PROVISIONAL PATENT APPLICATION TRANSMITTAL

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(b)(2).

Docket Number	RAPD-200	Type a plus sign (+) inside this box ->	+

INVENTOR(s)/APPLICANT(s)	
FIRST NAME, MIDDLE INITIAL, LAST NAME	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
1. JungJi John Yu	20842 Maureen Way, Saratoga, California 95070
2.	
3.	
4.	
5.	

TITLE OF THE INVENTION (280 characters max)

POLICY ENGINE ARCHITECTURE

CORRESPONDENCE ADDRESS

Alan S. Hodes
Limbach & Limbach L.L.P.
2001 Ferry Building
San Francisco
Phone: 415/433-4150; Fax: 415/433-8716

STATE	CA	ZIP CODE	94111-4262	COUNTRY	U.S.A.

ENCLOSED APPLICATION PARTS *(check all that apply)*

X	Specification *Number of Pages*	10		X	Small Entity Statement
	Drawing(s) *Number of Sheets*				Other (specify):

METHOD OF PAYMENT *(check one)*

X	A check or money order is enclosed to cover the Provisional filing fees.	PROVISIONAL FILING FEE AMOUNT(S)	$75.00
X	The Commissioner is hereby authorized to charge any additional filing fees and credit Deposit Account Number: 12-1420		

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government

X	No.
	Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE: _____ Date: December 17, 1998

TYPED or PRINTED NAME: Alan S. Hodes REGISTRATION NO. *(if appropriate)*: 38,185

CERTIFICATION UNDER 37 CFR §1.10

I hereby certify that this New Provisional Application and the documents referred to as enclosed herein are being deposited with the United States Postal Service on this date December 17, 1998, in an envelope bearing "Express Mail Post Office To Addressee" Mailing Label Number EL186214025US addressed to: Box Provisional Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

SIGNATURE: Cassandra Ku Date: 12/17/98

PROVISIONAL APPLICATION FOR PATENT

POLICY ENGINE ARCHITECTURE

Atty Docket: RAPD-200

This document describes the system architecture of an embodiment of an inventive policy-based network equipment. This system architecture is suitable for policy-based applications such as Virtual Private Networks (VPN), Firewall, Traffic Management, Network Address Translation, Network Monitoring, TOS Marking, etc. The architecture includes three major components – Policy-Based Application, Policy Engine API and a Policy Engine. Of course, each of these components may be inventive in and of itself.

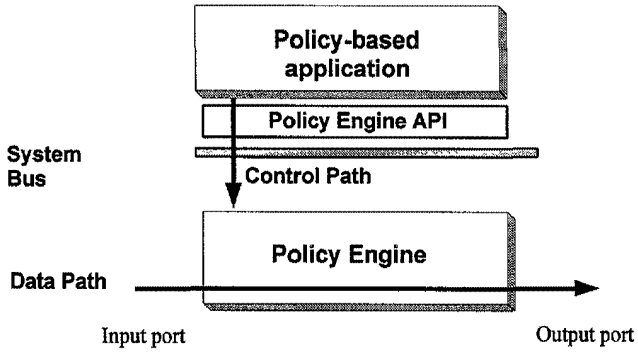

Fig. 1 – Block Diagram

Policy-Based Application: A policy-based application such as firewall, VPN, or traffic management is typically a software program residing on a host, equipped with its own policy data base and flow classification logic. The policy-based application typically has access to the media through an OS driver interface. It examines every packet coming in from the network along the data path, compares it against the flow classification criteria, and performs the necessary actions based upon the policies defined in the policy database.

Policy Engine API (PAPI): This API serves as an interface to the policy engine. It allows the policy-based application to access to all the media I/O through a generic OS driver interface. In addition, the API allows the application to invoke the acceleration functions provided by the policy engine. The application can speed up the overall system performance by turning on the appropriate acceleration functions (action processors) on the policy engine.

Policy Engine: A Policy Engine is a purpose-built hardware engine that takes in two inputs – network traffic and network policies. It then outputs regulated traffic flows based upon the specifications of the network policies. The Policy Engine preferably runs at wire speed.

Before describing a detailed embodiment, several terms are defined.

Service

A Service in a policy-based network defines a network application that is controlled and managed by a set of policies. Typical services are firewall, VPN, traffic management, network address translation, network monitoring, etc.

Policy

A Policy is defined by network managers to describe network traffic behaviors based upon business needs. It specifies what types of traffic are subject to policy control and how to control them. A policy has two components: Flow Classification Spec and Actions Spec.

Flow Classification Spec

Flow Classification Spec provides the screening criteria for a classifier to sort the packets of network traffic into flows. The Flow Classification Spec can be very elaborate. They can be as detailed as defining a pair of hosts running a specific application. Alternately, they can have a simple wildcard expression.

Action Spec

Action Spec is to describe what to do with packets that match the Flow Classification Spec. The Action Spec can be as simple as a discard or forward decision in the firewall case. It can also be as complicated as IPSec encryption rules based on SA (Security Association) spec.

Flow

All packets that match the same Flow Classification Spec form a Flow.

Stream

A Stream is a stream of packets that have the same source and destination address, source and destination port, and protocol type. (Optionally, the application can add the input and output media interface to the stream classification criteria in addition to the packet header if desired.) RapidStream policy engine sorts packets into Streams. A flow may include of one or more Streams. All packets belong to the same Stream share the same policy.

Stream Spec

A Stream Spec is the criteria used by the Stream Classifier to uniquely identify a stream. In one embodiment, it is the 5-tuple in a packet header – source and destination address, source and destination port, and protocol type.

Integrated Services

When multiple network services are to apply to the same flow, it is called "Integrated Services". Integrated Services simplify the management of various service policies, minimize potential policy conflicts and reduce TCO (Total Cost of Ownership).

Flow Classifier

In a typical conventional policy-based network, a policy decision is typically derived from a policy database. As discussed above, a flow is a stream of correlated packets to which policy decisions apply.

When a packet arrives, a flow classifier typically classifies the packet and finds a action spec according to some predefined matching criteria. The found action spec is then passed to a action processor for policy enforcement. The process of flow classification and action processing may repeat for many iterations as multiple policies are activated at the same time as shown in the Fig 2. For example, a VPN (virtual private network) application may comprise Firewall Policy, IPSEC Policy, IPCOMP (IP compression) policy, NAT (Network Address Translation) Policy, QoS (Quality of Service)policy, Monitoring Policy, L2TP/PPTP (L2 Tunnel Protocol/ Point To Point Tunnel Protocol) Tunnel Policy, and so on.

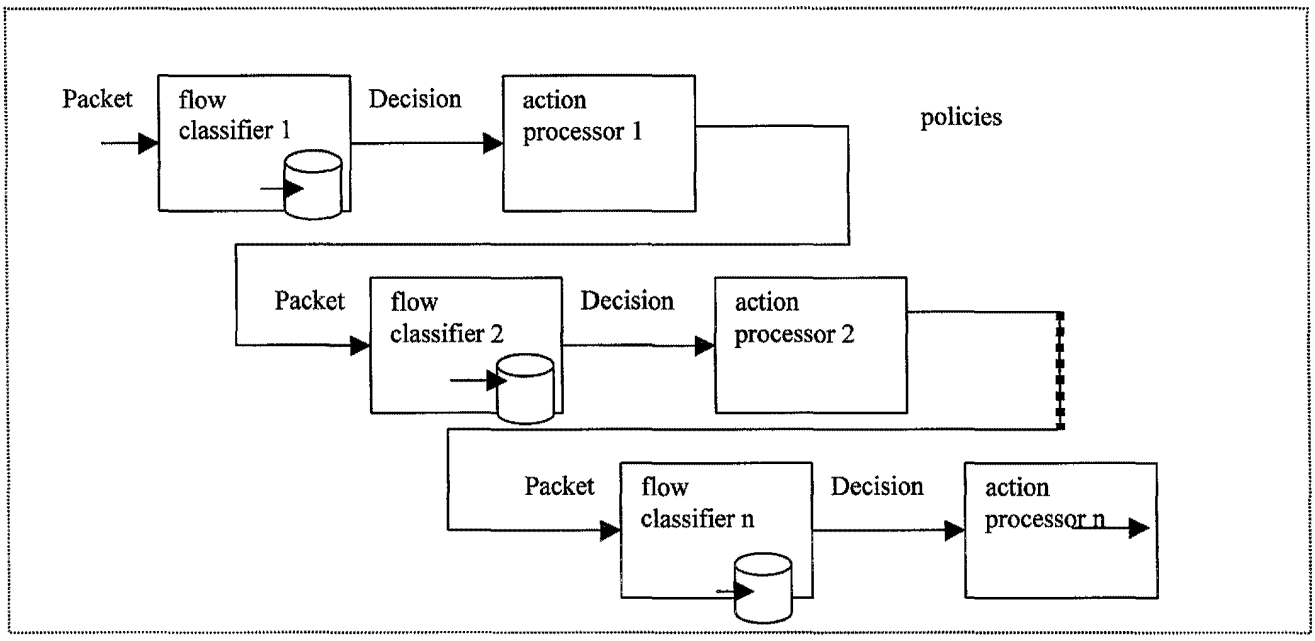

Fig. 2 - Example of Flow Classification and Action Processing in Typical Conventional Policy-based Network

The flow classification is a rule based operation can be very flexible to tune to application needs. For example, it may define a rule to identify packets with a pattern of any random bytes within a packet, and/or across many packets. The flow classifiers may also differ per action processor for performance optimization.

As a result the matching criteria used by a flow classifier to classify a flow may include a specific value, a range, or wildcard on interface port numbers, protocols, IP addresses, TCP ports, applications, application data, or any user specifiable criteria. The distinctions of various implementation makes it difficult to cache a flow with its decision in many ways.

Stream Classifier

Stream Classifier is the component that classifies packets into Streams based upon the packets' header info.

Action Processor

Action Processor is the component that executes the action based upon the action spec.

Policy Binding

Policy Binding is the process to bind a stream with its associated action specs.

Policy Cache

At the completion of the policy binding process, an entry for a given Stream is created on the policy engine which contains all the policy info (Action Specs, etc.). The collection of all active entries is called Policy Cache.

Packet Tagging

Certain applications (e.g. Network Monitoring) would like to receive flows based on the flow classification spec and would prefer the policy engine to do the flow classification for them. Packet Tagging is a way of tagging all incoming packets with an application specified "tag".

Now, some Policy Engine design considerations are discussed.

Today's policy-based applications are challenged with several key issues. These issues can be major inhibitors for the future growth of the emerging industry:

1) flow classification overhead – Flow classification specs can be complicated and lengthy for each network service. Each packet needs to be compared with potentially hundreds of rules in order to find the matching one and determine the proper actions. With stateful applications, state tracking is even more time consuming. Multiple network services on a same system simply make matters worse.

2) flow classification technique is evolving – Flow classification and analysis technique is more than just looking into the packet's address, port number and protocol type and or other header information. It often involves state tracking for newer applications. This technique is a continuous improving process and not suitable for hardware based implementation. And most importantly, flow classification techniques are often viewed as the key differentiators for vendors.

3) action execution speed – Once the classification process is complete, the proper actions need to take place. Some of the actions are simple like discard or forwarding decision for firewall, while some others are extremely time consuming like triple-DES encryption and SHA hashing algorithm or QOS scheduling algorithm. Software based implementations cannot keep up with the bandwidth expansion as the industry moves into newer and faster media technologies.

4) integrated services – As more and more policy-based applications become available, it is desirable to provide integrated services on a single platform because it reduces policy management complexity, avoids potential policy conflicts, and lowers the TCO (Total Cost of Ownership). On the other hand, integrated services impose a huge computing power requirement that simply can not be achieved with off-the-shelf general purpose machines.

A policy engine is designed to address some or all of the above performance considerations. It preferably comes equipped with a Policy Engine API (PAPI). PAPI design takes into account the following considerations:

1) Time-to-market for application developers – Understanding that time-to-market is a major concern for the application vendors, PAPI design preferably minimizes the development effort required by the application developers in order for the existing applications to take advantages of policy engine's performance.

2) Maintain flexibility for developers' value-added – PAPI may allow application developers to enhance or maintain their value-add so that vendors' differentiation is not compromised.

3) Platform for integrated services – PAPI has the model of an integrated services platform in mind. Application developers can over time migrate their services into an integrated platform without worrying about the extensibility of the API and the performance penalty.

Components of a detailed embodiment are now described.

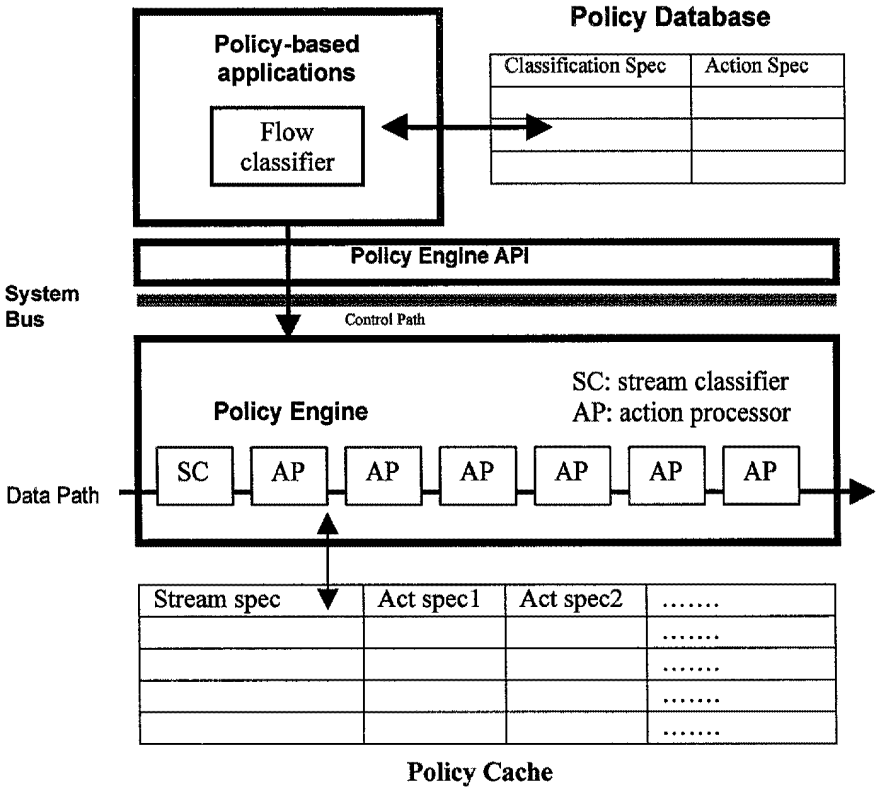

Fig. 3

Policy-based Application

A policy-based application provides a service to the network users. This service is managed by a set of policies. Firewall, VPN and Traffic Management are the most typical policy-based applications. As the industry evolves, policy-based applications are likely to consolidate onto a single platform called Integrated Services. Integrated Services has the benefits of centralized policy management and lower cost of ownership.

A policy-based application is typically equipped with its own policy database and flow classifier. A policy database contains a list of policies. Each policy has two fields – the flow classification spec and the action spec. The flow classification spec is used by the flow classifier to sort the incoming network traffic into flows. The action spec is to instruct the application what to do with the packets associated with that flow. The application performs the proper actions against the packets based upon the policy database for each flow. The massaged packets are then forwarded to the destination.

This is shown to a different level of detail in Fig. 4:

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.