This information preferably is used by both non-stateful and stateful filters 153 and 154. The implementation of the non-stateful filters is executed with a state machine or other logic preferably in the PLD that compares characteristics to the allowed non-stateful rules and makes a judgement as to whether the packet should be passed or failed. The non-stateful rules engine/logic uses a set of static rules to decide if a packet is allowed to pass through the firewall. These rules preferably are specified using a combination of control inputs, active PHY, and network packet characteristics.

Stateful filters are implemented to handle communication channel interactions that span multiple transmissions between hosts. The interactions typically occur at the Application Layer of the protocol stack, where examples may include FTP, RealAudio, and DHCP. These interactions may also take place at lower levels in the protocol stack, such as ARP and ICMP request/response.

In this embodiment, stateful filters 154 use protocol front-end and protocol back-end logic, along with a plurality of state registers to implement state-dependent filters. Each protocol that requires stateful packet filtering preferably has protocol handlers in the form of front-end and back-end logic, which decide when to issue a pass signal for a packet or store the identifying characteristics of a bitstream for later reference. Front-end logic 160-1 to 160-N monitors the network traffic to identify when the current communication state needs to be stored, deleted or updated. Front-end logic 160-1 to 160-N informs a corresponding back-end logic 158-1 to 158-N that a register will be allocated for storage for a bitstream. All store and delete state register requests are sent to back-end logic 158-1 to 158-N so it may update its internal information. Register controller 155 controls the actual selection of registers in state registers 156 and informs the corresponding back-end logic 158-1 to 158-N. Back-end logic 158-1 to 158-N monitors which state registers are dedicated to its protocol and issues a pass signal for packets that match an existing bitstream, as indicated by the appropriate packet characteristics and a matching state register. It should be noted that in alternate embodiments, different organizations of the functions of the programmable logic may be implemented in accordance with the present invention, incorporating various types of protocol handlers and state registers, as may be necessary.

Register controller 155 consolidates multiple store and clear signals from the various front-end logic 160-1 to 160-N and directs them to the appropriate registers in state registers 156. Register controller 155 also informs the various back-end logic 158-1 to 158-N which registers of state registers 156 are to be used for storage. The registers of state registers 156, under control of register controller 155, store the communication state of a bitstream; for example, a particular register records information about the two communication ends of the bitstream and also monitors each network packet to see if it matches the stored end-point characteristics. State registers 156 then sets a signal when its state matches the current packet characteristics. A "garbage collection" function also is implemented (as further illustrated in FIG. 13 below) to help free up state registers when the protocol information during the three-way handshake is not accessed within specific time frames.

As is known in the art, many protocols provide a way of identifying the end of a communication session. Accordingly, in preferred embodiments the data protection system detects when a stateful stream ends and frees up the associated state registers. Since clients and servers do not always cleanly terminate a communication session, the system preferably implements session time-outs to free state registers after a period of bitstream activity and to prevent indefinite state register exhaustion. If the network experiences a high rate of bitstreams requiring stateful inspections, the system's resources, which are allocated to tracking application data, can become exhausted. In this case, the system preferably resorts to allowing network traffic based on a set of static rules to pass through the non-stateful rules designed specifically for each protocol. This stateful to non-stateful transition is called "stateful relaxation." To maintain maximum security, a protocol handler that cannot gain access to an open state register will free up all of its state registers to help prevent other protocol handlers from entering into a relaxation state. The system will then wait for a state register to open, start a timer, and record protocol communication data in the state registers, while relying on the static rules. When the timer expires, the state filter will cease relying upon the static rules and approve packets solely on state register information.

FIG. 8 also illustrates toggle 152, which, in the additional illustrated example, selectively enables FTP (File Transfer Protocol) communications based on the switch state. Protocol back-end logic 158-1 to 158-N, as appropriate, utilize such toggle state information to selectively generate the pass/fail signals for the applicable protocols. For example, when the toggle switch is enabled, which is the default mode in most FTP client applications, it may send a signal to the internal FTP server to open a TCP connection to the client. Front-end logic 160-1 monitors the network traffic for data from the internal network, PORT command, source port number (greater than 1024) and destination port number (equal to 21).. When this information is matched, front-end logic 160-1 requests state register controller 155 to store both the PORT command IP address and the port number as the destination end and the destination IP address, as well as store port 20 as the source end of a future communication packet. (In other embodiments, additional checks may be conducted to ensure the active connection IP address is the same as the current source IP address.) When back-end logic 158-1 recognizes the storage request, it waits for the allocated state register in state registers 156 to be sent by register controller 155. For example, when the state register number is set as register #1, then it records that register #1 is dedicated to allowing active FTP connections through the data protection system. Back-end logic 158-1 then waits for register #1 to signify that the current packet matches its stored state. When back-end logic 158-1 recognizes that the three-way TCP handshake has been completed for the new connection, it will notify front-end logic 160-1 to delete the state register. If the state register is junked, then back-end logic 158-1 records that register #1 is no longer dedicated to active FTP connections, allowing register controller 155 to allocate that register to a different protocol or network connection in the future.

FIG. 9 illustrates a preferred physical implementation of one embodiment of the present invention. In this embodiment, one external network connection and one internal network connection are provided. It will be appreciated that the components of FIG. 9 can be altered to implement, for example, bastion network connections and multiple internal network connections, etc.

The Internet connection, for example, via a cable modem, DSL router or other network interface, preferably is coupled with a physical cable to connector 168, which may be an RJ-45 connector. The signals received via connector 168 are coupled to and from PHY 170, which provides the physical interface for the data signals received from, or coupled to, the external network. Signals are coupled between PHY 170 and PLD 162, and signals are coupled between PLD 162 and PHY 172, which couples signals between connector 174 (which again may be an RJ-45 connector). The connection to the internal network may be made through connector 174.

In the preferred embodiment, PLD 162 implements the various levels of filtering as previously described. PLD 162 provides logic/hardware based, parallel filtering rules logic/engines, which make a decision about whether the packet should be allowed to pass or fail prior to the time that the packet is passed on by the repeater core portion of PLD 162 (as described elsewhere herein). The logic of PLD 162 to implement the filtering rules is programmed/loaded by controller 164, which may be a RISC CPU such as a MIPS, ARM, SuperH-type RISC microprocessor or the like. The PLD code preferably is stored in memory 166, which preferably is a re-programmable, non-volatile memory, such as FLASH or EEPROM. In this manner, the PLD code may be updated by reprogramming memory 166, and the updated PLD code may then be programmed/loaded in to PLD 162 under control of processor 164.

FIG. 9 also illustrates the use of LEDs 177, 178 and 179 to provide visual feedback of the data protection system status. In accordance with the present invention, the use of such displays or light sources may be used to convey various types of information to the user. For example, LEDs 177 and 179 may be provided to indicate that PHYs 170 and 172 are detecting an active network connection (and thus provide an indication that the network connections are present and functioning properly). LED 178 preferably provides alarm type information. For example, LED 178 may be provided in the form of a multi-color LED, which may provide a first colored light (e.g., yellow) if the data protection system has rejected one or more packets (thereby indicating that the system may be detecting an attack), and which may provide a second colored light (e.g., red) if the data protection system is continually rejecting packets or rejecting packets at a high rate (thereby indicating that the system is likely under attack). Such visual indicators, which may be coupled with audio feedback as described elsewhere herein, serve to inform the user that the user's computer or network may be under attack, thereby enabling the user to take further action, such as disconnecting from the network.

It should be noted that such visual feedback may be implemented in a variety of forms. In addition to multi-color or multiple LEDs or other lights sources or displays, a single LED could be provided, with the LED blinking at a rate that indicates the level of severity as predicted by the data protection system. For example, if no packets have been rejected, then the LED may be in an off or safe (e.g., green) state. If packets have been rejected but not on a continual or high rate basis, then the LED (e.g., red) may be controlled to blink on and off at a first, preferably lower speed rate. If packets are being rejected on a continual or high rate basis (or otherwise in a manner that that system believes is suspect), then the LED may be controlled to blink on and off at a second, preferably higher speed rate. Thus, the LED blink rate desirably may be controlled to blink at a rate that corresponds to the level of severity of the security threat that is determined by the data protection system. Optionally coupled with audio feedback, such visual indicators may provide the user with alarm and status information in a simple and intuitive manner.

As further illustrated in the preferred embodiments of FIG. 9, a variety of physical switches or toggles 176, 180, 181 and 182 may be coupled to PLD 162 or controller 164. As illustrated by update button 176, toggles may be used to control the updating of the PLD code (for instance, to reconfigure or update the system, providing updated filtering algorithms). As illustrated by buttons 180 and 181, toggles may be used to selectively activate/deactivate filtering steps depending on whether a protected computer is enabled to operate in either a server mode or client mode (the state of such toggles preferably being used to control filtering decisions made within the filtering logic). As illustrated by reset button 182, toggles may also be used to control the reset of the data protection system (for example, to cause the PLD code to be re-loaded, as when the system enters an inoperable state caused by power supply irregularities or other unusual circumstances). The use of such physical switches/toggles allows the data protection system to be controlled in a straightforward manner, simplifying the user operability of embodiments of the present invention.

With reference to FIG. 9, additional details of preferred update program and protocols will now be described. The data protection system may be controlled to operate in an update mode by pressing update button or toggle 176, which preferably is provided on an external case (further described in FIG. 10 below). In accordance with preferred embodiments, during the interval when the update button is pressed by the user and the update either completes or is canceled by the user, the data protection system will not forward any packets (i.e., filtering is not active, so packet transmission is blocked). The user may then run an update program (which may be a browser-based or stand-alone application) from an internal host computer.

In the illustrated embodiment, it is assumed that the user previously downloaded a system update or is downloading an update through a browser. The update program preferably breaks the update into 1K size packets and forwards them, using a limited broadcast destination address (for example, 255.255.255.255). The source and destination ports are set to a predetermined value, such as 1 (1-4 are currently unassigned according to RFC 1010), and an IP option is set in the IP header. The program data preferably is preceded by the system update header that has the following structure in the illustrated embodiment: ID (1)/count (1)/bit length (2). The numbers in parentheses represent the field size in bytes. The ID for the entire transaction remains unchanged, except for the count field increments for each packet. In a preferred embodiment, the data protection system may receive the packets in order and perform several checks, such as ensuring the ID and count fields are correct, verifying the UDP checksum, and storing the configuration data in non-volatile memory. Preferably, these checks may be controlled by controller 164. Thereafter, the updated PLD code may be loaded into the PLD, with the filtering operations being based on this updated code.

As a result of the parallel filter rules evaluation as previously described, packets do not need to be buffered, except, for example, to create octets that facilitate determining protocol elements. (As is known, data needs to be combined into 8-bit, 16-bit, or 32-bit words because header and packet data often exist in these sizes or straddle a 4-bit nibble boundary.) Instead of buffering each packet, the data protection system generates another distinct data packet or chunk. This process of packet generation occurs while a plurality of filtering rules are applied in real time and in parallel, producing improved data protection systems and methods.

FIG. 10 illustrates a preferred embodiment of an exemplary design of an external case of a data protection system in accordance with the present invention (it being noted that the particular switches, lights, etc., and their physical arrangements being exemplary). For example, external case 184 may be a molded plastic box in the shape of a "U" or folded tube as illustrated. The exemplary features of this external case may include ports, buttons (or toggle switches), LEDs, a clock, a removable logo disk, and a power supply connector. Home (internal) port 186, Internet (external) port 188, and power supply connector 190 are preferably located on the same side of external case 184 with power supply connector 190 set between the two ports. Home port 186 connects to the internal network via cable 192; Internet port 188 connects to the external network via cable 194. Power supply connector 190 is coupled to an external DC power supply via cable 193. The PHY of each port preferably is coupled to a link LED, such as previously described: home port 186 is coupled to internal link LED 196; and Internet port 188 is coupled to external link LED 198. The link LEDs are thus coupled to the internal and external PHYs, respectively, and serve to indicate whether the PHYs have detected a network connection.

In the preferred embodiment, on the internal network side of the U-shaped case, server mode button 200 is provided to allow the user to selectively enable filtering depending on whether the internal computer is allowed to operate in a server mode (thus, the state of server mode button 200 may be used to selectively control filtering decisions based on whether internal computers will be operating in a server mode, etc.). Server mode button 200 preferably includes server mode LED 202. When illuminated (e.g., green), server mode LED 202 indicates that the internal computers are enabled to operate in a server mode and the filtering decisions will be controlled accordingly. Server mode button 200 and server mode LED 202 are coupled to PLD 162, as described in FIG. 9. In the illustrated embodiment, parallel to server mode button 200 on the external side of the case is alert button 204, which contains alert LED 206. Alert LED 206 is coupled to alarm controller 53, which preferably is implemented as a part of PLD 162 (as illustrated in FIGS. 3 and 9, respectively). Alert LED 206 may contain a single or multi-colored LED, which, when illuminated, indicates the data protection system is under attack and is rejecting suspect packets. The data protection system preferably registers the frequency of attacks and sends signals to alert LED 206 based on such information. In a preferred embodiment, alert LED 206 may contain a LED (e.g., red), which remains consistently illuminated during irregular attacks or blinks at regular intervals under heavy attack. In another preferred embodiment, alert LED 206 may contain a multi-colored LED, which similarly indicates when the system is under attack and is rejecting packets. However, with a multi-colored LED, the increase in frequency or intervals of attacks may be indicated by a change in color: for example, green (indicating no registered attacks by suspect packets) to yellow (indicating a few irregular attacks) to red (indicating more frequent attacks) to blinking red (indicating a heavy attack). The alert alarm may be reset by depresseing alert button 204.

In a preferred embodiment, speaker 55 or some form of audio transducer may be coupled to alarm controller 53 to also indicate the presence or severity of attacks (as described in connection with FIG. 3). For example, when the data protection system is under heavy attack and alert LED 206 is blinking (e.g., red), an alarm signal may be transmitted to speaker 55 to emit audio information to indicate a suspected severe attack or emergency. Alarm-type information may also be coupled to the internal network (such as via a UDP packet, as described elsewhere herein), and thus transmit alarm information over the network to a software interface on the desktop. In other embodiments of the data protection system, an array of different features, including buttons, LEDs, alarms, and graphical user interfaces, may be utilized to indicate the class, frequency and severity of attacks on the system.

Adjacent to alert button 204 on the external network side of the case preferably is protection button 208, which is coupled to protection-on LED 212 and protection-off LED 214. When protection button 208 is set in the "on" position, protection-on LED 212 preferably illuminates red and the filtering system is enabled; when protection button 208 is set in the "off" position, protection-off LED 214 preferably illuminates yellow and the filtering system is disabled. As will be appreciated, the particular colors utilized are exemplary.

Still referring to FIG. 10, power LED 210 is coupled in a manner to indicate power is being provided via power supply connector 190. When power LED 210 is illuminated (e.g., green), it indicates the power supply is providing power to the data protection system. It should be noted that in the illustrated embodiment, the present invention does not require an on/off switch for the power supply because the system is designed to be enabled once a DC power supply is provided. As previously described, reset button 182 is coupled to controller 164 and may be used to initiate loading or re-loading of the PLD code.

Adjacent to reset button 182 is update button 176, which is coupled to update-enabled LED 218 and update-disabled LED 220, as well as PLD 162 (as illustrated in FIG. 9). As previously described, an update program preferably is utilized to update the logic programming and rules tables. Preferably, after pressing update button 176, the data protection system is automatically restarted, causing the new PLD code to load. The load version bit preferably will be set in the flash configuration header, which causes the system to load using the new program file. In a preferred embodiment, update-enabled LED 218 will illuminate in green to indicate the data protection system is ready to receive the new updated programming. After the update begins, the system may continually flash update-enabled LED 218 until the successful completion of the update; LED 218 is extinguished upon successful completion of this process. However, if an update is incomplete and fails to occur, update-failed LED 220 may illuminate in red and blink. The user extinguishes LED 220 by pressing the update button a second time. If possible, the data protection system may generate a UDP packet to inform the internal client of the reason for the failure. As an additional example, if the system contains an LCD, it may display an error code. The data protection system will continue to filter packets after update-failure LED 220 is extinguished. LED 216 is preferably provided to be illuminated when the system is operating and filtering packets in the manner described.In addition to the various toggles in a preferred embodiment of the present invention, additional types of components may be used to enter filtering criteria and/or selectively enable or control the filtering, such as a LCD display coupled with input buttons, a touch screen, an audio input for speech recognition, and/or a clock. Thus, filtering decisions may be made based on such switch inputs, audio commands, time of day or date, etc.

As further illustrated in FIG. 10, a removable logo disk 222 may be located on a preferred embodiment of the case. This removable disk may include a company logo, registered trademark, and/or other copyrighted material that may be valuable for branding and marketing the data protection system under a separate wholesaler. The disk is thus removable and replaceable for a variety of branding purposes.

In an alternate embodiment, security levels switch 223 may be implemented to prevent stateful relaxation, in which a stateful to non-stateful transition may occur during state register exhaustion. As illustrated in FIG. 8, security levels switch 223 may preferably include a variety of features that prevent stateful relaxation, such as timers, protocol-specific filters, and other rules-based filters. For example, switch 223 may be configured for three positions: one which allows FTP protocols, but does not allow DNS protocols; another which allows DNS protocols, but does not allow FTP; and a third which may serve as an emergency back-up feature and block all network traffic.

In other embodiments, different designs may be used in accordance with the present invention, incorporating various buttons, switches, LEDs, ports, cables, slots, connectors, plug-ins, speakers, and other audio transducers, which in turn may be embodied in a variety of external case shapes, as may be necessary. As will be appreciated, the filtering criteria may be dependent upon physical switch position, packet characteristics, clock time, and/or user-specified criteria, all of which may be entered through one or more physical input device(s). Such a physical input device, for example, may be comprised of one or more switches (such as a toggle switch, button switch, or multi-state switch), an audio input device, or display input device. The user-specified criteria may be transferred from the configuration software to the system using a network protocol, infrared port, or cable attachment.

FIGS. 11and 12 are flow diagrams illustrating examples of "SYN flood" protection in accordance with preferred embodiments of the present invention. Such SYN flood protection is optionally provided as an additional computer protection mechanism in accordance with certain preferred embodiments.

As is known in the art, SYN flood is a common type of "Denial of Service" attack, in which a target host is flooded with TCP connection requests. In the process of exchanging data in a three-way handshake, source addresses and source TCP ports of various connection request packets are random or missing. In a three-way handshake, the system registers a request from an IP address, then sends a response to that address based on its source, and waits for the reply from that address.

As illustrated in FIG. 11, the data protection system waits for a packet from external PHY 14 (as illustrated in FIG. 2) at step 224. When the system receives a packet from the external PHY, it compares the IP address and ports to the flood list entries at step 226, then proceeds to step 228. At step 228, the system determines whether the packet type is TCP, the ACK bit is set, and the packet matches an entry in the flood list. If these criteria are met, then the system proceeds to step 230, where the packet is removed from the flood list. If the packet is removed from the flood list, then the system returns to step 224 and waits for the next packet from the external PHY. Otherwise, if the criteria at step 228 are not met, then the system proceeds to step 232, where the system determines whether the packet type is TCP, the SYN bit is set and the ACK bit is not set. If the criteria at step 232 are met, then the system proceeds to step 234; otherwise, the system returns to step 224. At step 234, the system determines if the flood list is full and if the client has reached the maximum connection requests. If the flood list is not full, then the system returns to step 224 to wait for more packets from the external PHY. However, if the flood list is full at step 234, then the system proceeds to step 236, where the packet is junked and the system returns to step 224.

As illustrated in FIG. 12, the data protection system also waits for a packet from internal PHY 18 (as illustrated in FIG. 2) at step 238. When the system receives a packet from the internal PHY, it accesses the flood list location and writes the bits into the list, swapping ACK bits as well as MAC, IP and port addresses. The system then proceeds to step 242, where it determines if the packet type is TCP and the SYN and ACK bits are set. If the criteria at step 242 are met, then the system proceeds to step 244; if not, then the system returns to step 238 and waits for another packet from the internal PHY. At step 244, the SYN flag is unset and number 1 is added to the new ACK number. The system then proceeds to step 246, where it determines if the flood list is full. If the flood list at step 246 is full, then the Reset flag is set, the checksums for TCP, IP and Ethernet protocols are recalculated, and the Reset packet is transmitted. The system then returns to step 238. However, if the flood list at step 246 is not full, then the system proceeds to step 248, where the checksums for TCP, IP and Ethernet protocols are recalculated and the ACK packet is transmitted. The system then proceeds to step 252, where the recalculated packet is added to the flood list and the system returns to step 238, where it waits for another packet from the internal network.

In accordance with the present invention, SYN flood protection as described does not require either an IP or MAC address. The data protection system uses the destination MAC address as the source Ethernet address when framing the response packet that completes the TCP three-way handshake. In all cases, when forming the new packet, the source and destination header information is swapped, so that the source IP address and port become the destination IP address and port. It should be appreciated that SYN flood protection, as preferably implemented by the system, does not buffer the incoming packet, but builds the TCP response packet in real-time. The new TCP packet is placed in a queue for transmission at the earliest time possible based on the rules dictated by the link level protocol..

As illustrated in FIG. 13, in order to keep the flood lists from filling up with stale entries, the data protection system must free up state registers when the protocol information is not accessed within specific time frames, such as when a three-way handshake is initiated by a client, but the transaction is not closed. After the system receives a packet, it for one second at step 254, then proceeds to step 256, where the packet is checked against each flood list entry and passed to step 258. At step 258, the system checks for stale entries (or garbage collection) in the flood lists and proceeds to step 260, where it determines if time has expired. If time has expired at step 260, then the packet proceeds to step 262; if not, then the system returns to step 256 to check each flood entry list again. At step 262, the system unsets the ACK bit and sets the Reset flag, adds 1 to the sequence number, recalculating the checksums, and then recalculates the checksums for TCP, IP, and Ethernet protocols. The system proceeds to step 264, where the Reset packet is transmitted; it then proceeds to step 266 and removes the packet from the flood list. The system then proceeds to step 256. It should be noted that if time expires for the request, then the system sends the Reset flag, terminating the connection.

Although the invention has been described in conjunction with specific preferred and other embodiments, it is evident that many substitutions, alternatives and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, the invention is intended to embrace all of the alternatives and variations that fall within the spirit and scope of the appended claims. For example, it should be understood that, in accordance with the various alternative embodiments described herein, various systems, and uses and methods based on such systems, may be obtained. The various refinements and alternative and additional features also described may be combined to provide additional advantageous combinations and the like in accordance with the present invention. As will also be understood by those skilled in the art based on the foregoing description, various aspects of the preferred embodiments may be used in various subcombinations to achieve at least certain of the benefits and attributes described herein, and such subcombinations also are within the scope of the present invention. All such refinements, enhancements and further uses of the present invention are within the scope of the present invention.

What is claimed is:

1. A method for communicating data between an external computing system and an internal computing system over a packet-based network, comprising the steps of:

receiving a communication packet from the external computing system over the network, the packet having at least a first portion and an end portion, and transmitting the packet to the internal computing system;

in parallel with the step of receiving and transmitting the packet, determining characteristics of the packet from the first portion;

in parallel with the step of receiving and transmitting the packet, performing a plurality of checks on the packet, wherein at least certain of the plurality of checks are performing in parallel with other of the plurality of checks;

in parallel with the step of receiving and transmitting the packet, determining if the packet should be a valid packet or an invalid packet based on the plurality of checks; and

after receiving the end portion of the packet, selectively altering the end portion of the packet based on whether the packet has been determined to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

2. The method of claim 1, wherein the packet is analyzed in real time to determine if the packet should be valid or invalid while the packet is being concurrently transmitted to the internal computing system.

3. The method of claim 1, wherein the packet is analyzed to determine if the packet is valid without the packet having been completely received and buffered.

4. The method of claim 1, wherein the packet is determined to be an invalid packet if it is determined that the packet contains a virus, is unauthorized or presents a risk of harm to the internal computing system.

5. The method of claim 1, wherein the plurality of checks are at least in part selectively performed based on a state of a physical switch.

6. The method of claim 5, wherein the physical switch comprises one or more user-controlled switches, wherein the plurality of checks are selectively performed based on a user-defined state of the one or more user-controlled switches.

7. The method of claim 6, wherein the one or more user-controlled switches comprise at least one user-controlled switch that controls a configuration or reconfiguration of a circuit that performs the plurality of checks.

8. The method of claim 7, wherein the configuration or reconfiguration of the circuit that performs the plurality of checks is performed without requiring user entry of configuration commands via software running on the internal computing system.

9. The method of claim 7, wherein the circuit that performs the plurality of checks is configured or reconfigured based on commands from the internal computing system and based on a state of the at least one user-controlled switch.

10. The method of claim 5, wherein at least a subset of the plurality of checks are selectively enabled or disabled based on the user-defined state of the user-controlled switches.

11. The method of claim 1, wherein the plurality of checks are performed with a programmable logic device, wherein logic within the programmable logic device is selectively programmed to perform the plurality of checks in parallel with the receiving and transmitting of the packet.

12. The method of claim 11, wherein a first physical interface circuit receives the packet from the network, wherein the packet is coupled to the programmable logic device, wherein the packet is coupled from the programmable logic device to a second physical interface circuit for transmission to the internal computing system.

13. The method of claim 12, wherein the programmable logic device performs the plurality of checks while the packet is being coupled from the first physical interface to the second physical interface.

14. The method of claim 1, wherein the plurality of checks are selectively performed based on a communication state between the external computing system and the internal computing system.

15. The method of claim 14, wherein the communication state comprises one or more network addresses and/or one or more port numbers.

16. The method of claim 16, wherein the network address comprises an IP address for the external computing system and/or the internal computing system.

17. The method of claim 1, further comprising the step of providing visual or audio feedback with one or more visual or audio feedback devices, wherein the one or more visual or

audio feedback devices selectively provide visual or audio feedback of the operation or status of a packet filter process.

18. The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a system performing the packet filter process is powered or operational.

19. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process is subjecting a packet to filtering criteria.

20. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system performing the packet filter process has rejected one or more packets.

21. The method of claim 17, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the internal computing system is suspected to be under attack.

22. The method of claim 21, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

23. The method of claim 18, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system performing the packet filter process until the one or more visual or audio feedback devices are reset by a user.

24. The method of claim 23, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

25. The method of claim 18, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system performing the packet filter process.

26. The method of claim 25, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system performing the packet filter process.

27. The method of claim 25, wherein the light source is controlled to selectively blink depending on the operation or status of the system performing the packet filter process.

28. The method of claim 27, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on the internal computing system.

29. The method of claim 25, wherein the at least one light source comprises an LED.

30. The method of claim 17, wherein the one or more visual or audio feedback devices comprise a speaker.

31. A system for filtering packets of data between at least an external network and an internal network, comprising:

a first interface circuit for coupling data to and from the external network;

a second interface circuit for coupling data to and from the internal network;

a programmable logic device coupled between the first interface circuit and the second interface circuit;

wherein, as a packet is being received and transmitted between the first and second interface circuits, the packet is simultaneously subjected to a plurality of filtering criteria by the programmable logic device, wherein an end portion of the packet is selectively altered by the programmable logic device based on the filtering criteria.

32. The system of claim 31, wherein the filtering criteria determine whether the packet is to be a valid packet or an invalid packet, wherein the packet is selectively altered to be invalid if it was determined that the packet should be an invalid packet.

33. The system of claim 31, wherein the programmable logic circuit includes at least first logic for determining characteristics of the packet being received and transmitted between the first and second interface circuits and at least a filter portion that subjects the packet to the plurality of filtering criteria while the packet is being received and transmitted between the first and second interface circuits.

34. The system of claim 33, wherein the filter portion includes at least a stateful filter portion and a non-stateful filter portion.

35. The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion and the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

36. The system of claim 34, wherein the stateful filter portion subjects the packet to one or more stateful filtering criterion while the non-stateful filter portion subjects the packet to one or more non-stateful filtering criterion.

37. The system of claim 34, wherein a result aggregator logic receives one or more signals from the stateful filter portion and the non-stateful filter portion, wherein based on the received signals the result aggregator logic controls whether the packet is selectively altered to be invalid.

38. The system of claim 37, wherein the result aggregator logic receives a completion signal that indicates whether the stateful and/or non-stateful filter portions have subjected the packet to all of the filtering criteria.

39. The system of claim 38, wherein, if the completion signal is not received by the result aggregator logic by a time when the end portion of the packet has been received, then the packet is selectively altered by the programmable logic device to be invalid.

40. The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in parallel with the packet being received and transmitted between the first and second interface circuits, wherein a decision is made whether to selectively alter the packet to be invalid by a time when the end portion of the packet has been received.

41. The system of claim 31, wherein the packet is subjected to the plurality of filtering criteria in real time with the packet being received and transmitted between the first and second interface circuits.

42. The system of claim 31, further comprising one or more physical switches, wherein the packet is selectively subjected to the filtering criteria based on the state of the one or more physical switches.

43. The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria.

44. The system of claim 42, wherein the state of the one or more physical switches selectively enable or disable a predetermined portion of the filtering criteria based on whether a computer coupled to the internal network is controlled to operate in a client mode or a sever mode.

45. The system of claim 42, wherein the state of the one or more physical switches selectively controls a configuration or reconfiguration operation of the programmable logic device.

46. The system of claim 42, wherein the state of the one or more physical switches selectively controls a reset operation of the programmable logic device.

47. The system of claim 31, further comprising one or more visual or audio feedback devices, wherein the one or more visual or audio feedback devices selectively provide visual or audio feedback of the operation or status of the system.

48. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is powered or operational.

49. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system is subjecting a packet to the filtering criteria.

50. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that the system has rejected one or more packets.

51. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback that a computer coupled to the internal network is suspected to be under attack.

52. The system of claim 51, wherein the one or more visual or audio feedback devices provide visual or audio feedback of an estimated severity of the attack.

53. The system of claim 47, wherein the one or more visual or audio feedback devices provide visual or audio feedback of a state of the system until the one or more visual or audio feedback devices are reset by a user.

54. The system of claim 53, wherein the one or more visual or audio feedback devices are reset by the state of a physical switch.

55. The system of claim 47, wherein the one or more visual or audio feedback devices comprise at least one light source, wherein the light source is selectively controlled to provide information indicative of the operation or status of the system.

56. The system of claim 55, wherein the light source is controlled to have a first color or a second color depending on the operation or status of the system.

57. The system of claim 55, wherein the light source is controlled to selectively blink depending on the operation or status of the system.

58. The system of claim 57, wherein the light source is controlled to selectively blink at a rate that is indicative of a severity level of a suspected attack on a computer coupled to the internal network.

59. The system of claim 55, wherein the at least one light source comprises an LED.

60. The system of claim 47, wherein the one or more visual or audio feedback devices comprise a speaker.

61. The system of claim 36, wherein the stateful filtering criteria are dependent upon physical switch position, packet characteristics, clock time and/or user-specified criteria.

62. The system of claim 61, wherein the user-specified criteria are entered via a physical input device.

63. The system of claim 62, wherein the physical input device comprises one or more switches, an audio input device, or display input device.

64. The system of claim 61, wherein the user specified criteria are entered via a configuration software.

65. The system of claim 64, wherein the user specified criteria are transferred from the configuration software to the system using a network protocol, infrared port or cable attachment.

66. The system of claim 63, wherein the one or more switches comprise a toggle switch, button switch or multi-state switch.

Abstract

Methods and systems for firewall/data protection that filters data packets in real time and without packet buffering are disclosed. A data packet filtering hub, which may be implemented as part of a switch or router, receives a packet on one link, reshapes the electrical signal, and transmits it to one or more other links. During this process, a number of filters checks are performed in parallel, resulting in a decision about whether each packet should or should not be invalidated by the time that the last bit is transmitted. To execute this task, the filtering hub performs rules-based filtering on several levels simultaneously, preferably with a programmable logic or other hardware device. Various methods for packet filtering in real time and without buffering with programmable logic are disclosed. The system may include constituent elements of a stateful packet filtering hub, such as microprocessors, controllers, and integrated circuits. The system may be reset, enabled, disabled, configured, and/or reconfigured with toggles or other physical switches. Audio and visual feedback may be provided regarding the operation and status of the system.

Attorney Docket No.: 802-001

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

INVENTOR AND SPECIFICATION IDENTIFICATION

My residence, post office address and citizenship are as stated below next to my name, I believe that I am the original, first and sole inventor (*if only one name is listed below*) or an original, first and joint inventor (*if plural names are listed below*) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS
TITLE OF INVENTION

the specification of which:

_X_ is attached hereto.

__ was filed on __________ as Application Serial No. ____________
and was amended on ____________________ (*if applicable*).

__ was described and claimed in PCT International Application No.____________filed on
_______________and amended under PCT Article 19 on __________ (*if any*).

REVIEW OF PAPERS AND ACKNOWLEDGMENT OF DUTY OF CANDOR

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I do not know and do not believe that the invention claimed in the above-identified specification was ever known or used in the United States of America before my or our invention thereof, or patented or described in any printed publication in any country before my or our invention thereof or more than one year prior to this application, and that the same was not in public use or on sale in the United States of America more than one year prior to this application.

I acknowledge the duty to disclose to the Patent and Trademark Office information which I know is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56.

FOREIGN APPLICATIONS AND PRIORITY CLAIM

The invention claimed in the above-described specification has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application. I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed.

COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 37 USC 119
			__Yes __No
			__Yes __No
			__Yes __No
			__Yes __No

DOMESTIC PRIORITY CLAIM

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States patent application(s) listed below and, insofar as this application discloses or claims subject matter in addition to that disclosed in the below listed priority applications, I acknowledge the duty to disclose to the Patent and Trademark Office all information known by me to be material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date(s) of the below-listed prior application(s) and the national or PCT international filing date of this application.

(APPLICATION SERIAL NO.) (FILING DATE) (STATUS: PATENTED, PENDING, ABANDONED)

(APPLICATION SERIAL NO.) (FILING DATE) (STATUS: PATENTED, PENDING, ABANDONED)

POWER OF ATTORNEY

I hereby appoint Alan R. Loudermilk (Reg. No. 32,788), who is registered to practice before the Patent and Trademark Office, as my attorney with full power of substitution and revocation, to prosecute this application, to make alterations or amendments therein, to receive the patent and transact all business in the Patent and Trademark Office connected therewith.

All **CORRESPONDENCE** should be addressed to:

Loudermilk & Associates
10950 N. Blaney Avenue Suite B
Cupertino, CA 95014

All **TELEPHONE INQUIRIES** may be directed to Alan R. Loudermilk at (408) 342-1866.

I hereby declare I have read this Declaration, and that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

HAND PRINT DATE BEFORE SIGNING

Full name of sole or first joint inventor ANDREW K. KRUMEL Citizenship USA

Inventor's signature ____________ Date 7/7/00

Residence 3635 Pleasant Knoll Drive, San Jose, CA 95148

Post Office Address 3635 Pleasant Knoll Drive, San Jose, CA 95148

Full name of second joint inventor ____________ Citizenship ____________

Inventor's signature ____________ Date ____________

Residence ____________

Post Office Address ____________

Full name of third joint inventor ____________ Citizenship ____________

Inventor's signature ____________ Date ____________

Residence ____________

Post Office Address ____________

Full name of fourth joint inventor ____________ Citizenship ____________

Inventor's signature ____________ Date ____________

Residence ____________

Post Office Address ____________

Full name of fifth joint inventor ____________ Citizenship ____________

Inventor's signature ____________ Date ____________

Residence ____________

Post Office Address ____________

_____ If this line is checked, the signature page is continued on the attached Addendum.

Attorney's Docket No. 802-001 **PATENT**

☒ Applicant ☐ Patentee

☐ Application No. ☐ Patent No.

☒ Filed on July 7, 2000 ☐ Issued on

Title: REAL TIME FIREWALL/DATA PROTECTION SYSTEMS AND METHODS

STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(c))—SMALL BUSINESS CONCERN

I hereby state that I am

☒ the owner of the small business concern identified below:

☐ an official of the small business concern empowered to act on behalf of the concern identified below:

Name of Small Business Concern 802 Systems, Inc.

Address of Small Business Concern 1580 Oakland Road, San Jose, CA 95131

I hereby state that the above identified small business concern qualifies as a small business concern, as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office under Sections 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third-party or parties controls or has the power to control both.

I hereby state that rights under contract or law have been conveyed to, and remain with, the small business concern identified above, with regard to the invention described in

☒ the specification filed herewith, with title as listed above.

☐ the application identified above.

☐ the patent identified above.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c), if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

☒ No such person, concern, or organization exists.

□ Each such person, concern or organization is listed below.

NAME____

ADDRESS______________________________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

NAME____

ADDRESS______________________________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

NAME____

ADDRESS______________________________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

☒ I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Name of Person Signing Andrew K. Krumel

Title of Person if Other Than Owner ____________________

Address of Person Signing 3635 Pleasant Knoll Drive, San Jose, CA 95148

____

____________________ Date July 7, 2000

Signature

Attorney's Docket No. 802-001 **PATENT**

☒ Applicant 802 Systems, Inc. □ Patentee

□ Application No. □ Patent No.

☒ Filed on July 7, 2000 □ Issued on

Title: Real Time Firewall/Data Protection Systems and Methods

STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(b))—INDEPENDENT INVENTOR

As a below named inventor, I hereby state that I qualify as an independent inventor, as defined in 37 CFR 1.9(c), for purposes of paying reduced fees to the United States Patent and Trademark Office under Sections 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office, with regard to the invention described in

☒ the specification filed herewith, with title as listed above.

□ the application identified above.

□ the patent identified above.

I have not assigned, granted, conveyed or licensed, and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c), if that person had made the invention, or to any concern that would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

☒ No such person, concern, or organization exists.

□ Each such person, concern or organization is listed below.*

**NOTE: Separate statements are required from each named person, concern or organization having rights to the invention as to their status as small entities. (37 CFR 1.27)*

FULL NAME____________________
ADDRESS____________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME____________________
ADDRESS____________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME____________________
ADDRESS____________________

□ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

FULL NAME____________________
ADDRESS____________________

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

(check the following item, if desired)

**NOTE: The following verification statement need not be made in accordance with the rules published on Oct. 10, 1997, 62 Fed. Reg. 52131, effective Dec. 1, 1997.*

**NOTE: "The presentation to the Office (whether by signing, filing, submitting, or later advocating) of any paper by a party, whether a practitioner or non-practitioner, constitutes a certification under § 10.18(b) of this chapter. Violations of § 10.18(b)(2) of this chapter by a party, whether a practitioner or non-practitioner, may result in the imposition of sanctions under § 10.18(c) of this chapter. Any practitioner violating § 10.18(b) may also be subject to disciplinary action. See §§ 10.18(d) and 10.23(c)(15)." 37 C.F.R. § 1.4(d)(2)*

☒ I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Andrew K. Krumel
Name of inventor

Signature of Inventor Date July 7, 2000

Name of inventor

Signature of Inventor Date

Name of inventor

Signature of Inventor Date

Name of inventor

Signature of Inventor Date

(Small Entity—Independent Inventor [7-1]—page 2 of 2

PATENT APPLICATION SERIAL NO. ____________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

```
07/18/2000 VTOWLER  00000020 500251   09611775

01 FC:201          345.00 CH
02 FC:203          414.00 CH
```

PTO-1556
(5/87)

*U.S. GPO: 1999-459-082/19144

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 29, 1999

Application or Docket Number

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
FOR	NUMBER FILED	NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	66 minus 20=	*46
INDEPENDENT CLAIMS	2 minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT		

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
	345.00	OR		690.00
X$ 9=	414	OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL	759	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A

	(Column 1)		(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1)		(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1)		(Column 2)	(Column 3)
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

MULTIPLE DEPENDENT CLAIM
FEE CALCULATION SHEET
(FOR USE WITH FORM PTO-875)

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/					
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31	/					
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
TOTAL IND.		↵		↵		↵
TOTAL DEP.						
TOTAL						

	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.		↵		↵		↵
TOTAL DEP.	16					
TOTAL	16					