(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 986 885 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
12.01.2005 Bulletin 2005/02

(21) Application number: **98926168.0**

(22) Date of filing: **01.06.1998**

(51) Int Cl.^{7}: **H04L 29/06**

(86) International application number:
PCT/US1998/011201

(87) International publication number:
WO 1998/056150 (10.12.1998 Gazette 1998/49)

(54) SHARED USE OF A NETWORK INTERFACE CARD BETWEEN HETEROGENEOUS COMPUTER SYSTEMS

VON HETEROGENEN RECHNERSYSTEMEN MITBENUTZUNG EINER NETZSCHNITTSTELLENKARTE

UTILISATION PARTAGEE D'UNE CARTE D'INTERFACE RESEAU ENTRE SYSTEMES INFORMATIQUES HETEROGENES

(84) Designated Contracting States:
DE GB

(30) Priority: **02.06.1997 US 48723 P**

(43) Date of publication of application:
22.03.2000 Bulletin 2000/12

(73) Proprietor: **UNISYS CORPORATION**
Blue Bell, PA 19424-0001 (US)

(72) Inventors:
- **JOHNSON, Robert, A.**
  Pottstown, PA 19464 (US)
- **EBERSOLE, Dwayne, E.**
  Lancaster, PA 17602-4139 (US)
- **DISNEY, William, W.**
  Downingtown, PA 19335 (US)

(74) Representative: **Modiano, Guido, Dr.-Ing. et al**
Modiano, Josif, Pisanty & Staub,
Baaderstrasse 3
80469 München (DE)

(56) References cited:
WO-A-97/01944

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND

Field of the Invention

[0001] The present invention relates to the field of computer networking, and, more particularly, to apparatus and methods for allowing two computer systems to access a network via a shared network interface card installed on one of them.

Description of the Prior Art

[0002] The ability for heterogeneous computer systems to communicate with each other over a network using standard and/or proprietary networking protocols is known. Most computer systems have some form of networking architecture that enables the computer system to perform networking in accordance with those protocols. Such a networking architecture typically comprises both system software and hardware. Figure 1 is a block diagram illustrating the components of a networking architecture employed by a Unisys A Series enterprise server 10 in order to communicate with other hosts. or nodes, on a network 15.

[0003] The A Series enterprise server 10 executes the Unisys MCP operating system 12, and has an I/O subsystem that comprises one or more I/O Modules (IOM) 14 housed within the A Series chassis. The IOM 14 implements a Unisys proprietary I/O bus architecture referred to as CS-BUS II or CS-Bus III (hereinafter "the CS Bus"). A plurality of card slots, e.g. slots 16a-d, are provided for connecting interface cards, referred to as "channel adapters", into the CS Bus. Different groups, or racks, of channel adapter slots are each controlled by a Channel Manager Unit (CMU) (e.g., CMUs 18a, 18b). An IOM can contain several CMUs, each of which controls a different rack of channel adapter card slots via the CS-Bus. The CMUs manage the physical and data layers of the I/O process.

[0004] Channel adapter cards, which each may occupy one or more channel adapter card slots within the IOM 14, provide various connectivity solutions for the A Series enterprise server 10. For example, Unisys provides a channel adapter card that implements the Small Computer System Interface(SCSI) protocol for connecting SCSI peripherals to the enterprise server.

[0005] For network connectivity, Unisys provides several channel adapters to support various physical networking protocols. These channel adapters are generally referred to as network processors (NP). For example, Unisys ICP22 and ICP26 network processors are channel adapter cards that implement the Ethernet network protocol and can be used to connect an A Series enterprise server to an Ethernet network. Unisys also provides network processors for connectivity to FDDI and ATM networks. As shown in Figure 1, a number of different network processors (e.g., NPs 20a, 20b, and 20c) can be installed in respective channel adapter slots (e.g., slots 16b, 16c, and 16d) of the IOM 14, in order to provide different network connectivity solutions.

[0006] As shown in the more detailed view of network processor 20c (installed in channel adapter slot 16d), a network processor may comprise a plurality of different lines, e.g., Line0, Line1 ... LineN. A line represents a physical endpoint within a network. For example, the Unisys ICP22 network processor has two lines, each of which comprises a separate Ethernet connection - one line could be connected to one Ethernet network, and the other to a different Ethernet network.

[0007] Each line of a network processor can have one station group defined on that line. A station group consists of one or more stations. A station is a logical endpoint that represents a logical dialog on that line. Thus, more than one logical dialog can take place over a given line of a network processor. This is achieved through multiplexing. For example, with a connection-oriented networking protocol, such as the Burroughs Network Architecture - Version 2 protocol (BNAv2), one station may represent a logical dialog with one other BNAv2 host on the network, whereas another station may represent a logical dialog to a different BNAv2 host. As illustrated in Figure 1, for example, Station0 of LineN may represent a logical dialog with BNAv2 host 22, and Station1 of LineN may represent a logical dialog with BNAv2 host 24. For networking protocols that are not connection-oriented, like the Internet Protocol (IP), only one station needs to be defined to handle all communications for that protocol stack. For example, in Figure 1, StationN of LineN could be defined as the logical endpoint for all IP traffic over LineN. A Local Area Network Station Group (LANSG) module 26, which comprises software executing on the network processor 20c, provides callable procedures for creating and maintaining stations and station groups on the various lines of the network processor 20d and for sending and receiving data over them.

[0008] Other software components that execute on the network processor 20d include a Queue Service Provider (QSP) module 28, which handles the multiplexing and demultiplexing of data for all stations defined on a given NP, and two stub modules - a Network Services Manager stub (NSM-stub) 30 and a Link Layer Manager stub (LLM-stub) 32 - which interface with corresponding modules of a Core Network Services (CNS) software component 34, to and from modules within the MCP environment.

[0009] Generally, a network processor (e.g., NP 20a, 20b, or 20c) implements the data link and physical layers of the 7-layer ISO Reference Model. Higher level networking protocols that a client application 46 may wish to employ in order to communicate with applications running on different hosts of the network 15, such as the BNAv2 and TCP/IP networking protocols, are implemented as network protocol providers on the A Series system 10. A network protocol provider is a software module that implements these higher level networking protocols. For example, Unisys provides both BNAv2 Host Resident Network Provider (HRNP) modules and TCP/IP HRNP modules. In the example of Figure 1, a BNAv2 HRNP 42 and a TCP/IP HRNP 44 are shown.

[0010] The Core Network Services (CNS) software 34 provides support for the network protocol providers 42, 44 and handles the initialization and maintenance of network processors and the station groups defined thereon. Specifically, CNS 34 comprises a Network Services Manager (NSM) 36 that initializes and manages the network processors (e.g., 20a, 20b, 20c) installed in the system, and a Link Layer Manager (LLM) 38 that initializes and maintains the identity and attributes of each station group defined on a given network processor. Another component (not shown) of CNS 34 validates attributes associated with station groups and stations created on a network processor. These attributes are passed between the network processor and CNS 34 via a control dialog when the stations arc defined. Like the stub procedures for the NSM and LLM modules 36, 38, network processors also have a stub procedure (LLAH, not shown) that corresponds to the attribute handler of CNS 34. An NPSUPPORT software library 40, as well as portions of the MCP operating system 12, provide routines and procedure calls that serve as an interface between a network processor and the CNS 34 and network protocol providers 42, 44, and control loading of software to the NPs and dumping of their state.

[0011] Each network processor has an associated identifier that uniquely identifies that network processor within the system 10. When a network processor is initialized and brought on-line, the NSM-stub 30 in the network processor interfaces with the NSM 36 of CNS 34 via a control dialog in order to pass its identifier to the NSM 36. The NSM 36 manages the identifiers of all active network processors.

[0012] Each station group and station defined for a given network processor also has a unique identifier associated with it. Via a control dialog established between the LLM-stub 32 on the network processor and the LLM 38 of CNS 34, the station and station group identifiers are passed to the LLM 38 during initialization. Within the LLM 38, a station corresponds to a connection, and a station group corresponds to a connection group.

[0013] As mentioned above, the ability to define multiple stations (*i.e.,* a station group) on a single physical line of a network processor is achieved through multiplexing. Specifically, the QSP 28 in the network processor multiplexes inbound and outbound data for multiple stations on a given line. Moreover, the QSP is responsible for distributing request and response data between the NSM 36 and NSM-stub 30 and between the LLM 38 and LLM-stub 32. To that end, each entity on the network processor that receives outbound data from the MCP, including every station, the NSM-stub 30, and the LLM-stub 32, is assigned a unique Remote Queue Reference (RQR) by the QSP. The NSM-stub RQR is reported to the NSM 36 within CNS 34 via NPSUPPORT when the NP is loaded. The LLM-stub RQR is reported to the LLM 38 via the NSM 36 by the NSM-stub 30 when the NP initializes. All of the station RQRs are reported to the HRNPs as the stations open.

[0014] When a client application is required to send data via network 15 to some other host or node on the network 15, such as another BNAv2 Host or another TCP/IP host, it invokes the services of the appropriate network protocol provider, *e.g.*, 42, 44. The network protocol provider 42, 44 determines the appropriate network processor and station on which the data is to be output, adds protocol headers, and makes a corresponding request to the MCP 12 that includes the identifier of the network processor and the RQR of the station. The data and associated RQR are passed from the MCP 12 to the QSP 28 on the network processor (e.g., network processor 20c), which, in combination with the LANSG module 26, sends the data out to the network 15 via the appropriate line (*e.g.,* Line0, Line1, ... or LineN) as part of the logical dialog represented by the designated station.

[0015] When data is received from the network 15 on a given line, the LANSG module 26 determines, from header information associated with the data, the station (*i.e.* logical dialog) for which the data is intended. The LANSG and QSP modules 26, 28, in combination with portions of the MCP 12 and NPSUPPORT library 40, pass the received data to the appropriate network protocol provider 42, 44 associated with that station, along with an indication of which station received the data. For example, one of the stations on LineN of the network processor 20c of Figure 1 (*e.g.,* station0) may be defined as the logical endpoint for the BNAv2 HRNP 42, while a different station (*e.g.*, station1) may be defined as the logical endpoint on which all IP traffic over LineN is received for the TCP/IP HRNP 44. When a frame of data is received from the network on LineN, the LANSG module 26 determines from header information which of the network protocol providers (*i.e.*, stations) is intended to receive the data. This determination is performed in accordance with the methods described in commonly assigned, U.S. Patent No. 5,379,296, entitled "Method and Apparatus for Interfacing a Workstation to a Plurality of Computer Platforms" (Johnson et al.).

[0016] In addition to its use in A Series computers, the foregoing networking architecture is also employed in Unisys ClearPath HMP NX enterprise servers. A ClearPath HMP NX server comprises an A Series enterprise server tightly integrated with a server running Microsoft Window NT. Please note that "Microsoft," "Windows," and "Windows NT" are registered trademarks of Microsoft Corporation. Additional information concerning the foregoing networking architecture can be found in the following documents, each of which is available from Unisys Corporation, assignee of the present invention, and each of which is hereby incorporated by reference in its entirety:

ClearPath HMP NX Series with Windows NT Network Services Implementation Guide (Part No. 4198 6670);
BNA/CNS Network Implementation Guide, Volume 2: *Configuration* (Part No. 3789 7014);
ClearPath HMP NX Series with Windows NT Implementations and Operations Guide (Part No. 8807 6542);
ClearPath HMP NX Series with Windows NT Migration Guide (Part No. 8807 7730);
Networking Capabilities Overview (Part No. 3789 7139)
Networking Operations Reference Manual, Volumes 1 and 2: Commands and Inquiries (Part No. 3787 7917); and
Networking Products Installation Guide (Part No. 4198 4840).

[0017] Using a Unisys ICP22 network processor, which is an Ethernet-based channel adapter, it has been possible in the past for a Unisys A Series enterprise server to communicate with a workstation or personal computer (PC) over a network. An example of this ability is illustrated in Figure 2. In this example, the A Series enterprise server 10 communicates with an Intel-based workstation 48 running the Microsoft Windows NT operating system (hereinafter "the NT server"). The A Series enterprise server is connected to the network via network processor 20a. which may, for example, be a Unisys ICP22 Ethernet-based network processor.

[0018] The I/O subsystem of the NT server 48 comprises portions of the NT operating system kernel, an EISA or PCI bus 52, and appropriate device driver software. To provide network connectivity, a network interface card (NIC) 50 is installed in an available bus slot on the NT server 48. The NT server may support one or both of the PCI and EISA bus standards. NICs are available for both bus standards.

[0019] A NIC device driver 54 that typically is sold with the NIC card 50 is installed in the kernel space of the NT operating system. The NIC device driver 54 interfaces with a higher level network protocol provider, such as an implementation of the TCP/IP protocol. Microsoft Corporation provides an implementation of the TCP/IP protocol in the form of a kernel level device driver, also referred to as a transport protocol driver, named TCPIP.SYS 58. TCPIP.SYS 58 interfaces with the NIC device driver 54 via NDIS, an industry standard Network Driver Inferface Specification jointly developed by Microsoft and 3Com. NDIS defines an interface for communication between hardware-independent protocol drivers, such as TCPIP.SYS, which implement the Data Link, Network, and Transport layers of the OSI model, and hardware-dependent NIC drivers which provide an interface to the NIC hardware and which correspond to the Physical Layer of the OSI model. A client program 60 on the NT server can communicate over the network 15 in accordance with the TCP/IP protocol by issuing suitable calls via the NT operating system to the TCPIP SYS protocol driver.

[0020] Network interface cards and associated device drivers for NT servers are available from a number of Original Equipment Manufactures (OEMs). OEM NICs are available at relatively low cost for a variety of different network media standards, including Ethernet, Fast-Ethernet, etc. As new network standards evolve, OEMs are quick to design and produce NICs to support these standards. Because these NICs are developed for industry standard I/O bus architectures, such as EISA and PCI, which are found in the many computer systems today, the economies of scale result in fast cycle development times and extremely low prices for consumers.

[0021] On the contrary, it takes significantly longer and costs significantly more to design and produce a new network processor for a proprietary bus architecture, such as the CS-BUS II architecture of Unisys A Series enterprise servers. Vendors of proprietary systems cannot achieve the same economies of scale as the open system NIC vendors, and network processors, or NIC cards, for proprietary systems therefore typically cost significantly more than their open systems counterparts. To avoid the costs associated with the development of NIC cards for proprietary systems, it would be desirable if it were possible for two computer systems to each access a network via a shared network interface card installed on one of the systems. The present invention provides such a capability.

[0022] WO 97 01944 describes a mechanism that enables emulated environments within a host computer system to share a single TCP/IP protocol stack with the host system, but it does not teach or suggest a mechanism that enables two host environments, each having its own protocol stack; to share a single network interface card located in one of the systems. The mechanism described in WO 97 01944 does not address the same problem as the present invention.

SUMMARY OF THE INVENTION

[0023] The present invention is directed to methods and apparatus that enable a first network protocol provider, executing on a first computer system and having a first network address associated therewith, and a second network protocol provider, executing on a second computer system and having a second network address associated therewith, to both send and receive data over a network via a same network interface card installed on the second computer system. In accordance with a preferred embodiment thereof, the present invention comprises an interconnection that couples the input/output (I/O) subsystem of the first computer system to the I/O subsystem of the second computer system and over which data can be transmitted between systems, and a router executing on the second computer system that routes data among the first network protocol provider (via the interconnection), the second network protocol provider. and the network interface card in a manner that enables the network interface card to be shared between the first and second network protocol providers. Specifically, when data is received from the network by the network interface card, the router routes the data to the first network protocol provider, via the interconnection, when a network address received with the data matches the first network address, but passes the received data to the second network protocol provider when the network address received with the data matches the second network address. Data received from the network with a broadcast address is routed to both the first (via the interconnection) and second network protocol providers. Outgoing data from each network protocol provider is routed to the network via that same network interface card. However, when outgoing data from one of the first and second network protocol providers is addressed to the other network protocol provider, a "loopback" capability enables the router to route that data directly from one network protocol provider to the other, bypassing the network interface card. Also, outgoing data from one of the first and second network protocol providers that has a broadcast address is routed to the network via the network interface card, and also to the other network protocol provider via the loopback path (bypassing the network interface card). In all cases, data routed between the first and second computer systems is transmitted over the interconnection. Additionally, it is understood that the present invention can be used to route data to and from a plurality of network protocol providers on the first computer system and/or a plurality of network protocol providers on the second system, via the same network interface card.

[0024] Preferably, the router simulates the operation of a network protocol provider when interfacing with a device driver of the network interface card, and the router simulates the operation of a device driver of the network interface card when interfacing with the first and second network protocol providers, whereby the functionality of the router is transparent to the network interface card and to the first and second network protocol providers.

[0025] The interconnection between the I/O subsystem of the first computer system and the I/O subsystem of the second computer system preferably comprises a physical connection between the I/O subsystems over which data can be transmitted between them, and an interconnection device driver on the second computer system that controls access by the second computer system to the physical connection. The interface between the interconnection device driver and other components on the second computer system is preferably implemented in the form of a procedure registration mechanism. In this manner, different interconnection device drivers can be installed on the second computer system for different physical connections, in a manner that is transparent to the other components of the present invention. For example, when the first and second computer systems are separate physical units, the physical connection may comprise suitable hardware (*e.g.,* interface boards) installed in available slots of the I/O buses of each system and a cable that provide a connection between them. Alternatively, where the first computer system is emulated within the second system, the physical connection may be emulated within the second system in the form of a memory-to-memory connection.

[0026] One advantage of the present invention is that it can be used to enable a computer system that has a proprietary I/O subsystem to access a network via a standard network interface card developed for a more open, or widely used I/O subsystem such as that found in systems that support the Microsoft Windows NT operating system. This avoids the need to develop network interface cards for the proprietary system as new networking protocols and standards evolve. Instead, the present invention allows the proprietary system to share network interface cards designed for the open system.

[0027] Additional features and advantages of the present invention will become evident hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] The foregoing summary, as well as the following detailed description of the preferred embodiment, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings an embodiment that is presently preferred, it being understood, however, that the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

Figure 1 is a block diagram illustrating the components of a prior art networking architecture employed by Unisys A Series enterprise servers in order to communicate with other hosts, or nodes, on a network;

Figure 2 is a block diagram showing a prior art method by which a Unisys A Series enterprise server can communicate via a network with a server running Microsoft Windows NT;

Figure 3 is a block diagram illustrating one embodiment of apparatus that enables two computer systems to share a single network interface card installed on one of them, in accordance with the present invention;

Figure 4 is a block diagram illustrating an alternative embodiment of an interconnection of the apparatus of Figure 3;

Figure 5 is a block diagram illustrating yet another embodiment of the interconnection of the apparatus of Figure 3;

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.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.