PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

# INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification $^6$ :
H04L 12/28, 12/46, 12/66

A1

(11) International Publication Number: WO 00/19665

(43) International Publication Date: 6 April 2000 (06.04.00)

(21) International Application Number: PCT/US99/22773

(22) International Filing Date: 1 October 1999 (01.10.99)

(30) Priority Data:
60/102,656 1 October 1998 (01.10.98) US

(71) Applicant: CORNICE COMMUNICATIONS, INC. [US/US]; Andover Tech Center, Suite 135, One Tech Drive, Andover, MA 01810-2452 (US).

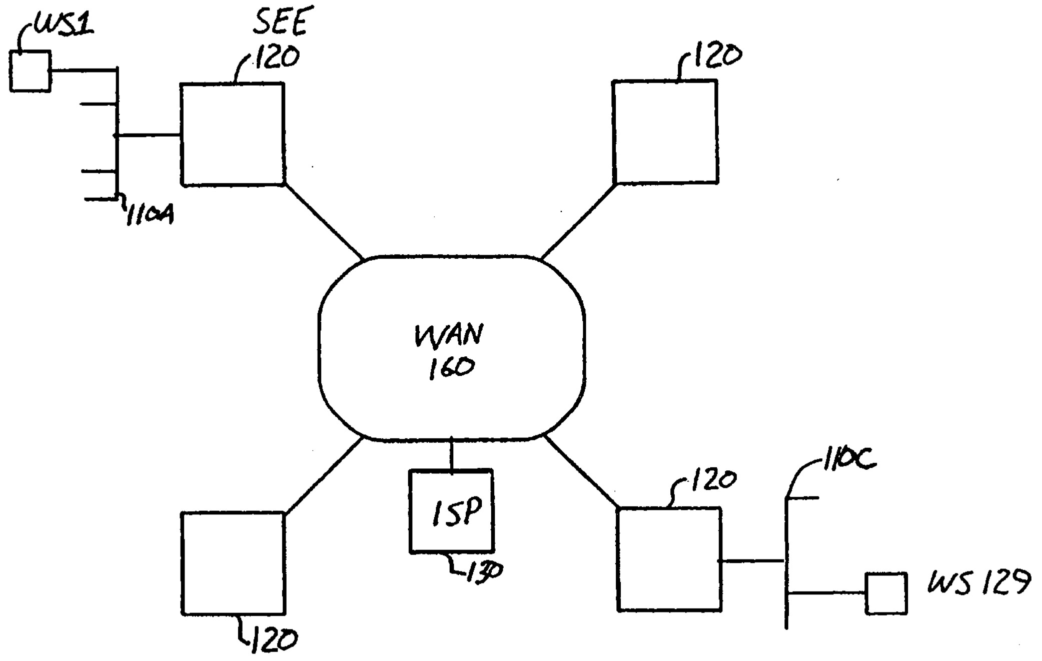
(72) Inventors: LENROW, David, R.; 12 Phinney Road, Lexington, MA 02421 (US). MILLER, Mark, W.; 15 Oak Ridge Drive, Artkinson, NH 03811 (US).

(74) Agents: STEWART, David, L. et al.; McDermott, Will & Emery, 600 13th Street, N.W., Washington, DC 20005 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, 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, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, 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
*With international search report.*

(54) Title: NETWORK TO NETWORK BANDWIDTH ON DEMAND



(57) Abstract

Subscriber endpoint equipment (120) which connects local area networks (110), such as those using TCP/IP, over wide area networks (160), such as those using ATM or frame relay, automatically detects the quality of service needed for a communications session by an application and establishes a switched virtual circuit having that quality of service. The interface can accommodate both signaled and unsignaled LANs. Protocol differences are accommodated.

*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 |

# NETWORK TO NETWORK BANDWIDTH ON DEMAND

## BACKGROUND OF THE INVENTION

### Cross-Reference to Related Applications

This application is related to Provisional Patent Application Serial No. 60/102,656, filed October 1, 1998 by David Lenrow and Mark Miller, the contents of which are incorporated herein by reference in their entirety.

### Microfiche Appendix

This application includes a microfiche appendix containing, inter alia, a listing of pseudo-code for implementing certain aspects of the invention. The microfiche appendix consists of 14 microfiche and a total of 828 frames. The microfiche appendix also includes copies of certain documents, including draft versions of standards, which, although well-known in the art at the present time, are subject to ongoing change and revision. These documents are included to capture the state of these evolving standards at this point in time.

## FIELD OF THE INVENTION

The invention relates generally to the field of interconnection of networks, and, more particularly, to subscriber end point equipment capable of selectively allocating bandwidth on demand to communications originating from a network or sub-network at a subscriber's premises for transmission across another network, such as a wide area network.

### Description of Related Art

Local area networks are well known in the art in which a plurality of subscriber terminals or workstations are interconnected over a network. Typically, local area networks are confined to a collection of devices that are located in reasonably proximity to each other. Wide area networks are similarly known in which stations which are relatively widely separated in geography can be interconnected. In certain network configurations it is desirable and known to interconnect a local area network with a wide area network. Typically, a local area network is connected to a wide area network at a node of the wide area network which is commonly referred to as an edge-switch.

It is commonly the case that large organizations have facilities that are widely separated from each other, such as being located in different cities. It is often required that local area networks for each location be interconnected. Typically such interconnection occurs over a wide area network. It is possible to interconnect a plurality of local area networks over a wide area network in such a manner as to cause it to appear to users of the local area networks that they are the only users of the wide area network and that each of the users of the local area network is interconnected with each other user, regardless of location, as if a single network existed linking them all. Such a network arrangement is commonly referred to as a virtual private network.

Interconnections between the LAN interfaces are commonly made using Permanent Virtual Circuits (PVC). Permanent Virtual Circuits are communication paths that are set up in advance and remain established so that they are available for use at any time by a subscriber desiring communications.

A Switched Virtual Circuit (SVC), on the other hand, is selectively established at the beginning of a communications session and torn down at the end of the session. A Permanent Virtual Circuit has the advantage that the signalling overhead associated with establishment and tearing down of a connection between end points is not needed since the PVC is always available. Some types of networks deliver packets of a communications session only on a “best efforts” basis. That means that no special precautions are taken to ensure a given Quality Of Service (QOS) in terms of network metrics like bandwidth available and end to end delay.

Modern wide area networks utilize ATM and frame relay switches and protocols. Other types of switches and protocols are known in the art for wide area networks. A number of protocols are also commonly used for local networks. Increasingly, local area network protocols utilize TCP/IP (Transmission Control Protocols/Internet Protocols) for communications. This is particularly convenient because TCP/IP is the communications standard for the Internet.

A number of problems result from attempting to interconnect a local area network with another local area network over a wide area network. Differences in protocol between that utilized for the local area network (e.g., TCP/IP) and that use for the wide area network (e.g., ATM) need to be accommodated. Certain types of local area networks utilize signalling, such as out of band signalling, to establish connections among users where as other networks do not. Typically, local area networks using TCP/IP do not use signalling. However, the network to which the local area networks connect may in fact be signaled. When attempting to interconnect workstations on a local area network with a wide area network, it is highly desirable that the architecture in operation at both the workstation and the underlying LAN remain unchanged from that which it was prior to interconnection with the WAN. It would also be desirable for applications running on a workstation of a local area network to be able to automatically specify a quality of service of a connection to be established over a wide area network without any modifications to the application, workstation or LAN. It would also be desirable to allow a user of a workstation to specify the quality of service needed for a particular connection and to have that quality of service vary from application to application or from session to session to permit an appropriate matching of the user needs with the type of communications undertaken.

## SUMMARY OF THE INVENTION

The present invention provides apparatus, methods, systems, techniques and computer program products which overcome the problems of the prior art and provide the advantages identified as desirable above.

The invention permits flexible interconnection between local and wide area networks by detecting a quality of service (QOS) needed for an application invoking a communication session and by creating or selecting a virtual circuit having that QOS. Singnalled and unsignalled networks can be accommodated.

Still other objects and advantages of the present invention will become readily apparent to those skilled in the art from the following detailed description, wherein only a preferred embodiment of the invention is shown and described, simply by way of illustration of the best mode contemplated of carrying out the invention. As will be realized, the invention is capable of other indifferent embodiments, and as several details are capable of modification and various obvious respects, all without departing from the invention. Accordingly, the drawings and description are to be regarded as illustrative in nature and not as restrictive.

## BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the system of the present invention will be apparent from the following description in which:

Figure 1 illustrates a network arrangement for interconnecting LANs over a WAN in accordance with one aspect of the invention.

Figure 2 is a description of the hardware architecture of subscriber and point equipment 220, shown in Figure 1, in accordance with one aspect of the invention

Figure 3 is a network diagram illustrating standard IP PVC routing as used in the prior art.

Figure 4 is a network diagram illustrating set-up of a simple switched virtual circuit (SVC) across an ATM network as shown in the prior art.

Figure 5 is a block diagram showing the software architecture of the service end point equipment 220.

Figure 6 is a block diagram of software used in the fast routing layer

Figure 7 is a flow chart used to explain operation of the fast routing layer.

Figure 8 is a network diagram illustrating set-up of a switched virtual circuit (SVC) across a wide area network in accordance with one aspect of the invention.

Figure 9 is a block diagram illustrating basic LAN-WAN Bandwidth on Demand features with an unsignaled LAN.

Figure 10 is a block diagram illustrating basic LAN-WAN Bandwidth on Demand features when a signalled LAN is utilized.

Figure 11 is a flow chart of a process for detecting applications bandwidth requirements.

Figure 12 is a block diagram used to illustrate detection of a mulitple flows for a connection.

Figure 13 is a data structure used to represent information about a flow.

Figure 14 is an exemplary drop down menu for selecting quality of service for a particular connection.

Figure 15 is a block diagram showing selection from among existing PVCs of differing quality levels when establishing a connection across a network.

Figure 16 is a block diagram showing selection from among existing types of connections (TOC) when establishing a connection across a network.

Figure 17 is a block diagram illustrating reservation of bandwidth.

Figure 18 is a pseudocode representation of an exemplary technique for evaluating new reservation requests.

Figure 19 PCT illustrates exemplary pseudo data structures used with the fast routing layer.

Figure 20 PCT is a pseudocode representation for the upper IP drivers 206 shown in figure 5.

Figure 21 PCT is a pseudocode representation of implementation of the fast routing layer 205 shown in figure 5.

Figure 22 PCT is a pseudocode representation for the lower IP drivers 204 shown in figure 5.

Figure 23 PCT is a pseudocode representation for the rules engine 301 shown in figure 5.

Figure 24 PCT is a pseudocode representation for the flow group monitor 306 shown in figure 5.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates a network arrangement for interconnecting LANs over a WAN in accordance with one aspect of the invention. Wide area network 200 serves to interconnect a plurality of local area networks 110A and 110 C. In accordance with the invention, the interface between the local area networks 110 and the WAN 160 is subscriber end point equipment (SEE) 120. Although not shown in this illustration, SEE 120 connects to the WAN 160 through an edge computer which constitutes the entry point into the WAN 160. An internet service provider, 130, is also shown connected to the WAN. For purposes of illustration, each LAN will be presumed to utilize a TCP/IP communications protocol and, for purposes of illustration, arranged in a 10 base T configuration for communications. The number of terminals or workstations such as WS-1 and WS-129 are connected to respective LANs 110. As discussed more hereinafter, each SEE 120 contains memory and routing tables such that a virtual private network can be defined in the memory of SEE 220 devices.

Figure 2 is a description of the hardware architecture of subscriber end point equipment 120, shown in Figure 1, in accordance with one aspect of the invention. The main controller for the hardware shown in Figure 2 is CPU 114. It is preferably a Motorola MPC860SAR processing element. The CPU 114 has a plurality of input-output ports which are utilized as shown in Figure 2. A memory bus 250 connection to a memory subsystem including a boot device 115a, SD RAM 115b and RAM 115c (flash memory). Nonvolatile memory 116 (Serial EEPROM) stores booting information is connected over port SPI to CPU 114. LED indicators 111 permit certain output information concerning the state of the device to be displayed to an external user. RS 232 serial console port 113 connects to the CPU over line SMC1. This permits a PC class device to be connected to the SEE 120 to serve as a console for troubleshooting and other purposes. The CPU 114 has integral protocol processing capabilities for Ethernet, HDLC, and ATM/SAR.

Network input/output is provided by items 101, 102, 103, 104, 105, 106, and 107 which include Ethernet, ATM, POTS, and frame relay capable bi-directional ports. To support multiple Utopia capable I/O devices to the CPU 114, a programmable logic device 112 is provided. This allocates the Utopia bus as needed to the connected network interfaces. A routing field programmable gate array 110 provides data path routing to signals between the main CPU/Protocol processor 114 signals and the V.35 port 104 and the second Ethernet port 105. This routing FPGA also routes TDM data between the

CPU 114, DSPs (e.g. 2 DSPs) 117, T1/E1 ports 106 and POTS port 107. The POTS port 107 includes necessary analog to digital conversion circuitry for interfacing standard POTS telephone connections to the networks.

The DSPs 117 provide high computation capabilities for any signals delivered to them via the TDM bus as routed by FPGA 110. Examples of the types of computing utilizing the DSPs include voice or data compression, DTMF tone detection and generation. Two separate content addressable memories (CAM) hang off the memory bus 250. CAM1 (108) is used for layer 3 fast look ups and CAM2 (109) is used for layer 4 fast look ups.

Figure 3 is a network diagram illustrating standard IP PVC routing as used in the prior art. Network 300 is an IP network having nodes 310 which each connect to one or more sub-networks 110 which are to be linked over network 300. The simple illustration shown, with only three nodes, each node is linked to all the other nodes over respective permanent virtual circuits. Thus, node 310a is linked to node 310c over permanent virtual circuit PVC 1. Node 310a is linked to node 310b over PVC 2 and node 310b linked to node 310c over PVC 3. As shown in the illustration, a routing table and a linked table are maintained at each node so that packets from a user such as WS1 can be directed over sub network 110 to the appropriate permanent virtual circuit to reach a node associated with a sub network of a destination user, such as WS 129 as shown in the illustration. The routing table contains entries showing each destination net (e.g., sub network) and the "next hop" to be taken to get to that destination network. The link table identifies in one field each destination node and in a related field the particular permanent virtual circuit to be taken to reach that node. For example, a packet from user WS1 is routed over sub network 110a to node 310a (IP1) where the routing table is referenced to determine that the next hop should be to IP3 in order to reach destination net 210c on which WS29 is located, to which the packet is to be directed. Once the next hop is identified, the link table is referenced to determine that the link to node IP3 should be over PVC1.

One problem that is masked by the simplicity of the example shown in Figure 3 is that the routing tables expand in their requirements as a function of N x (N-1), where N is the number of nodes on the network. Thus on a very large network, the routing tables are of substantial size and the access time needed to search for particular entries can become excessive.

Figure 4 illustrates a set-up of a simple switch virtual circuit (SVC) across an ATM network as shown in the prior art. IP1, IP2, and IP3 in this instance are gateways between a TCP/IP network 110a and 110c, respectively in an ATM network 400. Gateway IP1 connects to edge switch 410a over a well known address and port. IP1 initiates a request for a connection to be established to work station 129 of sub network 110c. In this case, no permanent virtual circuit exists between IP1 and IP3. However, the edge switch has an ATM address table as shown which relates the various destination nodes to their respective ATM addresses. Gateway IP1 requests then a virtual circuit be established for use by its user WS1 for communicating with user WS129. The routing table identifies the net portion of the address WS129 and identifies that it is served by gateway IP3. However, there is no virtual circuit identified at this time in the link table. Gateway IP1 requests the establishment of an SVC for use during this communication from edge switch 410a. Edge switch 410a has the ATM address for gateway IP3 which is serviced by edge switch 410b. Accordingly, edge switch 410a requests the establishment of a virtual circuit between itself and 410b for use during these communications. The edge switches establish the virtual circuit and make the assignment to the gateway, notifying it of the virtual circuit to be utilized for the communications requested. At this time, the identity of the virtual circuit is inserted into the link table for reference and routing. With the tables populated, communications may begin between WS1 and WS129.

Figure 5 is a block diagram showing the software architecture of the service end point equipment 120. A rough comparison to the ISO/OSI model will be useful in explaining the software architecture. Block 217 corresponds roughly to layer 1 or the physical layer of the ISO/OSI model. Blocks 201, 202, 203 and 204 correspond generally to layer 2 of the ISO/OSI model (the link layer) and block 205, 206 and 209 correspond generally to layer 3 of the ISO OSI model (the network layer).

The group of functions labeled SmartAgent handle the top level management of the software. The Web/SNMP/XML controls 210 permit the overall box configuration to be managed. Initial loading of the executable image for program memory or network is handled by the bootloader 211. Routing (both basic and items like routing table and parametrics exchange) is handled by Routing Protocols 213.

The Network I/O ports in Figure 2 (201, 202, 203, 204, 205, 206, 207) are logically presented in Figure 3 as a group 217.

The I/O configuration manager 216 configures these ports according to management commands from the SmartAgent Controls 210. The result of this configuration is the activation of one or more ports towards the enterprise side and one or more ports towards the service provider. This configuration information is then made available to the Fast Routing Layer (FRL) 205.

The FRL 505 is the central software entity for data plane activities in the SEE 120. It performs the functions necessary to move data at maximum bandwidth when the data packet is for an existing flow. A typical configuration would have the Ethernet port 517 connected to an Ethernet driver 510 as the enterprise side and a T1 port in 517 connected to an ATM driver 503. A packet arrives at 501 and is passed up to 504 after it is confirmed as valid. 504 would strip any Layer 2 information/headers (the 14 byte MAC header in this case) and send the IP packet up to the FRL 205. The FRL 205 would see if this is an existing flow (i.e., does it match source and destination for the IP/Layer 3 information, the source and destination TCP/UDP port, and protocol type) and if it does, immediately sends it to appropriate transmit point in 204. After encapsulating the IP packets (e.g., RFC 1483) it would be transmitted via the ATM driver 203 out the configured service provider port in 217.

The first time a packet travels between two new layer 3 entities (i.e., IP addresses) the packet needs to be routed by being sent up to the matching protocol adapter in 206 which in turn sends it up to the IP layer 209. Any locally destined traffic travels this same path.

Any time a flow or flow group begins or ends it is the responsibility of the group of entities 207, 208, 214, 218 referred to as Service Mapping to address the incremental work which needs to be performed. (Figure 4 details these activities.)

The final entity in Figure 5 is the Queue Manager 515. To the Fast Routing Layer 505 it appears as another formatting module in 504. The Queue Manager, however, holds and reorders the transmission of packets according to priorities given it by the

Controls 210. When a packet is ready for actual transmission, the appropriate entry point in the formatting module 204 is called. This allows for a 'shaping' (e.g., prioritization) of packets traveling on the same virtual circuit.

Figure 6 is a block diagram of software used in the fast routing layer. It will be used to describe the operation of the fast routing layer in more detail in conjunction with the flow chart of figure 7..

A packet arrives at the FRL 205 from 204, as shown in Figure 5. Within 205, a flow match 307 is performed by taking the source and destination IP address from the packet and comparing it to all known pairs via CAM 1 108. If there is no match, the packet is sent 'up' the stack to 206. If there is a match, then the match number from CAM 1 108, type of packet, source port number and destination port number are used to try for a match on CAM 2 109. If there is no match, then the packet is sent to the rules engine 301.

Packets that have matched both CAMs are existing flows and can be completely handled by FRL 205. The match information gives an index into the flow context table 312. This table contains the information for the remaining processing that needs to take place. First, if the entry in 312 indicates TOS marking is to be done, block 308 takes the TOS value in 312 and applies to the packet. Next, block 309 decrements the TTL and recalculates the IP header checksum (including TOS change if applicable.) The transmit bytes and transmit packets counters are then updated by 310. Finally, block 311 using the transmit function pointer, transmit arg, and the new packet (via the information in 312), calls the proper entry point in 204 to send the packet. The flow info 312 also contains a flag for control flows which need to be sent to the rules engine 301 so that control flows can be continously monitored.

Packets that were sent 'up' the IP stack will come back to the FRL 205 through the IP adaptation layer 206 after a next hop address has been determined by the IP layer 209. These packets are then processed by the rules engine 301. As noted above, the rules engine is also called if CAM 2 109 has no match (a layer 4 miss.)

The rules engine 301 must make a determination as to whether this is an ordinary/low priority application or a special/high priority application. This is done by comparing both static and dynamic TCP/UDP port information (i.e. Layer 4.) Static rules are constant port number assignments that exist for many applications. Dynamic rules are determined by monitoring the packets in the control channel flows for each of the different applications 303 and determining what the applications have negotiated for dynamic port numbers.

If the rules engine 301 determined a new VC is required (i.e. this is a new application session with special handling needs) by checking the application policy database 315. This activity is run by the Flow Group Monitor 306. First, if the policy indicated multiple selection option, the Customer Contact Manager 218 would be directed to select from these options. At this point, a new flow group 313 would be created, flow group activites in 304, 305, 306 would be started and the new VC creation would be started.

To initiate the new VC, the rules engine 301 would tell block 214 to initiate SVC signalling. Next, an entry into the system flow table in 312 would be added, indicating either transmit onto a default channel or queue packets as a starting state. When 214 completes the WAN signalling this flow table entry 312 is updated to reflect the new SVC information.

The flow management monitor also 208 creates the information necessary for billing and Service Level Agreement (SLA) monitoring based on and contained in 313. Billing information includes start time, end time, application type, source IP, destination IP, statistics (transmitted bytes and packets, received bytes and packets), and QOS related parameters.

When the primary control channel monitoring by 301 and 303 indicates the session is ending, the flow monitor 208 will tell 214 to terminate (teardown) the VC. Next it will gather all of the statistics and SLA information from 312, 313, and 305 and a session or call detail record will be sent to the billing host by 304.

The final role for the flow group monitor 306 is to perform periodic (minute granularity) housekeeping on the flows that are in the default group (for timeouts) so that billing can also be performed on this class of information. This way, applications in 315 can be marked to go onto the default channel but still be billed, while the bulk data (non-application specific) traffic information is also periodically reported.

Figure 8 is a network diagram illustrating set-up of a switched virtual circuit (SVC) across a wide area network in accordance with one aspect of the invention.

Figure 8 is similar to the drawing similar to figure 4. However, the gateway of Figure 4 lack the ability to set up a switched virtual circuit with the quality of service specified by the gateway. The SEE 120, as described more and hereafter, detects the quality of service required for a connection and establishes a switched virtual circuit having that quality of service across the WAN to the destination. In this particular mode of operation, each new connection utilizing the services of the WAN gets a level of service commensurate with either the application utilizing the communication services, or a level of service specified by a user as described more and hereinafter.

Figure 9 is a block diagram illustrating basic LAN-WAN Bandwidth on Demand features with an unsignaled LAN. First, a packet arrives at LAN (typically Ethernet) port 1. Based on the LAN Flow Identification 2, a packet may go to either: a) up through to local IP stack, b) directly to the rules engine 4. Packets for a) are sent 'up' the IP stack for other local clients and require no additional special handling. Packets which are destined for the WAN side (as determined by 2) but which are not part of an existing Layer 3 (IP) connection need to be routed first by 7 (i.e. a next hop IP address is required) which is part of the system's IP stack. Packets for b) go to the flow rules engine 4. If the rules engine 4 determines that the packet is part of an existing flow, the packets are first encapsulated in block 3 (e.g. via an RFC1483 standard header addition) and then sent along to be transmitted out the proper VC by 6 and then out the WAN port 8.

If the rules engine 4 finds this is a new flow, a determination is made as to whether this is an ordinary/low priority application or a special/high priority application. This is done in block 4 according to a rules comparison engine based on both static and dynamic TCP/UDP port information (i.e. Layer 4) and IP addresses (i.e. Layer 3.) Static rules are constant port number assignments that exist for many applications. Dynamic rules are determined by the system in block 4 monitoring the packets in the control channel flows for each of the different applications and determining what the applications have negotiated for dynamic port numbers. If the rules engine 4 determined a new VC is required (i.e. this is a new application session with special handling needs), the system would initiate SVC signalling out the WAN port via block 5. Next, an entry into the system flow table in 4 would be added. When the WAN signalling completed this flow table entry would be updated by 5 to reflect the new SVC information. This SVC to flow information provides the basis for block 6 to send packets on the VC in 8 associated with a given flow.

Figure 10 is a block diagram illustrating basic LAN-WAN Bandwidth on Demand features when a signalled LAN is utilized.

Packet arrives at LAN (typically Ethernet) port 11. Packet may go to one of several places: a) proxy for LAN signalling 15, b) up through to local IP stack, c) directly routed to appropriate output port based on the determination made by block 12.

Packets for a) go to the local LAN signalling agent 15. Upon arrival at 15 several actions would take place. First, the appropriate response to the signalling would be sent back out the LAN port. Second, if the signalling indicated a new flow was going to be taking place, the system would determine is a new WAN SVC was appropriate or if the new flow should go onto an existing VC in which case the flow information would be added in 14 for this.

If 14 determined a new VC was needed, the system would initiate SVC signalling out the WAN port via block 16. Next, an entry into the system flow table in 14 would be added. When the WAN signalling completed this flow table entry would be updated by 16 to reflect the new SVC information. This SVC to flow information provides the basis for block 18 to send packets on the VC in 19 associated with a given flow.

Packets for b) are sent 'up' the IP stack for local clients and require no additional special handling. Packets which are destined for the WAN side (as determined by 12) but which are not part of an existing Layer 3 (IP) connection need to be routed first by 17 (i.e. a next hop IP address is required) which is part of the system's IP stack.

Packets for c) (i.e. they are part of a known Layer 3/IP connection) are first encapsulated in block 13 (e.g. an RFC1483 standard header addition) and then sent along to be transmitted out the proper VC by 18 and then out the WAN port 19.

Figure 11 is a flowchart of a process for establishing SVC's with selectable QOS. When packets are received at the start of a packet flow (1100), a check is made at the SEE 120 to determine whether an existing virtual circuit exists between the SEE 120 and the desired destination (1110). If such a virtual circuit exists, (1110-Y) the packet flow will be directed to the existing virtual circuit for transmission to the destination (1120). If there is no existing virtual circuit (1110-N), the flow is routed to a queue (1130) and simultaneously to an application specific analyzer. If the connection is a multi-flow connection, such as found in ITU standard H.323, one will add table entries as additional flows are detected that are associated with the H.323 connection (1140). While packets are flowing to the queue, the user will be queried for the quality of service needed or desired for the connection and the user's response recorded (1150). Once the user has indicated the quality of service needed, or in the case of automatic detection of the quality of service needed as discussed more hereinafter, the SEE 120 requests establishment of an SVC at the QOS specified (1160). That SVC is then established at that QOS (1170) and packets are released from the queue to flow over the newly established SVC (1180). Communications can then begin over the new SVC (1190). If communications began initially over an existing SVC (1120), once the new SVC has been established at the appropriate QOS, the flows associated with that connection will be routed away from the existing SVC and over the newly established SVC.

Figure 12 is a block diagram used to illustrate detection of a multiple flows for a connection. When a workstation WS1 desires to communicate in a mode which requires multiple flows, such as when device WS1 desires to communicate using the ITU H.323 standards (multimedia applications), a connection is requested to a destination device addressed to a particular well known port which is utilized for establishment of connection with H.323 devices. In an example shown, an initial flow is established carrying a request for a connection. The request specifies a connection going to port 86, in this example, and results in an assignment received from the connected device of a second port number, such as 122 in the example shown over which control information regarding the set-up of data flows needed for the H.323 connection can be transmitted. In response to signalling exchange to over port 122, one or more data flows such as 193 shown can be established to handle respective streams or flows of multimedia information. In the SEE 120, when control flows, such as those to ports 86 and 122 are detected, the packet flow is teed off the normal routing path to permit an application specific analyzer to monitor the flows at the same time packet information is flowing across the communications connection. The plurality of flows associated with a service, such as the H.323 service constitute part of a flow group. At the end of the communications, the disconnecting device will send a tear-down message across control channel 2 (port 122 in this example), directing tear-down of the SVC used to handle the flow group.

Figure 13 is a data structure used to represent information about a flow. There will of course be a flow ID constituting a five tuple of information. That five tuple of information includes: source IP address, destination IP address, source port number, destination port number and protocol type. The data structure also includes SVC information, such as port number, virtual path indicator (VPI) and virtual connection indicator (VCI) and an encapsulation header. By storing the flow ID in the fast routing layer, one can "route once" and "forward many". That is, the routing information is already stored for the flow and can be utilized repeatedly to route packets without a full routing process utilized for a new connection.

Figure 14 is an exemplary drop down menu for selecting quality of service for a particular connection. Such a drop down menu can also be utilzied to specify a quality of service for a particular application type. This particular drop down menu provides the user with the opportunity to select one of several options as to network quality and as to network security. This particular form of drop down menu has additional advantages in that it provides the opportunity for advertising to be displayed to a user as part of the network connection quality selection process.

As discussed earlier, as table size increases with the number of connections through a network, the look up time and resources required to access routing information increases dynamically. If the look up is done in the software, as conventionally the case, the amount of time required for a particular routing process is a function of the number of flows underway.

By monitoring newly created flows in the SEE 120, one may determine the particular type or class of application that has been invoked. Various types of applications can be distinguished by the well known ports to which they direct their initial connection request. Another example of the type of communication where quality of service will vary depending upon well known ports to which a connection request is directed is that utilized by SAP software which is widely used in business for accounting and inventory control and management.

In a typical network environment, one would expect many applications to accept a "best efforts" type of service, whereas others, such as video phone, might require a high quality of service. Distinguishing among these types of applications can be done effectively as described. Alternatively, other distinguishing characteristics of applications can be detected and utilized to establish quality of service across the network. As discussed above, the SEE 120 utilizes content addressable memories (CAM) for routing determinations. In a preferred embodiment, one content addressable memory is utilized for level 3 routing and one is utilized for level 4 routing. This particular configuration is flexible and provides a richer set of capabilities than would otherwise be the case. Content addressable memories have the advantage that their time per calculation is deterministic. As shown in figure 5, allowing the support for any particular port and protocol to be split across hardware and software gives a greater flexibility of assignment than either purely hardware or purely software approach. It allows, for example, various portions to be swapped out in the software while retaining the existing hardware infrastructure.

Figure 15 is a block diagram showing selection from among existing PVCs of differing quality levels when establishing a connection across a network. The SEE 120 can also be utilized to select among a plurality of existing PVC's, each having a different QOS. In certain Internet applications, for example, when schemes like Diffserv are utilized, it is possible for multiple PVC's to be established at different levels. In the case of Diffserve, for example, multiple levels (e.g. bronze, silver, gold and platinum) service are defined. If PVC's are established across a network at different levels of service, the level of bronze might be utilized for "best efforts" communications and the others dedicated to particular QOS specific connections.

There are two problems associated with this approach. First, the user has no say about changing the quality of service because the establishment of the connection is associated relatively permanently with either the user or the application. Therefore, the user may be paying for more capability than needed for a particular session or may be unable to guarantee the level of service needed for a particular communications. By utilizing the selection capabilities of the SEE 120, the user can specify, on a connection by connection basis, or by class of application, the quality of service needed for a particular session. The SEE 120 can also be used in a topology like figure 15 to select the PVC based on the "type of service" (TOS) bits in the IP header of incoming packets, resulting in QOS across the network for appropriately marked packets.

Figure 16 illustrates a network arrangement for interconnecting LAN over a WAN. In certain types of WAN networks, the QOS for a given packet is based on the TOS bits settings in the IP header of the packet. The SEE 120 can be utilized to set the TOS bits appropriate for the particular session or appropriate for the user selection, so that the WAN network will transport these packets with the QOS desired.

Most access to network bandwidth today is on a first-come, first-serve basis. This invention also allows a more controlled manner of access to this bandwidth, one in which the bandwidth can be reserved in advance (minutes, hours, days) so that a user knows it will be available when needed. This bandwidth reservation system (BRS) involves a user 'talking' to a network box to reserve a certain amount and quality of bandwidth at a time in the future. For instance, a user wants to have a teleconference across the WAN to a remote office 'next Thursday from 10:00AM to 11:30AM'. The teleconference software requires 384Kbit/second with less than 30 milliseconds of latency. The users would communicate this information to their WAN access boxes on either end of this link and if there was enough low latency bandwidth available on both boxes at the time required, then the boxes would confirm the appointment and not let anyone else reserve that time slot or allow other users to interfere when that appointment time was reached. If the user cancelled his reservation at any time prior to the appointment then this reservation would be put back into a pool and others could access it freely.

This permits a number of advantages, namely:

- it's ability to allow advance reservation of bandwidth amount and quality rather than hoping its available when needed;
- allowance for recognition of 'higher authority' requests to override existing reservations. E.g. if the CEO makes a reservation request and it's turned down, the system will allow a hierachical override to take place. I.e. the CEO bumps whoever had the reservation. (However, the CEO is never going to have a reservation bumped!)
- it's intuitive match to normal business requirements
- allows predictable access to network resources
- recognizes priority users versus ordinary users
- clean fit into current business practices

Figure 17 is a block diagram illustrating a Bandwidth Reservation System (BRS). The reservation request is received by the reservation server and is first checked for a valid format. Bad format results in rejected request. A valid request includes sufficient information to completely quantify the network based resources required and the time, date and duration of the requirement. The request is then compared for validity against the basic capabilities of the affected network elements (i.e. is it within the basic bandwidth, processing resources, etc. that the transport equipment can support). Inadequate capabilities result in a failed reservation request. The request enters the conflict/priority resolution process. This process will determine whether the required resources can be reserved according to initiator's criteria. For any of the resources being requested, the time, date and duration of the reservation request is compared against a database of existing reservations for network resources. If the required resources are not committed to a conflicting reservation the new reservation function is initiated . If there is a conflicting resource reservation, then a check of the user priority in the policy database is performed. If the user making the new request has been defined to have a higher priority than the holder of the existing reservation, the lower priority reservation is released and the new reservation function is initiated. Notification of released reservation to its holder also takes place. If a pre-existing reservation holder has higher priority than the new request, the new request returns a "failed" indication. The new reservation function commits a transaction with the reservation database resulting in a new reservation for specified time, date duration, and resources.

The BRS includes a user policy database. This database would include a field for every system user with an eight bit reservation priority value. A value of zero would indicate the lowest priority for reservations and a value of 255 would indicate the highest reservation priority

The BRS also includes a reservation database. This database would include, for each network resource which can potentially be reserved, a variable number of data structures describing reservations for the resource. Each such data structure would include the following information:

ID of the user holding the reservation,
Start time of reservation,
End time of reservation, and
Amount of resource reserved.

Figure 18 is a pseudocode representation of an exemplary implementation for evaluating new reservation requests.

Pseudo code and Pseudo data structures and standards utilized for implementing the SEE 120 are set forth in detail in the attached microfiche appendix and are incorporated in their entirety into the disclosure hereof by reference.

Figure 19 PCT illustrates exemplary pseudo data structures used with the fast routing layer.

Figure 20 PCT is a pseudocode representation for the upper IP drivers 206 shown in figure 5.

Figure 21 PCT is a pseudocode representation of implementation of the fast routing layer 205 shown in figure 5.

Figure 22 PCT is a pseudocode representation for the lower IP drivers 204 shown in figure 5.

Figure 23 PCT is a pseudocode representation for the rules engine 301 shown in figure 5.

Figure 24 PCT is a pseudocode representation for the flow group monitor 306 shown in figure 5.

Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

What is claimed is:

1. Apparatus for interconnecting a first digital network that does not use signalling to establish connections with a second network that uses signalling , comprising:
   a. An analyzer for recognizing that at least one packet originates from a particular application that requires a new communications session and for determining the destination for the least one packet; and
   b. A signalling generator for sending signalling to said second network to establish a connection to said destination.
2. The apparatus of claim 1 in which said first digital network is a Ethernet network.
3. The apparatus of claim 1 in which said second digital network is an ATM network.
4. The apparatus of claim 1 in which signalling sent to said second network comprises an indication of quality of service desired for said connection.
5. The apparatus of claim 1 in which said second network is a network selected from the group consisting of an ATM network, a frame relay network, packet over SONET and a TDM network..
6. The apparatus of claim 1 in which packets destined for an address on said first digital network are directed to a protocol stack for said first network.
7. The apparatus of claim 1 in which packets destined for said second network are directed to a circuit for processing said packets for transmission across said second network.

8. Apparatus for interconnecting a first digital network to establish connections with a second network, comprising:
   a. An analyzer for determining of the destination of at least one packet from said first digital network and directing it to a rules engine for processing to determine quality of service needed for a connection across said second network; and
   b. A signalling generator for sending signalling to said second network to establish a connection with said quality of service to said destination.

9. Apparatus of claim 8 in which the rules engine requests quality of service information from a user.

10. Apparatus for routing communications comprising:

   a. A monitor for monitoring at least one packet from a first digital network to detect destination of said packets; and

   b. A circuit for directing said at least one packet both to a second network and to a fast router.

11. The apparatus of claim 10 in which said fast router either identifies a route, if a connection has been previously established or directs the at least one packet to a rules engine for determination of the characteristics of a circuit required for servicing a new connection.

12. The apparatus of claim 11 in which the new connection is a switched virtual circuit.

13. Apparatus of claim 11 in which said connection is to an H.323 compatible device.

14. Apparatus of claim 13 in which said monitor detects at least two flows constituting a flow group associated with connection to said H.323 device.

15. Apparatus of claim 14 further comprising a circuit for establishing a virtual circuit for transporting all flows of a flow group.

16. Apparatus of claim 14 in which said monitor monitors said at least two flows until an indication is received in at least one flow of said flow group that the connection is to be torn down in response to which the monitor initiates tear down of the connection.

17. Apparatus of claim 11 in which said connection is to a device running SAP software.

18. Apparatus of claim 16 in which said monitor detects a packet directed to a well known port used for accessing said SAP software.

19. The apparatus of claim 10 further comprising a circuit for encapsulating the contents of said at least one packet with a protocol of said second network.

20. The apparatus of claim 10 in which said second digital network is an ATM network.

21. The apparatus of claim 10 in which said at least one packet is an ATM packet containing at least part of an encapsulated ip packet and said second network is an ATM network.

22. The apparatus of claim 10 in which said second network is a frame relay network.

23. The apparatus of claim 10 in which said at least one packet is an frame relay packet

containing at least part of an encapsulated ip packets and said second network is an ATM network.

24. The apparatus of claim 10 in which said second network contains multiple PVCs at least two of which have differing quality of service specification and a rules engine selects which PVC to use.

25. Apparatus for interconnecting a first digital network that uses signalling to establish connections with a second network that uses different signalling , comprising:

a. an analyzer for determining of the destination of at least one packet from said first digital network;

b. a signalling generator for sending said different signalling to said second network to establish a connection to said destination, and

c. a mechanism for linking data planes from said first and second network.based on connections established by said signalling.

26. A method for interconnecting a first digital network that does not use signalling to establish connections with a second network that uses signalling, comprising the steps of:

a. recognizing that at least one packet originates from a particular application that requires a new communications session;

b. determining of the destination of said at least one packet from said first digital network; and

c. sending signalling to said second network to establish a connection to said destination.

27. A method for interconnecting a first digital network to establish connections with a second network, comprising:

a. determining of the destination of at least one packet from said first digital network and processing to determine quality of service needed for a connection across said second network; and

b. sending signalling to said second network to establish a connection with said quality of service to said destination.

28. A method for routing communications comprising comprising the steps of:

a. monitoring at least one packet from a first digital network to detect destination of said packets; and

b. directing said at least one packet both to a second network and to a fast router.

29. A method for interconnecting a first digital network that uses signalling to establish connections with a second network that uses different signalling , comprising the steps of:

a. determining of the destination of at least one packet from said first digital network;

b. sending said different signalling to said second network to establish a connection to said destination; and

c. linking data planes from said first and second network.based on connections established by said signalling.

30. A computer program product comprising;

a. a memory medium, and

b. a computer program, stored on said memory medium, said computer program comprising instructions for recognizing that at least one packet originates from a particular application that requires a new communications session, determining the destination of at least one packet from a first digital network that does not use signalling and sending signalling to a second network that does use signalling to establish a connection to said destination.

31. A computer program product comprising;

a. a memory medium, and

b. a computer program stored on said memory medium, said program comprising instructions for determining of the destination of at least one packet from a first digital network and processing said at least one packet to determine quality of service needed for a connection across a second network; and sending signalling to said second network to establish a connection with said quality of service to said destination.

32. A computer program product comprising;
    a. a memory medium, and
    b. a computer program, stored on said memory medium, said computer program comprising instructions for monitoring at least one packet from a first digital network to detect destination of at least one packet and for directing said at least one packet both to a second network and to a fast router.

33. A computer program product comprising;
    a. a memory medium, and
    b. a computer program, stored on said memory medium, said computer program comprising instructions for determining of the destination of at least one packet from said first digital network using one type of signalling and sending different signalling to a second network to establish a connection to said destination.

34. In a communication system having a wide area network interconnecting local area networks, a bandwidth reservation capability by which connections across said WAN can be reserved to guarantee availability when a connection is needed.

35. A method of billing, comprising the steps of:
    a. detecting an application type when a communications session is requested; and
    b. generating a billing record based on quality of service.

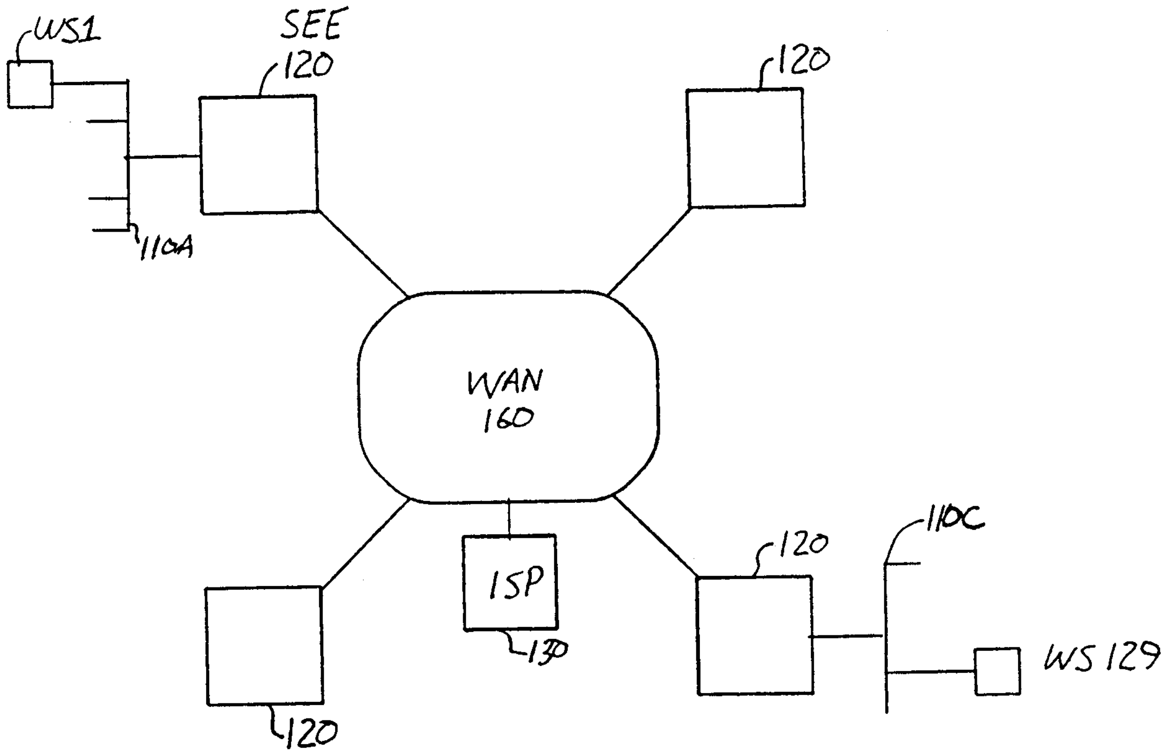


FIGURE 1

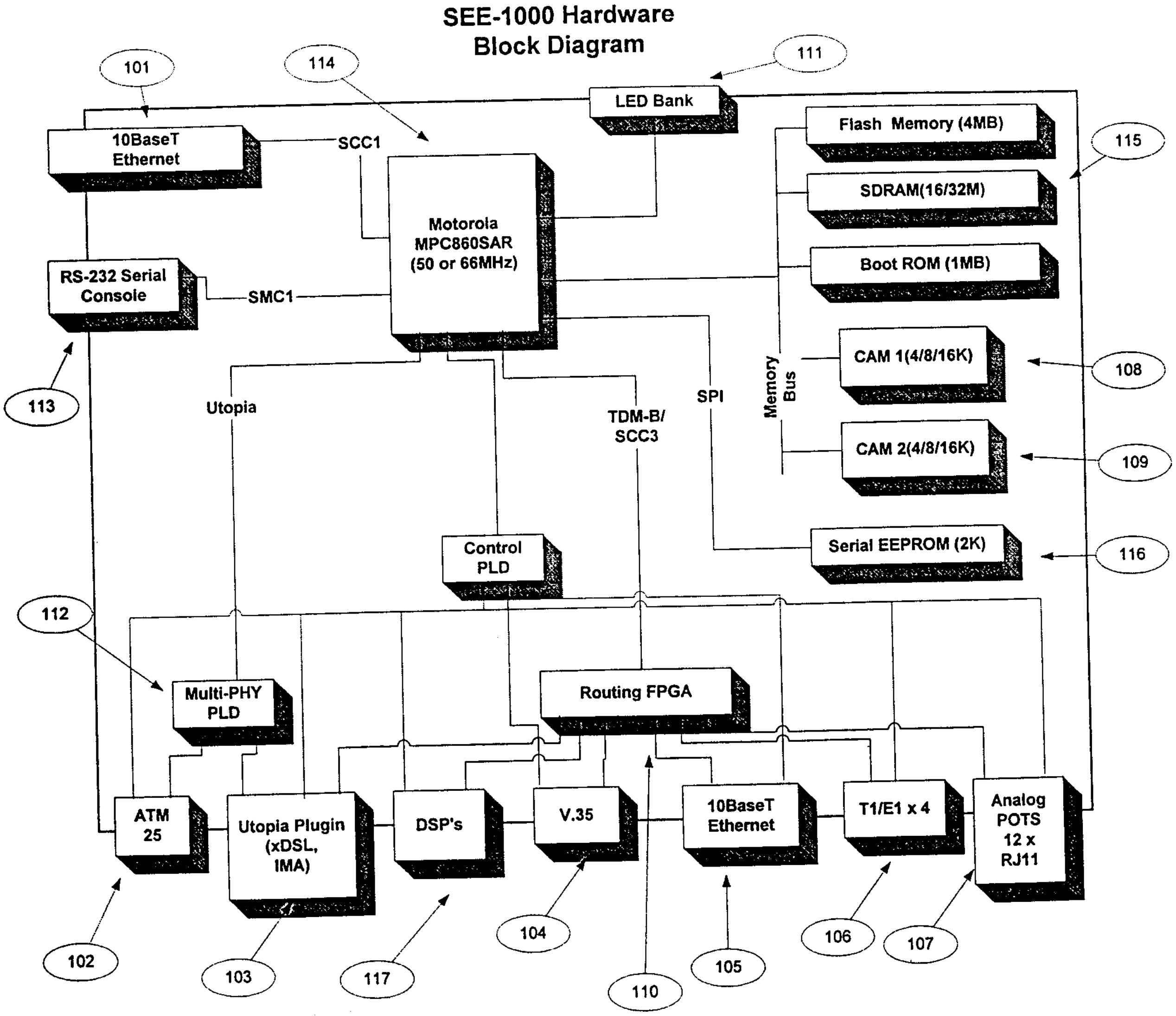
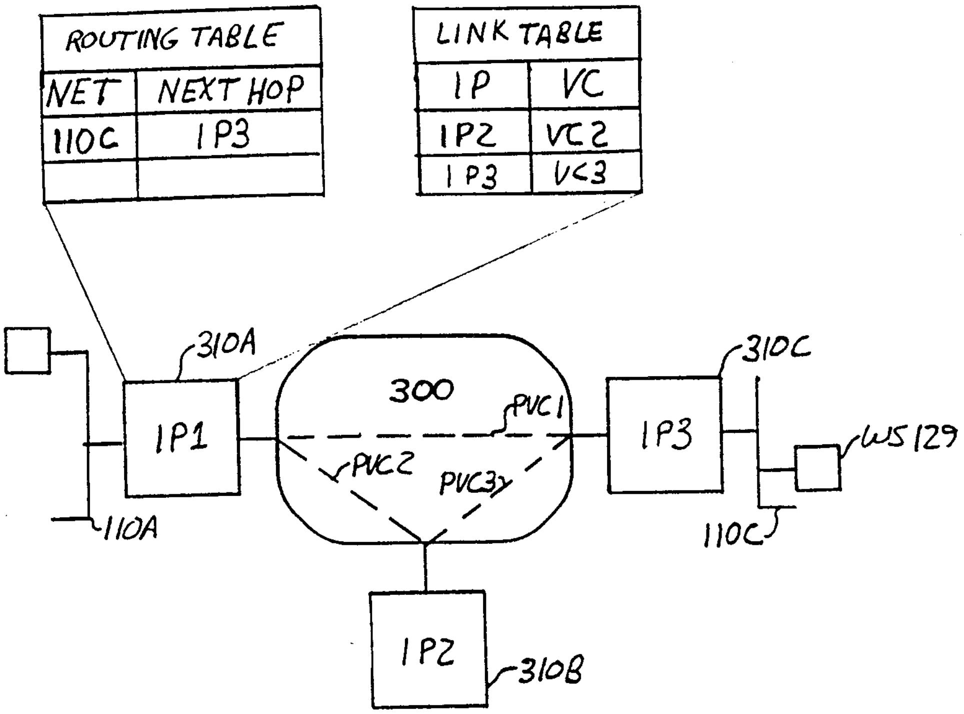

SEE-1000 Hardware
Block Diagram
101
114
111
LED Bank
10BaseT Ethernet
SCC1
Flash Memory (4MB)
115
SDRAM(16/32M)
Motorola MPC860SAR (50 or 66MHz)
RS-232 Serial Console
SMC1
Boot ROM (1MB)
CAM 1(4/8/16K)
108
113
Utopia
TDM-B/ SCC3
SPI
Memory Bus
CAM 2(4/8/16K)
109
Control PLD
Serial EEPROM (2K)
116
112
Multi-PHY PLD
Routing FPGA
ATM 25
Utopia Plugin (xDSL, IMA)
DSP's
V.35
10BaseT Ethernet
T1/E1 x 4
Analog POTS 12 x RJ11
102
103
117
104
110
105
106
107

FIGURE 2



| ROUTING TABLE | |
|---|---|
| NET | NEXT HOP |
| 110C | IP3 |
| | |

| LINK TABLE | |
|---|---|
| IP | VC |
| IP2 | VC2 |
| IP3 | VC3 |

FIGURE 3

| ROUTING | |
|---|---|
| NET | NEXT HOP |
| | |
| 110C | IP3 |

| LINK | |
|---|---|
| IP | VC |
| | |
| IP3 | VC123 |

| ATM ADDRESS | |
|---|---|
| IP | ATM ADDRESS |
| | |
| IP3 | ATM3 |

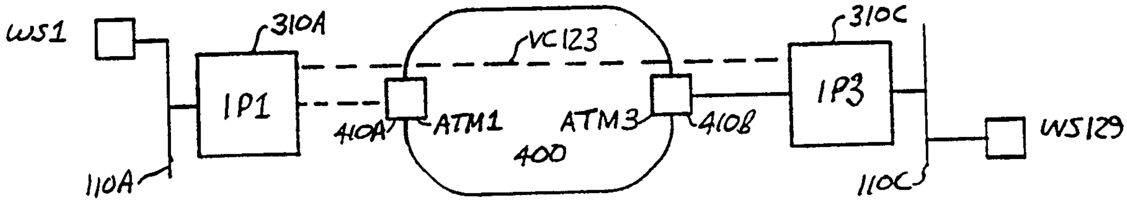


FIGURE 4

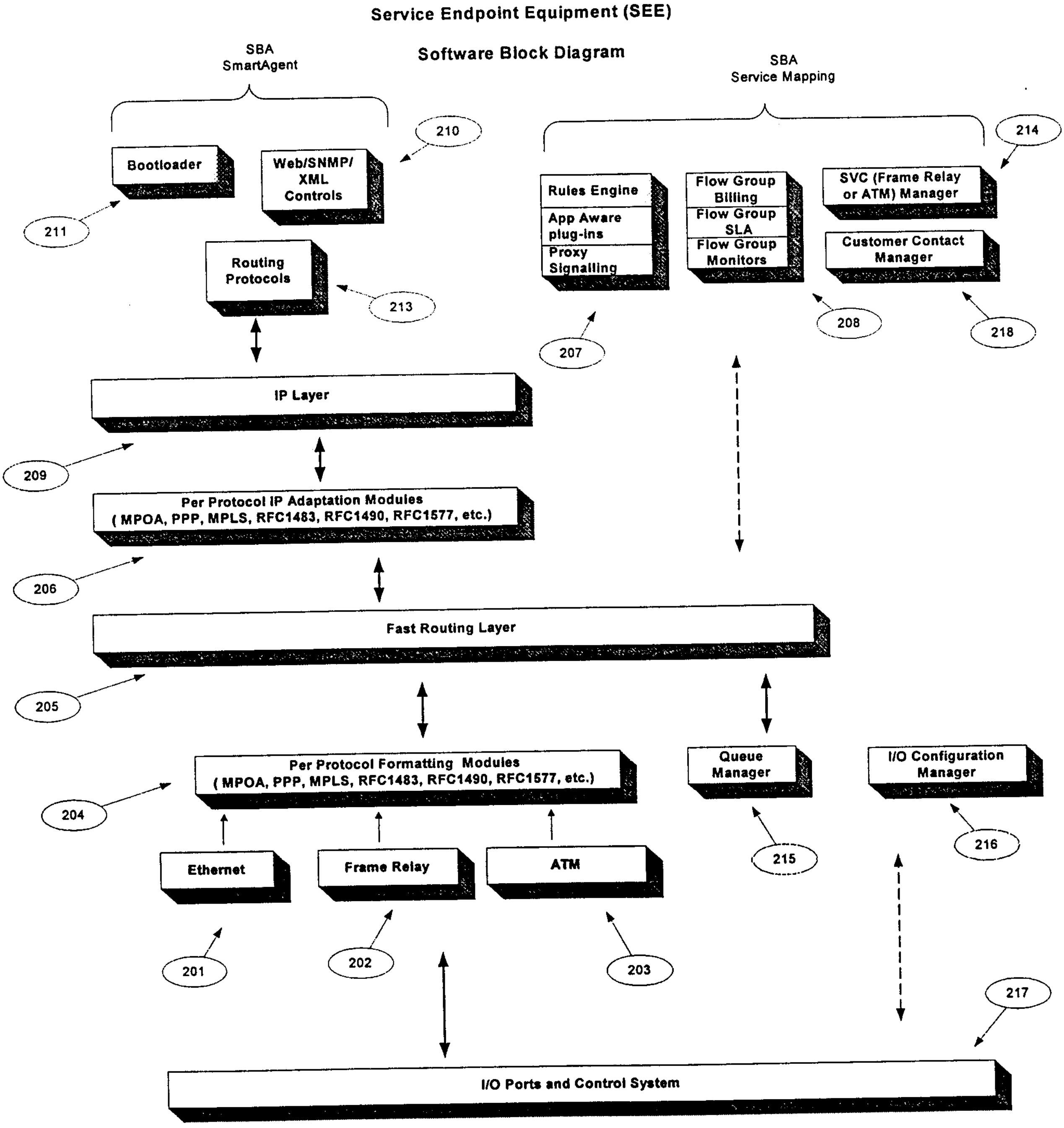

Service Endpoint Equipment (SEE)
Software Block Diagram
SBA SmartAgent
SBA Service Mapping
Bootloader
Web/SNMP/ XML Controls
Routing Protocols
Rules Engine
App Aware plug-ins
Proxy Signalling
Flow Group Billing
Flow Group SLA
Flow Group Monitors
SVC (Frame Relay or ATM) Manager
Customer Contact Manager
IP Layer
Per Protocol IP Adaptation Modules ( MPOA, PPP, MPLS, RFC1483, RFC1490, RFC1577, etc.)
Fast Routing Layer
Per Protocol Formatting Modules ( MPOA, PPP, MPLS, RFC1483, RFC1490, RFC1577, etc.)
Queue Manager
I/O Configuration Manager
Ethernet
Frame Relay
ATM
I/O Ports and Control System
201
202
203
204
205
206
207
208
209
210
211
213
214
215
216
217
218

FIGURE 5

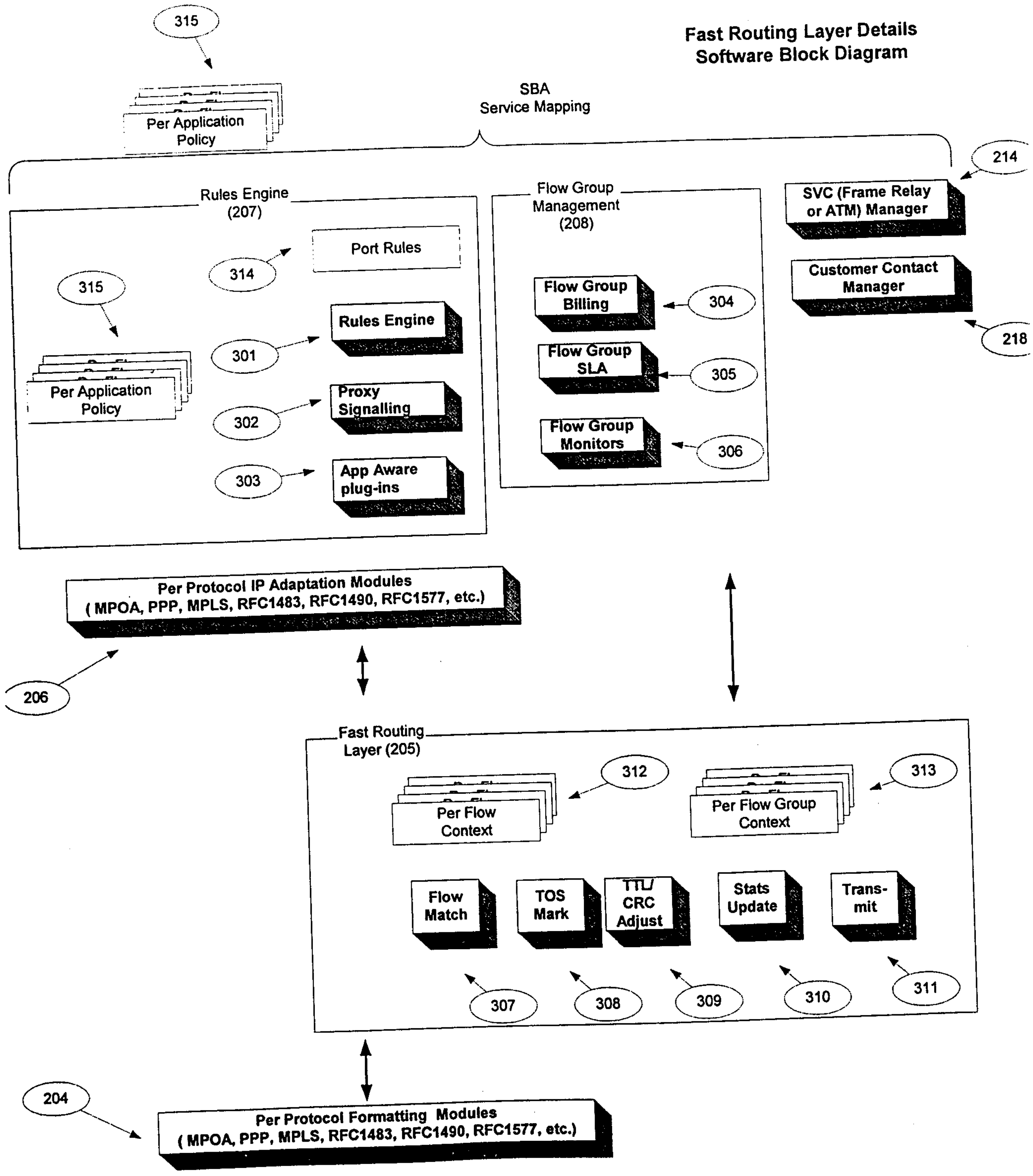

Fast Routing Layer Details
Software Block Diagram
315
Per Application Policy
SBA
Service Mapping
214
SVC (Frame Relay or ATM) Manager
Customer Contact Manager
218
Rules Engine (207)
Port Rules
314
315
Rules Engine
301
Per Application Policy
Proxy Signalling
302
App Aware plug-ins
303
Flow Group Management (208)
Flow Group Billing
304
Flow Group SLA
305
Flow Group Monitors
306
Per Protocol IP Adaptation Modules
( MPOA, PPP, MPLS, RFC1483, RFC1490, RFC1577, etc.)
206
Fast Routing Layer (205)
312
Per Flow Context
313
Per Flow Group Context
Flow Match
TOS Mark
TTL/ CRC Adjust
Stats Update
Trans-mit
307
308
309
310
311
204
Per Protocol Formatting Modules
( MPOA, PPP, MPLS, RFC1483, RFC1490, RFC1577, etc.)

FIGURE 6

## Fast Routing Layer Details Flow Chart

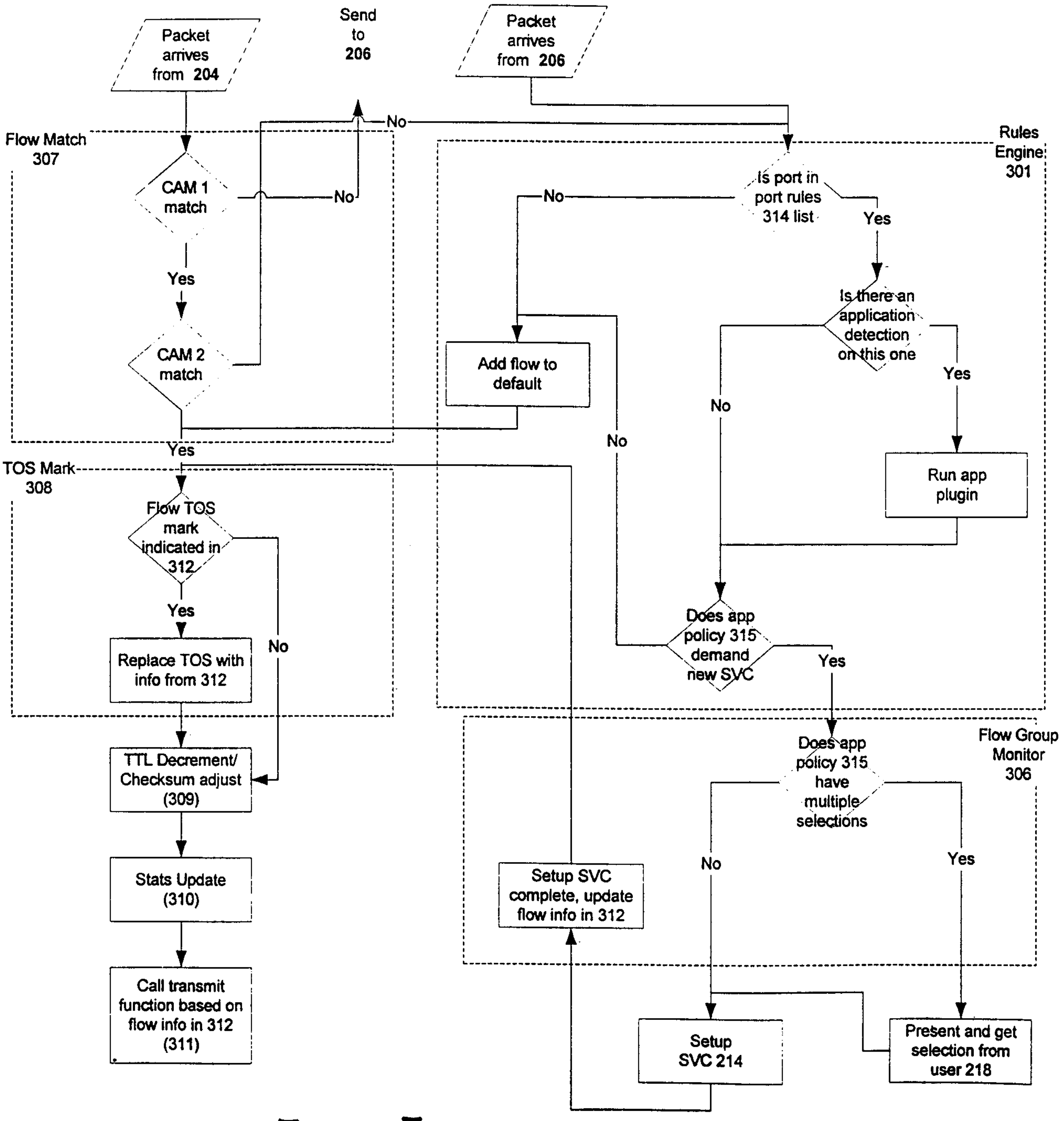
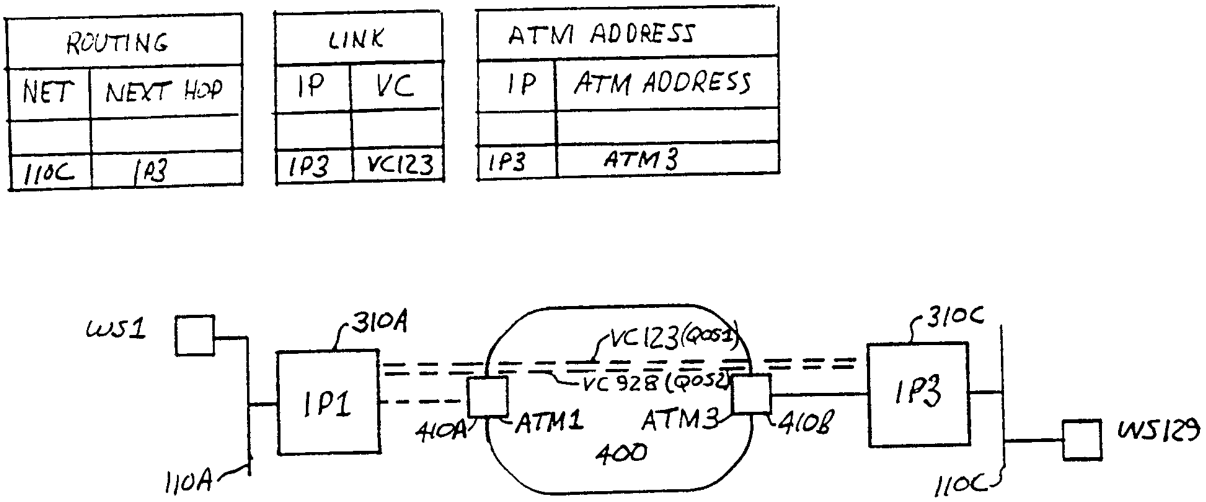

Packet arrives from 204
Send to 206
Packet arrives from 206
No
Flow Match 307
CAM 1 match
No
Yes
CAM 2 match
Yes
Rules Engine 301
Is port in port rules 314 list
No
Yes
Is there an application detection on this one
Add flow to default
No
Yes
No
Run app plugin
TOS Mark 308
Flow TOS mark indicated in 312
Yes
No
Replace TOS with info from 312
Does app policy 315 demand new SVC
Yes
TTL Decrement/ Checksum adjust (309)
Flow Group Monitor 306
Does app policy 315 have multiple selections
No
Yes
Stats Update (310)
Setup SVC complete, update flow info in 312
Call transmit function based on flow info in 312 (311)
Setup SVC 214
Present and get selection from user 218

FIGURE 7



| ROUTING | |
|---|---|
| NET | NEXT HOP |
| | |
| 110C | IP3 |

| LINK | |
|---|---|
| IP | VC |
| | |
| IP3 | VC123 |

| ATM ADDRESS | |
|---|---|
| IP | ATM ADDRESS |
| | |
| IP3 | ATM3 |

FIGURE 8

**LAN-WAN Bandwidth On Demand**
**(Unsignaled LAN)**
**Software Block Diagram**

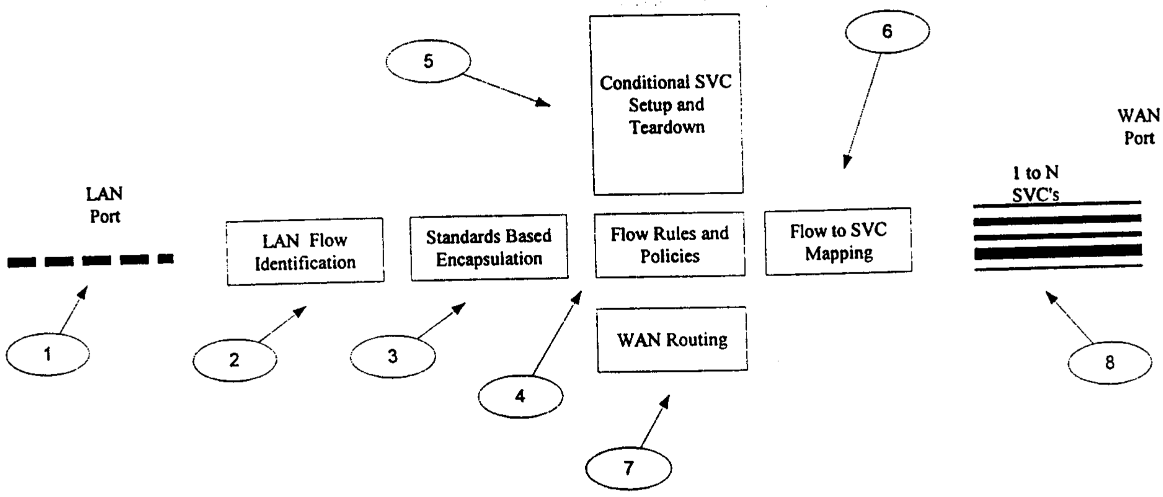


FIGURE 9

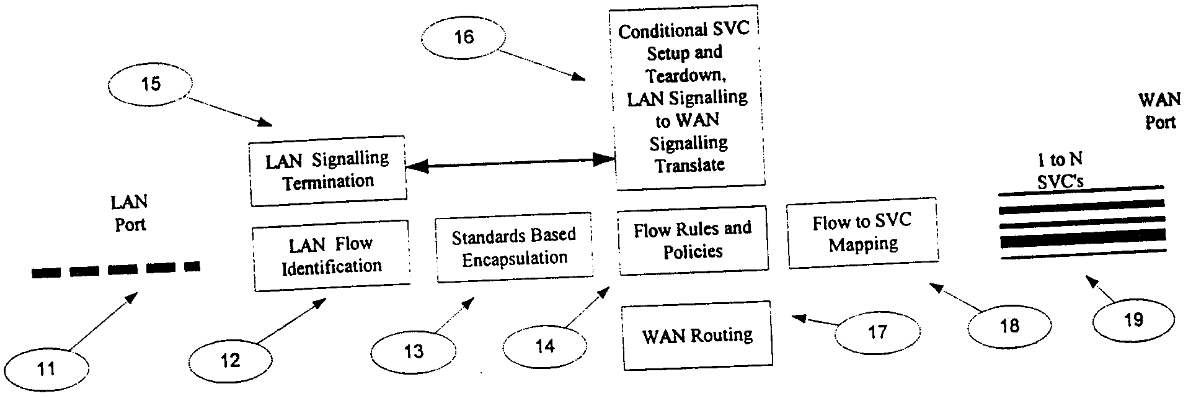

16
Conditional SVC Setup and Teardown, LAN Signalling to WAN Signalling Translate
15
WAN Port
LAN Signalling Termination
1 to N SVC's
LAN Port
LAN Flow Identification
Standards Based Encapsulation
Flow Rules and Policies
Flow to SVC Mapping
WAN Routing
11
12
13
14
17
18
19

FIGURE 10

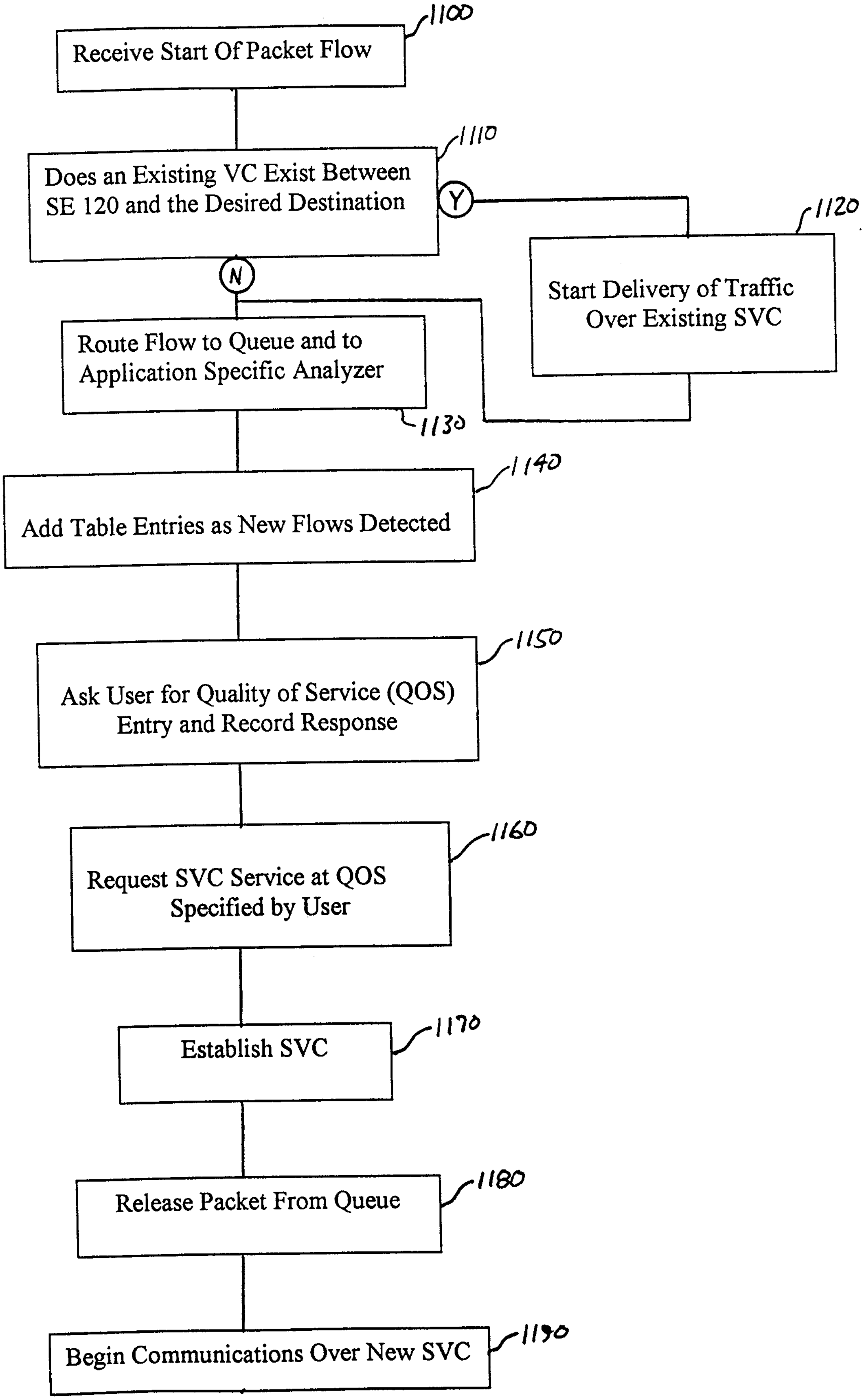

1100
Receive Start Of Packet Flow
1110
Does an Existing VC Exist Between SE 120 and the Desired Destination
Y
N
1120
Start Delivery of Traffic Over Existing SVC
Route Flow to Queue and to Application Specific Analyzer
1130
1140
Add Table Entries as New Flows Detected
1150
Ask User for Quality of Service (QOS) Entry and Record Response
1160
Request SVC Service at QOS Specified by User
1170
Establish SVC
1180
Release Packet From Queue
1190
Begin Communications Over New SVC

FIGURE 11

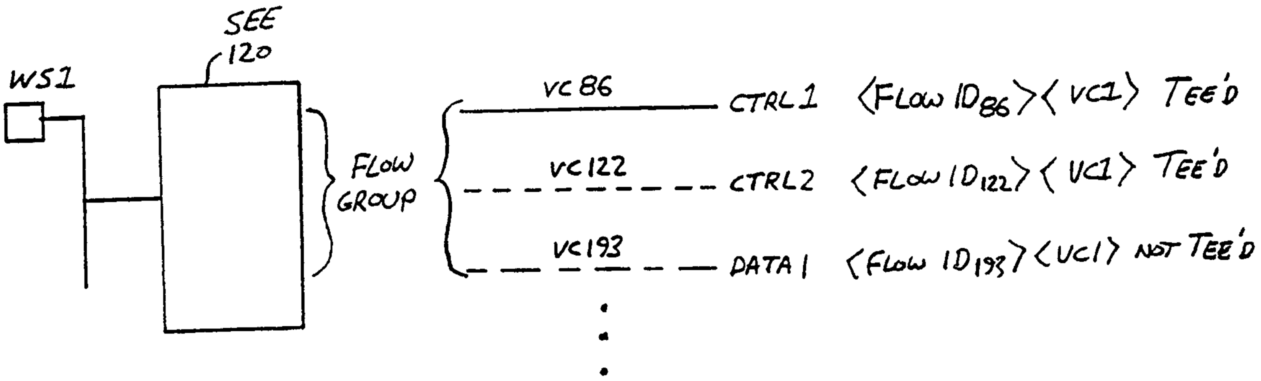
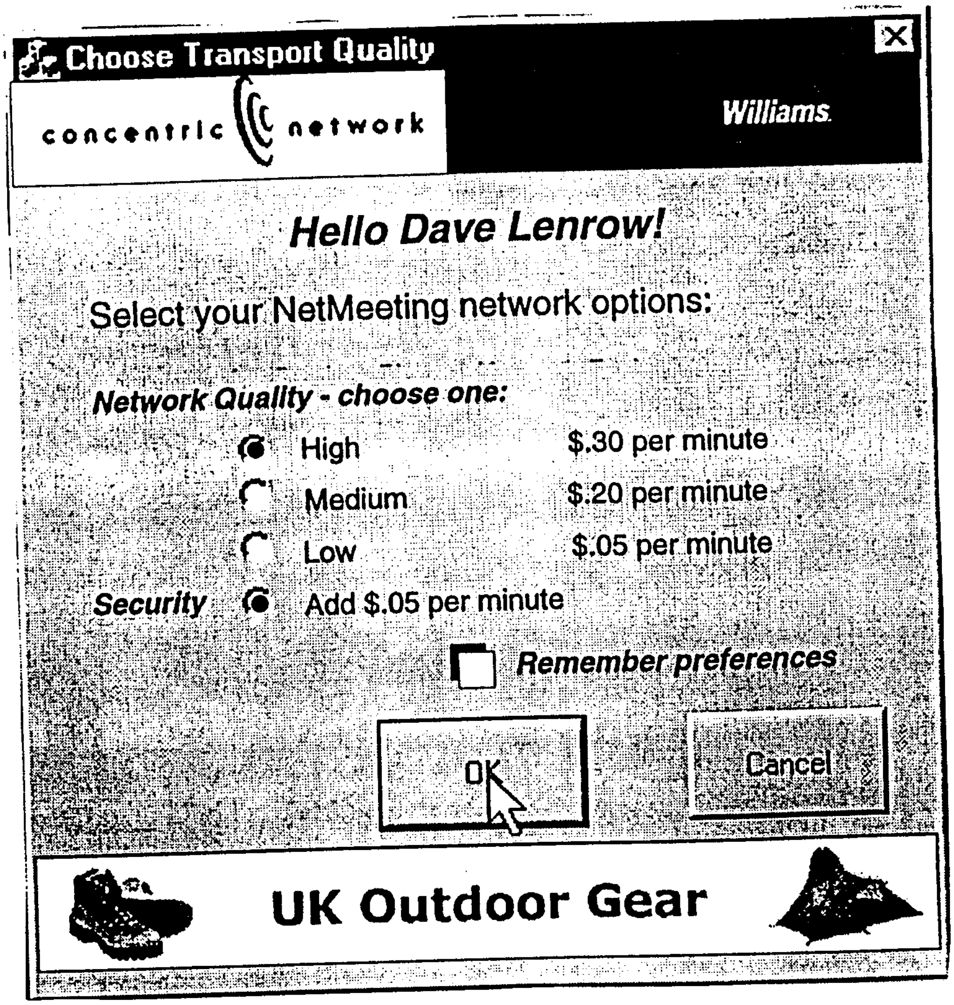

SEE
120
WS1
FLOW GROUP
VC86
VC122
VC193
CTRL1 <FLOW ID86><VC1> TEE'D
CTRL2 <FLOW ID122><VC1> TEE'D
DATA1 <FLOW ID193><VC1> NOT TEE'D

FIGURE 12

FLOW ID, < 5TUPLE >, SVC INFO <PORT NO.: VPI, VCI>, ENCAP HEADER <ABCDEF>

FIGURE 13



Choose Transport Quality

Hello Dave Lenrow!

Select your NetMeeting network options:

Network Quality - choose one:

|  |  |  |
|---|---|---|
| ● | High | $.30 per minute |
| ○ | Medium | $.20 per minute |
| ○ | Low | $.05 per minute |

Security ● Add $.05 per minute

□ Remember preferences

FIGURE 14

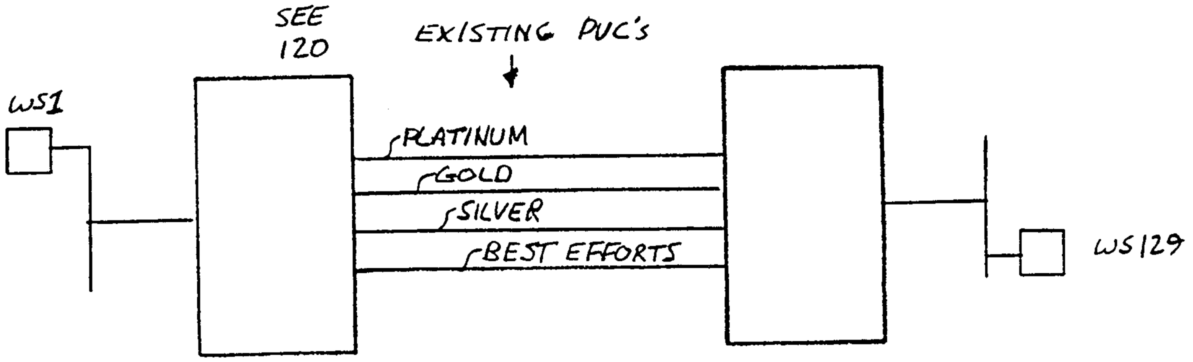

SEE
120
EXISTING PVC's
WS1
PLATINUM
GOLD
SILVER
BEST EFFORTS
WS 129

FIGURE 15

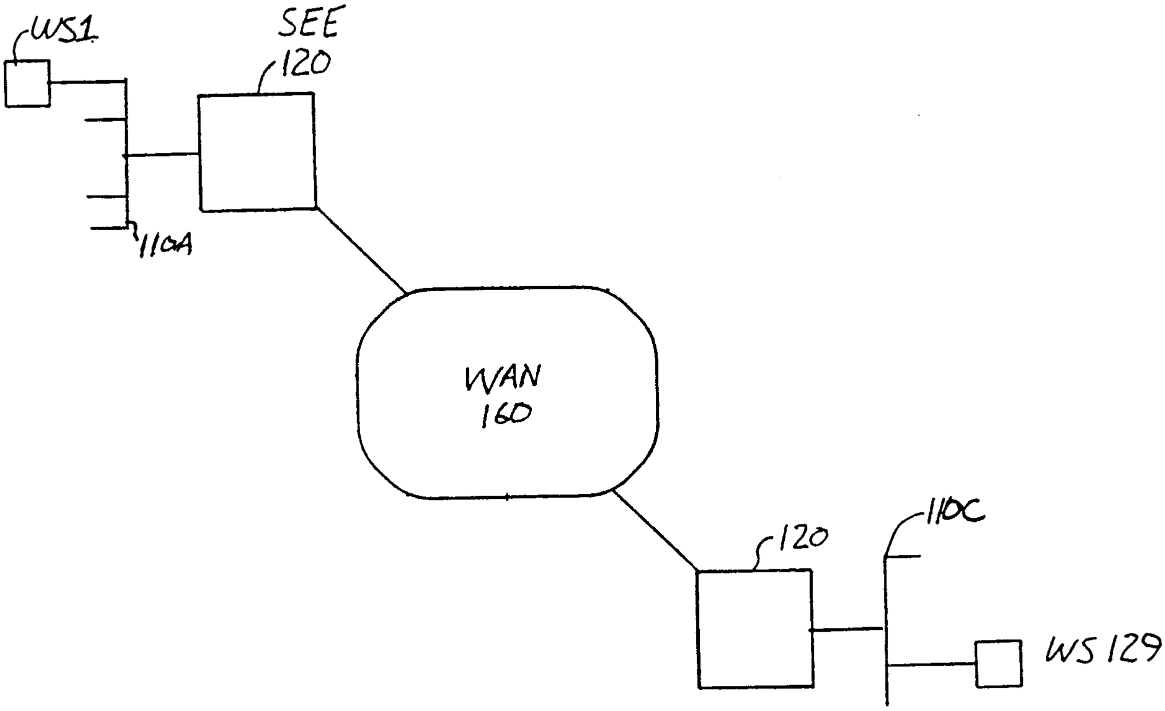


FIGURE 16

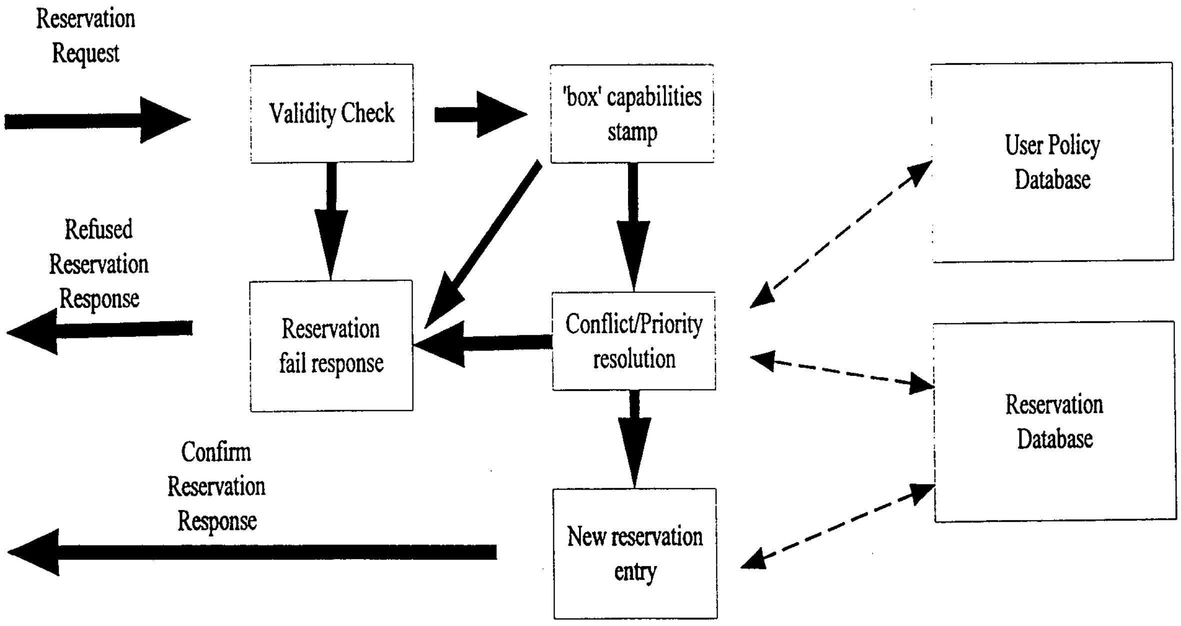

Reservation
Request
Validity Check
'box' capabilities
stamp
User Policy
Database
Refused
Reservation
Response
Reservation
fail response
Conflict/Priority
resolution
Reservation
Database
Confirm
Reservation
Response
New reservation
entry

FIGURE 17

```
Algorithm for evaluating new reservation requests
        Amount of resource commited = 0;
        For (all reservation records associated with desired resource)
        {
                If ( reservation overlaps time desired by new request)
                        Amount of resource commited += amount reserved by this
reservation
        }
        if ((Total amount of resouce - amount of resource committed)
                < amount being requested) Go To Attempt priority override
        else
        {
                commit new reservation
                acknowledge new reservation
                exit
        }
Attempt priority override:
        Overridable reservations = 0
        For (all reservations with overlapping time)
        {
                get priority of reservation holder
                if (priority of reservation holder is less than priority of requester)
                        overridable reservations += amount reserved by this reservations
        }
        if (overridable reservations < amount of resource requested)
        {
                refuse new reservation
                exit
        }
        else
        {
                overridden = 0;
                while  (overridden < amount of resource requested &&
                        more reservations with overlapping time)
                {
                        get priority of reservation holder
                        if (priority of reservation holder is less than priority
                                of requester)
                        {
                                Free reservation and notify ex-holder;
                                Overriden += amount reserved by this reservations
                        }
                }
                send notice accepting new reservation
                exit
```

FIGURE 18

## 4.2.1. Pseudo data structures 312 and 313

Flow structure **312**

```
--------------
int          flowId
int          flowGroupId
int          active
int          creationTick
int          lastTxTick
int          agingCounter
uns32  localIpAddress
uns32  remoteIpAddress
uns16  camFirstAddress
uns8   cam2Protocol
uns8   cam2IpTos
uns16  localTcpUdpPort
uns16  remoteTcpUdpPort
pUpperRxRoutine - pointer to upper half Rx function
pUpperTxRoutine - pointer to upper half Tx function
pLowerRxRoutine - pointer to lower half Rx function
pLowerTxRoutine - pointer to lower half Tx function
pDriverVcContext - pointer to SAR driver VC context (pChan)
pHdlcCb - pointer to HDLC control block structure
pNativeIfcb - pointer to IFCB structure
pConnectionId - pointer to VPI/VCI structure
int          dlci
int          cpcsLp
int          cpcsCi
int          cpcsUu

pIbodFlowTable - pointer to IBOD flow table structure
pIbodSession - pointer to IBOD session structure
pIbodRuleCriteria - pointer to IBOD rule criteria structure
string  ibodFlowName
int          isDefault
ibodPacketType - instance of IBOD packet type structure
ibodSessionType - instance of IBOD session type structure
int          forceIpoaPackets
int          matches
int          txMatches
int          rxMatches

statistics:
int          txPackets
int          txBytes
int          txErrors
int          rxPackets
int          rxBytes
int          rxErrors

pNextFlow - pointer to flow structure
pPrevFlow - pointer to flow structure
[end of flow structure]
```

FIGURE 19A

Flow group structure **313**

```
--------------------
int             flowGroupId
int             active
int             creationTick
int             flowCount
pHeadFlow - pointer to flow structure
pTailFlow - pointer to flow structure

pIbodInstance - pointer to IBOD instance structure
pIbodSessionTableStruct - pointer to IBOD session table structure
pIbodRoute - pointer to IBOD route structure
pIbodProfile - pointer to IBOD profile structure
pIbodRule - pointer to IBOD rule structure
pIpoaCctxt - pointer to IPOA channel context structure
pIbodVcStruct - pointer to IBOD VC structure
int             uiClientId
int             qosRequestIsDone
int             qosRequestResult
ibodPacketType - instance of IBOD packet type structure
ibodSessionType - instance of IBOD session type structure
string   ibodSessionName
ibodSessionState - instance of IBOD session state structure
isDefault
forceIpoaPackets
billingStructure

pNextFlowGroup - pointer to flow group structure
pPrevFlowGroup - pointer to flow group structure
[end of flow group structure]
```

FIGURE 19B

## 4.2.2. Pseudo code for upper IP drivers 206

```
Part 1 - per-protocol IP adaptation module (upper half of driver) 206
-------------------------------------------------------------

Ethernet transmit side
---------------------
enetUpperTx(pUnit)
if there are packets to send and the lower half has available
buffers, then dequeue a packet
ifDequeue(pUnit->sendQueue, &pMbuf)

copy from IP layer buffers (mbufs) into lower layer buffers (USRBUFs)
copy_from_mbufs(pUsrbuf->pUsrdata, pMbuf, &length)

call the lower half of the Ethernet transmit routine 204
enetLowerTx(pUsrbuf)
return
------

non-Ethernet transmit side
-------------------------
nonEnetUpperTx(pUnit, pMbufs)
copy from IP layer buffers (mbufs) into lower layer buffers (USRBUFs)
copy_from_mbufs(pUsrbuf->pUsrdata, pMbuf, &length)

call the lower half of the non-Ethernet transmit routine 204
nonEnetLowerTx(pUsrbuf)
return
------

Ethernet receive side
------------------
enetUpperRx(pUnit, pIpPacket)
copy from lower layer buffers (USRBUFs) into IP layer buffers (mbufs)
pMbuf = copy_to_mbufs(pUsrbuf->pUsrdata, length, 0, pUnit->ac_if)

pass up the IP protocol stack 209
do_protocol_with_type(IPtype, pMbuf, pUnit->idr, length)
increment input packet counter
return
------

non-Ethernet receive side
----------------------
nonEnetUpperRx(pUnit, pIpPacket)
copy from lower layer buffers (USRBUFs) into IP layer buffers (mbufs)
pMbuf = copy_to_mbufs(pUsrbuf->pUsrdata, length, 0, pUnit->ac_if)

pass up the IP protocol stack 209
do_protocol_with_type(IPtype, pMbuf, pUnit->idr, length)
increment input packet counter
return
------
```

FIGURE 20

## 4.2.3. Pseudo code for Fast Routing Layer 205

```
Fast Routing Layer 205
--------------------------

FRL transmit side
----------------
frlTx(pIpPacket, pLowerTxRoutine)
skip local traffic
if (sourceIpAddress != localIpAddress)
          assume new flow and new flow group (because it must be an earlier 'miss'
                    which has come through the IP routing 213)
          create new flow 312
          create new flow group 313
          call rules engine (since we know it's a new flow) 301
call given lower transmit routine 311
(*pLowerTxRoutine)(pIpPacket)
return
------

FRL receive side
---------------
frlRx(ENTERPRISE/WAN, pIpPacket, pUpperRxRoutine, pLowerTxRoutine)
check CAM for a layer 3 miss 307
camCompare3(pIpPacket->localIpAddress, pIpPacket->remoteIpAddress)
if layer 3 miss
          call per-protocol IP adaptation module 206
          (*pUpperRxRoutine)(pUnit, pIpPacket)
check CAM for a layer 4 miss 307
camCompare4(firstCompare, pIpPacket->localTcpUdpPort,
                              pIpPacket->remoteTcpUdpPort, pIpPacket->ipProtocol,
                              pIpPacket->ipTypeOfService)
if layer 4 miss
          call rules engine 301
else
          call per-protocol formatting module 204
          (*pLowerTxRoutine)(pUnit, pIpPacket)
          if flow is marked 312 as a 'always rule check' (i.e. it's a control channel we
                              need to continuously monitor)
                    call rules engine 301
endif
return
------
```

FIGURE 21

## 4.2.4. Pseudo for lower half of IP drivers 204

Per-protocol formatting module (lower half of driver) **204**

```
----------------------------------------------------------

non-Ethernet transmit side
--------------------------
nonEnetLowerTx(pFlowContext, pUsrbuf)
prepend the appropriate encapsulation header
call SAR / HDLC driver for actual packet transmission 202/203
m8xxSarHjSend(pFlowContext->pDriverVcContext, pFlowContext->pNativeIfcb,
                              pFlowContext->pConnectionId, pUsrbuf, pFlowContext->cpcsLp,
                              pFlowContext->cpcsCi, pFlowContext->cpcsUu)
  - OR -
m8xxHdlcSend(pFlowContext->pHdlcCb, dlci, pUsrbuf, 0)
return
------

Ethernet transmit side
----------------------
enetLowerTx(pUsrbuf) 201/204
set data length from pUsrbuf in the Tx buffer descriptor (BD)
set ready bit in the Tx BD
increment the next Tx BD count
return
------

non-Ethernet receive side
-------------------------
nonEnetLowerRx(pFlowContext, pIpPacket/pUsrbuf) 204
if (pIpPacket->destIpAddress == localIpAddress)
        pass local traffic up to the upper half
        nonEnetUpperRx(pUsrbuf/pIpPacket)
else
        call FRL receive routine
        frlRx(WAN, pIpPacket, pFlowContext->pUpperRxRoutine,
                        pFlowContext->pLowerTxRoutine)
return
------

Ethernet receive side
---------------------
enetLowerRx(pUnit, pEth802.3Packet, length, pFlowContext)
check for non-IP traffic
if (pEth802.3Packet->type != ETHERTYPE_IP)
        copy USRBUF to mbuf and send received packet up the appropriate
                non-IP protocol stack
        pMbuf = copy_to_mbufs(pUsrbuf->pUsrdata, length, 0, pUnit->ac_if)
        do_protocol_with_type(pEth802.3Packet->type, pMbuf, pUnit->idr, length) 209
        increment input packet counter
check for local IP traffic
if (pIpPacket->destIpAddress == localIpAddress)
        pass local traffic up to the upper half 206
        enetUpperRx(pUsrbuf/pIpPacket)
else
```

FIGURE 22A

```
        call FRL receive routine 205
        frlRx(ENTERPRISE, pIpPacket, pFlowContext->pUpperRxRoutine,
                        pFlowContext->pLowerTxRoutine)
return
------
```

FIGURE 22B

## 4.2.5. Pseudo code for Rules Engine 301

```
Rules Engine **301**
------------
if there was a layer 3 miss, then a new flow and flow group is required
        Create new flow group - send a message to the flow group state machine manager
            (FGSMM) 306
if there was a layer 3 hit but a layer 4 miss, then a new flow is required
        Create new flow 312.
        If part of existing group (determined by per port number rules table 314 and App Aware
              plugins 303)
                add to existing flow group 313 - [in line]
        else
                Create new flow group - send a message to the flow group state machine
                    manager (FGSMM) 306
if a flow is being terminated, remove the flow from its existing flow group
        Delete old flow 312 and remove from existing flow group 313 - [in line]
        if the flow just removed was the last flow in its flow group, then
        the flow group must be terminated - send a message to the flow group state machine
        manager (FGSMM) 306
```

FIGURE 23

## 4.2.6. Pseudo code for Flow Group Monitor 306

Flow Group Monitor **306** - Flow Group State Machine Manager (FGSMM)

```
----------------------------------------
flowGroupStateMachineManager(flowGroup, event)
switch (flowGroup->state)
        case CLOSED:
                Create a new flow group 313
                Send a message to the Signalling entity 214
                flowGroup->state = WAITING_FOR_SIGNALLING_RESPONSE_1

        case WAITING_FOR_SIGNALLING_RESPONSE_1:
                if (event->signalling_response_1 == OK)
                        fill in the rest of the flow group information 313
                        flowGroup->state = OPEN
                else
                        send an error to someone
                        flowGroup->state = CLOSED
                endif

        case OPEN:
                if (event->terminate_flow_group == YES)
                        if (FlowGroup->using_SVC == YES)
                                Tear down the flow group's SVC(s)
                                Send a message to the Signalling entity 214
                                flowGroup->state =
                                    WAITING_FOR_SIGNALLING_RESPONSE_2
                        else
                                flowGroup->state = CLOSE_FLOWS
                        endif
                else
                        send an error to someone
                        flowGroup->state = UNKNOWN
                endif

        case WAITING_FOR_SIGNALLING_RESPONSE_2:
                if (event->signalling_response_2 == OK)
                        ready to close the flows
                        flowGroup->state = CLOSE_FLOWS
                else
                        send an error to someone
                        flowGroup->state = CLOSED
                endif

        case CLOSE_FLOWS:
                close flows and send billing information to 304
                send SLA information to 305
                flowGroup->state = AFTER_CLOSE_FLOWS

        case AFTER_CLOSE_FLOWS:
                delete CAM entries for all closed flows 312
                delete flow group
                flowGroup->state = CLOSED
```

FIGURE 24

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/22773

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 12/28, 12/46, 12/66
US CL :370/404, 410, 467
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. : 370/404, 410, 467, 395, 400, 401, 468, 522

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)

WEST
search terms: ATM, frame relay, Ethernet, TCP/IP, QOS, signaling

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A,P | US 5,903,559 A (ACHARYA ET AL.) 11 May 1999, see entire document. | 1-35 |
| A,P | US 5,923,659 A (CURRY ET AL.) 13 July 1999, see entire document. | 1-35 |

□ 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 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<br>05 DECEMBER 1999 | Date of mailing of the international search report<br>17 DEC 1999 |
| Name and mailing address of the ISA/US<br>Commissioner of Patents and Trademarks<br>Box PCT<br>Washington, D.C. 20231<br>Facsimile No. (703) 305-3230 | Authorized officer<br>MIN JUNG James R. Matthews<br>Telephone No. (703) 305-4363 |