Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2698082 A1 2010/09/30

(21) 2 698 082

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2010/03/30

(41) Mise à la disp. pub./Open to Public Insp.: 2010/09/30

(30) Priorités/Priorities: 2009/03/31 (US12/415,594); 2010/03/26 (US12/732,344)

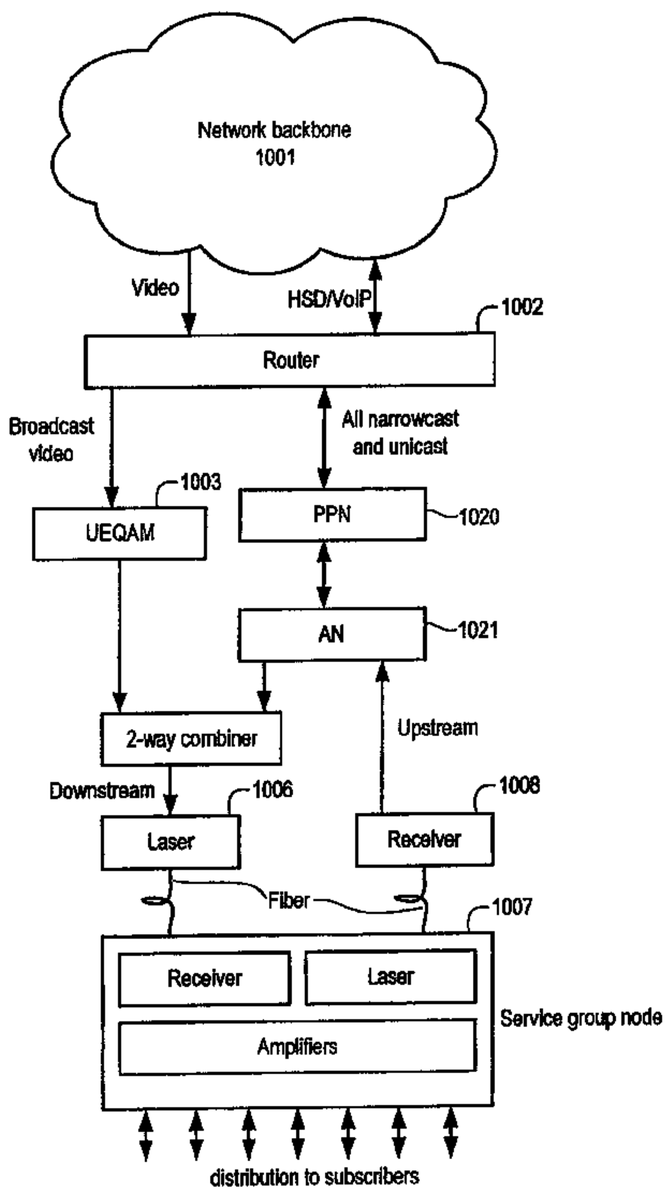
(51) Cl.Int./Int.Cl. *H04L 12/66* (2006.01), *H04L 12/26* (2006.01)

(71) Demandeur/Applicant:
COMCAST CABLE COMMUNICATIONS, LLC, US

(72) Inventeurs/Inventors:
SALINGER, JORGE DANIEL, US;
URBAN, DAVID, US;
SAXENA, VIK, US

(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : ARCHITECTURE DE RESEAU D'ACCES COMPORTANT DES SOUS-RESEAUX A ACCES DISSEMBLABLE
(54) Title: ACCESS NETWORK ARCHITECTURE HAVING DISSIMILAR ACCESS SUB-NETWORKS



(57) Abrégé/Abstract:
A subscriber network architecture includes a packet processing node that communicates with a network backbone and with a plurality of access nodes. The access nodes communicate with a plurality of user devices. The packet processing node controls Quality of Service (QoS) levels for the subscriber devices, routes user data to devices via the access nodes and receives data from devices via the access nodes for forwarding across the network backbone.

ABSTRACT

A subscriber network architecture includes a packet processing node that communicates with a network backbone and with a plurality of access nodes. The access nodes communicate with a plurality of user devices. The packet processing node controls Quality of Service (QoS) levels for the subscriber devices, routes user data to devices via the access nodes and receives data from devices via the access nodes for forwarding across the network backbone.

ACCESS NETWORK ARCHITECTURE HAVING DISSIMILAR ACCESS SUB-NETWORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

[01] This application is a continuation in part of U.S. Patent Application Serial No. 12/415,594, filed March 31, 2009, and titled "Subscriber Access Network Architecture," which application in its entirety is incorporated by reference herein.

BACKGROUND

[02] High speed data service has become a ubiquitous part of modern life, and the availability of such service is of ever-increasing importance. Typically, numerous data service users or subscribers send and receive data through a service provider. The subscribers may be individual homes or businesses and the service provider may be a separate entity, though this need not be the case. Subscriber data service is often provided over a physical medium that is also used to provide other types of service. One well-known example is the provision of data service over a coaxial cable that is also used to provide cable television (CATV) service. Other examples include provision of such services over optical networks, wireless networks, or hybrid networks. In many systems, a first portion of the spectrum is used for television service, a second portion is used for upstream data transmissions from subscribers to a provider's service location, and a third portion used for downstream data communications from the provider to the subscribers or users. The communicated data may include emails, communications to and from the Internet, voice over IP (VoIP) telephone service, video on demand (VOD), etc.

[03] Existing subscriber network architectures pose ongoing challenges. In the example of a cable network, combined CATV and subscriber data systems are often designed to comply with the Data over Cable Service Interface Specifications (DOCSIS) group of standards promulgated by Cable Television Laboratories, Inc. A DOCSIS head end may include a cable modem termination system (CMTS) that sends and receives communications to individual subscribers. Under later versions of the DOCSIS standards, the CMTS functions

may be divided between a modular CMTS (M-CMTS) core and one or more Edge Quadrature Amplitude Modulation (EQAM) devices. Because of the manner in which functions are divided between an M-CMTS core and EQAMs, the separation between these devices is usually limited to several hundred feet. This can significantly constrain design of cable systems. Other types of transmission systems also present challenges.

[04] The relative expense of an M-CMTS core relative to an EQAM and the distribution of functions between an M-CMTS core and EQAM can inhibit system scalability. In particular, adding additional system capacity requires adding additional M-CMTS cores and additional EQAMs. The additional M-CMTS cores are needed so as to accommodate increased demand for certain functionality, but other functionality of the additional M-CMTS core(s) may not be needed. Other scalability, operational and environment issues can also arise. Using known architectures, additional equipment for increasing QAM channels could result in large space requirements, considerable increases in heating and cooling requirements, and a significant increase in monitoring, management and support requirements.

[05] FIG. 1 is a block diagram generally showing an example of a network architecture employed by service providers. The various blocks in FIG. 1 correspond to categories of network elements, and the arrows connecting those blocks indicate flows of data between those network element categories. For example, data corresponding to services is received from and sent to one or more backbone IP networks 1001 by routers represented by block 1002. Service data includes both broadcast data (e.g., television and network programming), narrowcast data (e.g., VOD and switched digital video (SDV) programming) and unicast data (e.g., television and high speed data (HSD) service providing Internet connectivity to individual subscribers and VoIP or other type of telephone service). The backbone network may be, e.g., a system operator's national IP network, the Internet, some combination of the Internet and a system operator's network etc. Typically, several layers of routers (e.g., at the national, regional and local levels) are part of block 1002. Broadcast and narrowcast data is routed to universal edge QAM (quadrature amplitude modulation) devices (UEQAMs) that are typically located in distribution hubs, which devices are represented in FIG. 1 by block

1003. Unicast data is routed to and from a termination system core, such as a cable modem termination system (CMTS), 1004, with those CMTS cores also typically located in distribution hubs. Downstream unicast data is sent from CMTS cores to UEQAMs. The UEQAMs then modulate the broadcast, narrowcast and unicast downstream data into RF frequency channels that are combined (block 1005) and communicated to lasers 1006 for fiber optic transmission to individual service group nodes (block 1007). Those nodes convert the downstream optically-transmitted signals to electrical signals for distribution over coaxial cables, or other types of transmission lines, to subscriber devices such as modems, set top boxes (STBs), media terminal adapters (MTAs) end user devices, etc. Upstream transmissions from subscribers are received at nodes 1007, converted to optical signals and forwarded to CMTS cores, where those optical signals are converted to electrical signals and further processed.

[06] The architecture of FIG. 1 was designed to support a few narrowcast and/or unicast channels in the presence of a large proportion of broadcast channels, both analog and digital. Such an architecture is optimized for combining at the RF layer, i.e., combining signals from (or to) many RF Ports. Each signal may contain an analog channel, a digital broadcast video multiplex QAM, or a small number of High Speed Data (HSD) channels. Architectures such as are shown in FIG. 1 can be unduly restrictive and/or pose problems if a cable system operator wishes to change the mix of available services. Other architectures may present similar problems.

SUMMARY

[07] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the invention.

[08] According to some embodiments, a network architecture includes a packet processing node that communicates with a network backbone and with a plurality of access nodes in the domain of the packet processing node. Each access node communicates with subscriber

devices located in multiple subscriber premises within an access sub-network served by the access node. The packet processing node can manage subscriber information, control Quality of Service (QoS) levels for the subscriber devices, forward data to subscriber devices via the access nodes based on service flows, and receive data from subscriber devices via the access nodes for forwarding across the network backbone. Access nodes within a single packet processing node domain may utilize different transport technologies and protocols. For example, one or more access nodes could employ hybrid fiber coax (HFC) transport technologies and protocols, one or more additional access nodes could employ passive optical network (PON) transport technologies and protocols, and one or more other access nodes could employ wireless communication transport technologies and protocols.

BRIEF DESCRIPTION OF THE DRAWINGS

[09] FIG. 1 is a block diagram generally showing a currently employed network architecture.

[10] FIG. 2 is a block diagram showing a portion of a network architecture that includes a subscriber access network according to at least some embodiments.

[11] FIG. 3 is a diagram of a network according to some embodiments.

[12] FIG. 4 is a functional block diagram of a packet processing node according to some embodiments.

[13] FIG. 5 is a functional block diagram of an access node according to some embodiments.

[14] FIGS. 6-9 are flow charts describing operations performed by the packet processing node of FIG. 4 according to some embodiments.

[15] FIGS. 10-14 are flow charts describing operations performed by the access node of FIG. 5 according to some embodiments.

[16] FIG. 15 is a block diagram of a packet processing node according to at least some embodiments.

[17] FIG. 16 is a block diagram of an access node according to at least some embodiments.

[18] FIG. 17 is a diagram of a network, according to another embodiment, having multiple access nodes of different types serving individual access sub-networks.

[19] FIG. 18 is a block diagram showing hardware for the packet processing node of FIG. 17 according to at least some embodiments.

[20] FIGS. 19 through 21 are block diagrams showing hardware for access nodes in the network of FIG. 17 according to at least some embodiments.

[21] FIG. 22 is a flow chart showing operations performed by the packet processing node of FIG. 17, in at least some embodiments, upon receiving service data that is destined for one or more subscriber premises.

[22] FIG. 23 is a flow chart showing operations performed in at least some embodiments by an access node in FIG. 17 in response to receiving service data forwarded by the packet processing node in FIG. 17.

[23] FIG. 24 shows another embodiment of a network having different types of access nodes.

DETAILED DESCRIPTION

[24] As indicated above, architectures such as are shown in FIG. 1 may prove unduly restrictive in many circumstances. For example, many operators are broadly deploying

digital programming and increasing narrowcast services. Such services include increased VOD, deployment of higher bandwidth HSD, such as DOCSIS 3.0-based HSD, bandwidth efficient SDV, and new services (e.g., "Start Over" services that permit a subscriber to restart a television program that has already begun). These factors result in a significantly growing need for additional QAM channels and QAM modulators in the operators' access networks. As explained in more detail below, upstream and downstream communications to and from subscriber or user (used interchangeably herein) premises devices (e.g., cable modems, set top boxes, media terminal adapters) occur in QAM-modulated RF sub-bands. An access network is a portion of a system connecting subscribers with their immediate service provider and includes the QAM devices used to modulate the signals sent to subscribers and the QAM devices used to demodulate the signals sent by subscribers. In the example of FIG. 1, for example, access network portions of the architecture include UEQAM block 1003, CMTS core block 1004, and other elements between those blocks and subscriber devices.

[25] . Narrowcast QAM channels can be re-used in different service groups of a particular hub. The need for additional QAM channels and QAM devices (e.g., modulators and demodulators) is thus compounded when there are more service groups in a particular hub or optical transport network (OTN) and/or if the number of such hubs or OTNs is increased. Indeed, many system operators are reducing the size of service groups in their networks. This change is motivated by operational benefits, including more reliable operation and improved service performance. The combined effect of these changes in a conventional access network architecture will result in the addition of a very large number of Edge QAM and CMTS ports, which will present operational and financial challenges to the system operators. To successfully deploy additional narrowcast services, a system operator may need to adapt its current network to support the deployment of many more QAM and termination system ports than is currently the case. The environmental requirements (e.g., equipment room space, power requirements, heat dissipation) to support this growth are a significant challenge for a system operator. Increased use of DOCSIS channels, for example, to support bonded services and/or deployment of IP video simulcast will also require a significant growth in CMTS capacity, both downstream and upstream channels, further

increasing the number of QAM channels and devices for modular CMTS (M-CMTS) implementations. Narrowcast and unicast QAM channels will consume a larger portion of the available RF spectrum.

[26] Increasing the number of narrowcast and unicast QAM channels also increases operational management complexity. Each additional QAM device or other network component requires individual configuration and ongoing management. Component failures require experienced staff at multiple levels to execute replacement and/or repair. Systems are also distributing larger numbers of digital channels as a consequence of migration to digital television. As part of this expansion of digital channels, there will likely be broader use of digital and high definition signals. In parallel, system operators are taking advantage of newly available technologies for deploying increased numbers of digital video services such as SDV and IP video simulcast, thus permitting those operators to provide larger channel line-ups with more variety of programming. Increasing the amount of available services, many of which may be narrowcast or unicast, will further increase the need for additional QAM channels (and equipment to provide those additional QAM channels), and/or the ability to quickly and easily reallocate QAM channels for different uses. These needs are not fully recognized in the art and are not adequately satisfied by architectures such as are shown in FIG. 1.

[27] By way of illustration, a typical 750 MHz network can support approximately 115 (6 MHz) channels. For such systems, a typical operator may have between 70 and 80 analog channels, leaving approximately 35 to 45 channels for use as digital QAM channels. The majority of those digital QAM channels may be used for broadcast digital programming, leaving 4-6 channels for VOD, 1-4 channels for HSD (depending on, for example, whether DOCSIS 3.0 downstream channel bonding is implemented) and 4-8 for SDV. This results in a distribution of channels such as the one depicted in Table 1 below.

Table 1: Illustrative Example of Current Channel Allocation for a Typical 750 MHz System

Analog Broadcast	70 to 80 Channels
Digital Broadcast	17 to 36 Channels
Digital Narrow- and Unicast	9 to 18 Channels
Narrowcast Proportion	8 to 15%

[28] A system operator may, however, wish to reallocate that channel usage so as to maintain approximately 25 to 35 analog channels and allocate the remaining channels as follows: 4-8 for HSD, 8-12 for VOD and Start Over (SO), 12-16 for SDV, and 12-16 for IP video simulcast, leaving the remaining channels for broadcast digital programs. This would result in a distribution of channels as depicted in the illustrative example of Table 2 below.

Table 2: Illustrative Projection of Future Channel Allocation for a Typical 750 MHz System

Analog Broadcast	25 to 35 Channels
Digital Broadcast	28 to 54 Channels
Digital Narrow- and Unicast	36 to 52 Channels
Narrowcast Proportion	30 to 45%

[29] For example, assuming an average size hub serving 40,000 homes, with a current average of 80 service group nodes serving 500 homes each, moving towards an average of 115 service group nodes of 350 homes served each, and considering 2 nodes per service group, the allocation of Table 2 would likely result in an average number of UEQAMs and, in a system employing DOCSIS, for example, DOCSIS downstream channels currently and projected as illustrated in Table 3 below.

Table 3: Illustrative Average Projection of Future Channel Allocation for a 750 MHz System

	Currently	Projected
Broadcast QAM channels	26	41
Narrow- and Unicast QAM channels	540	2,640
DOCSIS downstream channels	100	1,200

[30] The above illustrative example of a typical average size hub operated by a system operator in a typical industry manner shows growth from a current total of 566 QAM channels to a projected total of 2,681, and DOCSIS downstream channels from a current total of 100 to a projected total of 1,200. Despite a projected increase in QAM channel density per physical port in QAM devices, the above-illustrated growth in QAM channel needs results in large space requirements, considerable increase in power/cooling supply, and a significant increase in the number of devices to be monitored, managed and supported.

[31] FIG. 2 is a block diagram showing a network architecture that includes a subscriber access network according to at least some embodiments. In the architecture represented by FIG. 2, narrowcast and unicast data is routed to packet processing nodes (block 1020) instead of a termination system, such as CMTS cores. The packet processing nodes forward downstream data to (and receives upstream data from) access nodes (blocks 1021). Access nodes include QAM modulators and demodulators. As indicated below, a packet processing node and an access node can be contained in the same hub (or in the same chassis), or can be separated by several miles or more. Broadcast data is routed to separate UEQAM devices. Downstream data from those separate UEQAMS and from access nodes is combined and optically transmitted to service group nodes. The service group nodes operate in a manner similar to that indicated in connection with known architectures described in connection with FIG. 1. Upstream data from subscribers is received from service group nodes by access nodes and forwarded to packet processing nodes for further forwarding and routing. Notably, the architecture of FIG. 2 does not require any change in currently deployed subscriber premises equipment.

[32] In the embodiment of FIG. 1, CMTS cores 1004 incorporate considerable packet processing capabilities. These include subscriber management, service flow management, and other functions. UEQAMs also implement packet processing functions such as multiplexing single program transport streams into multi program transport streams, re-stamping program counters, and other functions. In the architecture of FIG. 2, and according to some embodiments, a packet processing node incorporates these functions into a single

device that would support both video over MPEG transport streams and DOCSIS streams. A packet processing node could also support the distribution of video streams over DOCSIS to implement IP video simulcast. The packet processing node would take into account the requirements of the different types of flows and treat them accordingly. For example, HSD and VoIP flows could be treated the same way that they are currently treated in a DOCSIS CMTS core. Video streams over MPEG transport streams would not require the same processing because such flows are of fairly constant packet size, unique source and destination, and unchanged purpose during the length of the stream which in itself is very long. Therefore, such well behaved video streams over DOCSIS receive a minimum amount of treatment. This differentiated treatment would optimize the burden that the various streams would place on the packet processing node, which would in turn leverage each component in the most efficient manner (rather than equally for all flows).

[33] In at least some embodiments, all narrowcast and unicast data traffic to an access network would be routed to a packet processing node. This would include all HSD, voice, IP video simulcast, VOD, SDV and SO flows. The packet processing node would handle each of these flows according to their specific needs in a corresponding way. Access nodes then communicate all narrowcast and unicast data to (and from) service group nodes. The access nodes control physical layer and at least some Media Access Control (MAC) functions previously controlled by a CMTS core in the architecture of FIG. 1.

[34] In some embodiments, a single packet processing node may serve multiple access nodes. A packet processing node and the access node(s) served by that packet processing node may be separate components housed in a common chassis or may be separate components housed in separate chassis. In some cases, a separately-housed packet processing node and one or more access nodes may be in the same location (e.g., a hub equipment room), while in other cases a packet processing node may be separated by its served access node(s) by several miles or more. For example, a system operator may locate packet processing nodes in a hub and locate access nodes within service group nodes.

[35] An architecture utilizing packet processing nodes and access nodes permits simplified port sharing between DOCSIS and MPEG transport streams. Thus, fewer ports will be needed for communications between a packet processing node and an access node than are required between a CMTS core and associated UEQAMs. This permits a system operator to leverage continuing advances in QAM technology that make available devices with more QAM channels per physical RF port (e.g., 16, 32, 64 or even more channels per port). This facilitates easier assignment of QAM channels on an RF port to DOCSIS or video over MPEG transport streams, enabling software remapping of downstream channels, thereby eliminating or reducing hardware-based combiners (e.g., such as represented by block 1005 in FIG. 1). In some embodiments, for example, a single RF port carrying video over MPEG transport streams and DOCSIS streams may have 16, 32, 64 or more QAM channels, with each those channels being flexibly re-assignable for use to convey HSD, VoIP, VOD, SDV, SO, IP video simulcast, etc.

[36] In current implementations, each CMTS core and each UEQAM is managed independently. Considerable knowledge is required to support and operate these devices. In at least some embodiments, an access node is managed via the packet processing node. In some embodiments, access node linecards could be configured using DHCP for IP address assignment and TFTP for configuration download. Additionally, a packet processing node could proxy all IPDR collection functions (and perhaps even SNMP functions).

[37] As explained in more detail below in connection with FIGS. 17-24, the architecture of FIG. 2 facilitates the use of access nodes for different technologies with a common packet processing node. This would permit optical and/or wireless access to be incorporated into a network more natively than is currently thought feasible in the art. In some cases, this may be implemented by incorporating certain functionality from the other access technologies into the packet processing node so such functions can be performed in a coordinated centralized manner.

[38] The architecture of FIG. 2 also permits, in at least some embodiments, consolidated data flows for all services into fewer, unified interfaces. By consolidating functions performed by a CMTS core and a UEQAM, for example, into a packet processing node, all input streams can be sent to the packet processing node instead of sending some to a CMTS core and many others directly to UEQAMS. This will reduce the number of interfaces to the backbone network (e.g., interfaces to a service operator's regional or national network) from one per QAM or one per UEQAM (usually a large number) and one per CMTS, to one or a few per packet processing node.

[39] If architecture of FIG. 2 is implemented in a modular fashion, the number of connections between a packet processing node and its served access node(s) would be significantly fewer than is currently the case between a CMTS core and UEQAMs. This reduction is due at least in part to the ability of the FIG. 2 architecture to leverage increased density of QAM channels per RF Port and the faster Ethernet interfaces. RF interfaces from an access node would be significantly less than are currently used since fewer RF Ports could be used for all narrowcast and unicast flows rather than one per QAM channel or one for each small group of QAM channels. If a packet processing node and access nodes are contained in a single device or chassis, implementing the architecture of FIG. 2 could be further simplified.

[40] FIG. 3 is a diagram of a network 1 according to at least some embodiments such as are described above in connection with FIG. 2. As discussed in more detail below, elements of network 1 utilize communication protocols and/or otherwise perform selected functions that may be similar to those described in one or more Data-Over-Cable Service Interface Specifications (DOCSIS) standards. Although certain aspects of network 1 will, for purposes of convenience, be described by reference to various DOCSIS standards, the invention is not limited to embodiments that employ DOCSIS-compliant components or protocols.

[41] As explained in more detail below, the embodiment of FIG. 3 differs from a conventional DOCSIS network in several respects. In network 1, for example, numerous

media conversion functions are consolidated into access nodes (AN) 4 through 6 and most packet processing functions are consolidated into a packet processing node (PPN) 9. Among other advantages, such an arrangement permits greater design flexibility and increases the ease with which the network can be scaled.

[42] As seen in FIG. 3, network 1 includes packet processing node 9, multiple access nodes 4 through 6 and user or subscriber devices (SD) 10-1 through 10-*j*. As used in this detailed description, an italicized lower case letter represents an arbitrary number. For convenience, the portion of network 1 that includes packet processing node 9, access nodes 4-6 and subscriber devices served by those access nodes will be referred to as an access network. The portion of an access network that includes an access node and user or subscriber devices served by that access node (e.g., access node 6 and subscriber devices 10-1 through 10-*j*) will be referred to as an access sub-network. Only three access nodes are shown for simplicity in FIG. 3, but multiple additional access nodes could be present in other embodiments. Packet processing node 9 and access nodes 4 through 6 may be separated by an arbitrary distance (e.g., by over one mile as shown in FIG. 3, by several hundred feet, by several inches, etc.), or may be located within the same room or within the same chassis. Each access node is in communication with multiple subscriber devices, although only subscriber devices 10-1 through 10-*j* in communication with access node 6 are shown in FIG. 3. Examples of user or subscriber devices include a DOCSIS-compliant cable modem for high speed data service, a media terminal adapter for VoIP service, a set-top box for providing VOD or other services, a display device such as a television, a handheld wireless device, etc. In the embodiment of FIG. 3, subscriber devices 10-1 through 10-*j* and access nodes 4 through 6 may communicate over a provider's access sub-network. In an HFC-type network, user devices and access nodes may communicate over a DOCSIS radio frequency (RF) interface 22 carried on conventional hybrid fiber coax (HFC) links 12, 13 and 14 according to one or more of the following standards: DOCSIS Downstream RF Interface Specification (CM-SP-DRFI-I06-080215), DOCSIS 3.0 Physical Layer Specification (CM-SP-PHYv3.0-I07-080522), DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification (CM-SP-MULPIv3.0-I08-080522). All of said standards are known in the art

and available from Cable Television Laboratories, Inc. (CableLabs®) of Louisville, Colorado.

[43] Packet processing node 9 communicates with a network backbone 15 over one or more links 18 via one or more routers 11. In at least some embodiments, packet processing node 9 and network backbone 15 communicate across a network side interface (NSI) 19 using internet protocol (IP) over one or more 40 Gbps or 100 Gbps Ethernet connections. Other types of NSIs may be used in other embodiments. Communications across NSI 19 include two-way IP traffic. That IP traffic includes data originating from or destined to one or more user or subscriber devices. That IP traffic may also include system management information communicated between packet processing node 9 and other elements of network 1. Examples of such other elements could include configuration servers and databases, management servers and databases, domain name servers, VoIP call management agents, a timing server for providing a common timing frequency/timestamp, etc. Network backbone 15 may be, e.g., a corporate IP network interconnecting multiple packet processing nodes and providing a connection to the Internet and/or other wide area networks.

[44] As discussed in more detail below, packet processing node 9 receives data packets from network backbone 15, encapsulates and/or otherwise processes those packets, and forwards those packets across interface 21 to access nodes 4 through 6 for further transmission to the appropriate subscriber devices. Packet processing node 9 similarly receives data from subscriber devices, via access nodes 4 through 6 and across interface 21, and forwards that data over network backbone 15. In at least some embodiments (e.g., where a packet processing node and access nodes are separated by over a mile), interface 21 is a high data rate multiGigabit Ethernet connection, carried on a fiber optic link or an electrical connection, over which packet processing node 9 transmits multiple layer-2 data flows or creates multiple IP tunnels with each of access nodes 4 through 6. However, other types of physical and data links, such as wireless links, can be used to connect packet processing node 9 with access nodes 4 through 6. Packet processing node 9 forwards packets for voice, data and/or video over IP services or multiprogram transport stream (MPTS) packets used to, e.g.,

provide video services. In some embodiments, packet processing node 9 proxies IPDR (Internet Protocol Data Record) and/or SNMP (Simple Network Management Protocol) functions to one or more servers 2.

[45] As noted above, communications between access nodes 4 through 6 and subscriber devices are over RF interface 22, which may be a DOCSIS RF interface. Communications from an access node to a subscriber device are referred to herein as "downstream" communications. Similarly, communications from a user device to an access node are referred to herein as "upstream" communications. As is known in the art, a DOCSIS RF interface includes multiple frequency channels that may be used for upstream communications, and multiple other frequency channels that may be used for downstream communications. Typically, downstream channels have a 6 MHz bandwidth and are located between 52 MHz and 1002 MHz. Upstream channels may have multiple bandwidth assignments and are typically located between, e.g., 5 MHz and 42 MHz. Although other types of modulation can be used, upstream and downstream communications across a DOCSIS RF interface are usually modulated using quadrature phase shift keying (QPSK), quadrature amplitude modulation (QAM), synchronous code division multiple access (S-CDMA) and possibly other modulation techniques. These modulation techniques may be employed by the access nodes and subscriber devices in the embodiment of FIG. 3, and each upstream and downstream frequency channel is thus also referred to herein as a QAM channel. Each upstream and downstream QAM channel is frequency divided, and may also (or alternatively) be code-division multiplexed.

[46] In the embodiment of FIG. 3, each QAM channel is a separate physical channel over which data flows to (or from) numerous subscriber devices. For convenience, frequency sub-bands used for upstream transmissions are also referred to herein as "channels," although the width of such upstream channels in some embodiments may vary from, e.g., 200 KHz to 6.4 MHz. Although the embodiment of FIG. 3 employs frequency multiplexing of the physical medium to achieve separate physical channels, other techniques could be used in other embodiments. For example, multiple separate physical channels could be achieved by time

division multiplexing. Similarly, the embodiment of FIG. 3 employs a wired medium to communicate between access nodes and subscriber devices. As used herein (including the claims), "wired" refers to tangible physical media used as a conduit for electromagnetic signals carrying data, and includes optical fiber, coaxial cable and other physical media. In other embodiments, communications between access nodes and subscriber devices in an access sub-network may be over a wireless medium such as open air radio frequency communication.

[47] In some embodiments, downstream data packets transmitted to subscriber devices (including system management messages) are encapsulated into MPEG data packets. A subscriber device will be tuned to one or more downstream QAM channels, and will "receive" all of the data packets transmitted over those channels. However, a particular subscriber device will only process data packets that are addressed (multicast, narrowcast or unicast, for example) to that device and will ignore other packets. A subscriber device tunes to one or more downstream QAM channels in response to instructions received from an access node. Upstream transmissions by a subscriber device are managed by the access node with which that subscriber device communicates. Those upstream transmissions are controlled using the media access control protocol typically set forth in a standard, such as in the previously-identified DOCSIS standards.

[48] Briefly, a "MAP" system management message is periodically inserted into each downstream QAM channel used for DOCSIS communications. The MAP message identifies times during which a subscriber device may transmit in a particular upstream QAM channel. In some cases, a MAP message may identify times at which any of numerous subscriber devices may attempt to transmit an upstream message on a contention basis. If more than one subscriber device transmits at that time and the transmissions collide, each device waits a random period of time and then attempts to transmit at a different time (e.g., a time identified in a later MAP message). Each device would know of the collision because it would fail to receive an acknowledgement of its transmission. In other cases, a MAP message may specify that certain upstream times are reserved for a specific subscriber device.

[49] The manner in which a subscriber device sends (or receives) data is affected by one or more service flows that have been assigned to the device. Under DOCSIS, for example, quality of service (QoS) levels are managed by establishing different "service flows" within a particular QAM channel. Each service flow is characterized by a set of QoS parameters such as latency, jitter, and throughput assurances. As one example, a best effort (BEF) service flow can be used to provide data communications with no guarantee that data will be delivered within a particular time period and with no guarantee of media access. Under some implementations of a BEF service flow, a subscriber device must request permission to transmit whenever it has data to send upstream. The request is typically sent in a contention-basis time slot that has been identified in a MAP message. In response to the request, a subsequent MAP message identifies one or more times reserved for that user or subscriber device to transmit its data. In some implementations, a data transmission during one of those reserved times may include a further request for additional transmission times.

[50] An unsolicited grant service (UGS) flow can be used to provide recurring times at which a subscriber device can transmit upstream without having to contend for such times with other subscriber devices. A subscriber device operating under a UGS flow need not request a reservation each time it has data to transmit. Once a UGS flow is granted, a series of MAP messages will identify recurring time blocks during which the subscriber device may transmit. MAP messages granting these time blocks will continue until the UGS flow is canceled. A UGS flow is used for, e.g., VoIP service.

[51] Although UGS and BEF flows are described as examples, numerous other types of service flows could be implemented. Moreover, and as explained in more detail below, a service flow also affects the manner in which data is transmitted downstream. For example, some service flows (e.g., for VoIP service) have higher priority than others (e.g., a BEF flow) and are transmitted with less latency and/or with a higher throughput assurance.

[52] In the embodiment of FIG. 3, upstream transmissions by a subscriber device are directed to an access node. For example, assume subscriber device 10-1 is a modem

transmitting an email message to a computer attached to subscriber device 10-4 (another modem or the computer itself, for example). The data for that email would be transmitted from subscriber device 10-1 to access node 6, from access node 6 to packet processing node 9, and from packet processing node 9 to other elements within (or across) network backbone 15 (e.g., an email server). Those other network elements would then forward the email data back to packet processing node 9, which would forward the email data to access node 6, which would then forward the data to subscriber device 10-4.

[53] For example, in a conventional DOCSIS system employing a modular cable modem termination system (M-CMTS) architecture, an M-CMTS core communicates with a network backbone and with one or more Edge QAM (EQAM) devices. The EQAMs then communicate with subscriber devices over HFC connections. A conventional M-CMTS architecture is described, e.g., in the DOCSIS Modular CMTS Downstream External PHY Interface Specification (CM-SP-DEPI-I05-070223), also available from CableLabs®. In the embodiment of FIG. 3, packet processing node 9 and access nodes 4 through 6 replace an M-CMTS and EQAMs. Unlike a conventional M-CMTS, however, packet processing node 9 of network 1 does not perform media conversion functions in at least some embodiments. Those functions are instead performed by access nodes 4 through 6, thereby permitting a packet processing node and an access node to be separated by distances of several miles or more. This allows much greater design flexibility than is available in a conventional M-CMTS architecture, where the separation between an M-CMTS core and EQAM is effectively limited to several hundred feet.

[54] FIG. 4 is a functional block diagram of packet processing node 9 according to some embodiments. Packet processing node 9 initially buffers incoming IP packets from network backbone 15 in a queue 24. Some incoming IP packets may contain narrowcast or unicast data (e.g., emails, Internet web pages, etc.) destined for a particular subscriber device, group of subscriber devices and/or device(s) coupled subscriber device(s) (e.g., data destined for a computer attached to a modem). Other incoming packets may contain narrowcast or unicast data such as single program transport stream (SPTS) data corresponding to a specific video

program or other service destined for one or more subscriber devices. Still other incoming packets may contain system management messages for packet processing node 9. For example, a user in another network may be seeking to establish a VoIP session with a user device in network 1. In such a case, one or more control messages would inform packet processing node 9 of the session and of the need to establish an appropriate service flow for the destination subscriber.

[55] Packet analyzer 25 removes packets from queue 24 and may determine what type of data is contained in the packet. Analyzer 25 makes this determination by filtering predefined fields of the packets. One or more of such fields have values identifying packets as subscriber data, SPTS data, or control messages. Additional fields identify the destination of the data in the packet. That destination may be a single subscriber device or multiple subscriber devices. Other fields identify various QoS parameters for the data in the packet, which parameters are used by packet processing node 9 to classify that data according to one or more service flows. The values identifying packet data type, the values identifying destinations, and values identifying QoS values, etc. are inserted by other nodes located in (or across) network backbone 15 that are sending and/or forwarding data to packet processing node 9 and/or to one or more of access nodes 4 through 6.

[56] The determinations made by analyzer 25 may be communicated to PPN controller 30. Based on those determinations, PPN controller 30 routes the data packet to an appropriate outgoing QAM channel queue (discussed below) or to the controller's incoming message queue 31 after passing through pre-processing block 26. Among other operations, pre-processing block 26 may assemble data split across multiple incoming IP packets. In some embodiments, pre-processing block 26 may also MPEG-encapsulate data packets. Pre-processing block 26 may also perform packet processing functions conventionally performed by UEQAMs in architectures such as shown in FIG. 1 (e.g., multiplexing single program transport streams into multi program transport streams, re-stamping program counters, and other functions).

[57] As indicated above, information regarding an analyzed packet may be provided to PPN controller 30. If the analyzed packet contains data intended for a subscriber device, PPN controller 30 may consult a subscriber database 33 and identify at least one downstream QAM channel over which the intended subscriber device receives transmissions. PPN controller 30 also determines an applicable service flow for the data. The service flow may be a DOCSIS service flow or may be a service flow established for VOD or other type of narrowcast data, for example. In some cases, information regarding an existing service flow may be stored in database 33 for the subscriber device. In other cases, PPN controller 30 may determine an applicable service flow based on information about the packet received from analyzer 25. After determining the appropriate QAM channel and service flow, PPN controller 30 causes the packet to be transferred to the appropriate queue. This is shown schematically with a control signal arrow from PPN controller 30 to a demultiplexer 34 that directs a packet to any of the queues associated with the downstream QAM channels of several access nodes. In particular, packet processing node 9 could maintain multiple outgoing packet queues for each downstream QAM channel of each access node with which packet processing node 9 communicates over interface 21. For example, access node 6 has *p* downstream QAM channels. Each of those QAM channels has multiple queues SF1 through SF*n* corresponding to *n* different service flows. Each of those QAM channels also has a queue "sys." for system management messages, examples of which are discussed below. For simplicity, FIG. 4 only shows outgoing queues for two downstream QAM channels of access node 6. The presence of additional queues, as well as additional signal lines and multiplexers for those queues, is represented generically with vertical ellipses.

[58] PPN controller 30 also determines the order in which data packets are taken from the outgoing downstream QAM channel queues. This determination is based, e.g., on the relative priorities of the various service flows, the amount of data in the queues, etc. For example, an outgoing queue for a high priority service flow of one QAM channel may have a large number of packets, thus requiring delay of packets in a queue for a lower priority service flow, while the opposite situation may be simultaneously occurring in the queues for a different QAM channel. Removal of data packets from the outgoing QAM channel queues is

shown schematically in FIG. 4 with multiplexers 36 and 37. PPN controller 30 also controls the order in which outgoing packets are taken for each QAM channel, shown schematically as multiplexer 38. PPN controller 30 then causes outgoing data packets to be appropriately encapsulated for communication across interface 21. For simplicity, such encapsulation will simply be called "PPN/AN encapsulation." Such encapsulation may include L2TPv3 encapsulation. Data packets directed to a particular downstream QAM channel, in at least some embodiments, include channel information addressing those packets using QAM channel flows as described in the above-mentioned DOCSIS Modular CMTS Downstream External PHY Interface Specification. However, channel information to identify a downstream communication channel from an access node could be included in other ways. From block 39, PPN/AN-encapsulated packets are forwarded to access nodes across interface 21.

[59] Packet processing node 9 also receives incoming data from access nodes across interface 21 and buffers that data in a queue 40. Packet analyzer 42 removes packets from queue 40 and determines what type of data is contained in those packets. As with packets received from network backbone 15, analyzer 42 filters predefined fields of the packets received from access nodes. Some of those fields may contain values inserted by a subscriber device, while other fields may contain data inserted by an access node. Some of those incoming packets contain data that is to be forwarded to the network backbone 15 across interface 19. Other packets include system management messages from access nodes or from subscriber devices. Information from analyzer 42 about a packet is used by PPN controller 30 to route that packet for further processing. That routing is shown schematically with demultiplexer 43. For packets identified as outbound data, PPN controller 30 may route those packets for further processing in block 44. That further processing may include, e.g., removing PPN/AN encapsulation, re-encapsulating the data in an IP packet, adding appropriate source and destination port numbers, etc. From block 44, data is transmitted to network backbone 15 or other points in a provider's system. In some embodiments, packet processing node 9 may include one or more queues for data packets to be forwarded across network backbone 15; said queues could correspond to different QoS levels or otherwise

have differing priorities. For packets identified as system management messages, PPN controller 30 may route those packets to message queue 31. PPN controller 30 could then retrieve messages from queue 31 for further processing.

[60] As indicated above, FIG. 4 is a functional block diagram. The number and arrangement of blocks in FIG. 4 are merely intended to explain operations performed by packet processing node 9. The individual blocks of FIG. 4 do not necessarily correspond to separate physical components. As discussed in more detail in connection with FIG. 15, the operations represented by blocks in FIG. 4 could be distributed across one or more microprocessors and memory devices. Similarly, the demultiplexers, multiplexers and signal lines are used in FIG. 4 to schematically show packet handling within packet processing node 9. Many embodiments will not be physically implemented using discrete demultiplexers, multiplexers and signal lines. For example, packets received from network backbone 15 could alternately be processed using software that stores data using memory addresses mapped to service flows and QAM channels and that selectively retrieves data from those assigned addresses for encapsulation and transmission. Similar software could be used to process packets received from access nodes.

[61] FIG. 5 is a functional block diagram of access node 6. Access node 6 initially buffers incoming PPN/AN-encapsulated packets in a queue 61. Some of those incoming packets contain data and/or control messages intended for one or more subscriber devices. Other packets contain system management messages intended for access node 6. As explained above, packet processing node 9 has already determined which subscriber devices are on a particular QAM channel. Packet processing node 9 has similarly ordered packets in the data stream based on applicable service flows. Accordingly, all packets bound for user devices on a particular downstream QAM channel are marked as corresponding to that QAM channel. Access node 6 need not prioritize transmission of downstream data packets based on service flows, and can simply place all downstream packets marked for a particular QAM channel into a single downstream packet queue.

[62] Packet analyzer 62 may remove packets from queue 61 and determine whether each packet contains data packets intended for a specific QAM channel or a system management message for access node 6. Analyzer 62 makes this determination by filtering predefined fields of the received PPN/AN-encapsulated packets. The determinations made by analyzer 62 may be communicated to AN controller 70. Based on those determinations, AN controller 70 routes data packets to the packet queues for the appropriate downstream QAM channels. System management messages for access node 6 may be routed to a message queue 71.

[63] If an analyzed de-queued packet is marked as containing packets for transmission on a downstream QAM channel, AN controller 70 causes the packets within the PPN/AN-encapsulated packet to be transferred from analyzer 62 to the packet queue for the appropriate downstream QAM channel. This is shown schematically with a demultiplexer 65 that directs a packet (or group of packets) to any of the downstream QAM channel packet queues. In some embodiments, PPN/AN encapsulation is first stripped in block 64 so as to yield MPEG-encapsulated data packets ready for upconversion and modulation. PPN/AN-encapsulated packets received from packet processing node 9 and marked for a particular QAM channel may have sequence numbers so that access node 6 can detect dropped PPN/AN packets and/or reorder PPN/AN packets that have been received out of sequence. Dropped packet detection and/or reordering of packets received out of sequence can also be performed in block 64.

[64] Each downstream QAM channel of access node 6 also has a management message queue. As explained in more detail below, AN controller 70 periodically generates MAP messages and other types of system management messages. AN controller 70 places these messages into the downstream management message queues of the downstream QAM channels. As shown schematically with multiplexers 68-1 through 68-*p*, AN controller 70 also controls the order in which messages are taken off the packet and management message queues for each downstream QAM channel and forwarded to the modulator/upconverter for that channel. The QAM modulator/upconverters for each downstream channel are indicated in FIG. 5 with references 69-1 through 69-*p*. In the embodiment of FIGS. 1-14, access node

6 can control the generation and transmission of many system management messages to subscriber devices. Although only two QAM upconverter/modulator blocks are shown, access node 6 may have hardware corresponding to many more QAM upconverter/modulators (e.g., 16, 32, 64 or more) outputting signals that emanate from a single physical port.

[65] Upstream transmissions from subscriber devices on an upstream QAM channel are received in QAM demodulators/downconverters 80-1 through 80-*k* and buffered in queues 81-1 through 81-*k*. Such upstream messages may be DOCSIS upstream messages, upstream messages corresponding to VOD commands (e.g., starting or stopping a program), VoIP data, etc. For each upstream channel, an analyzer 82 removes packets from the appropriate queue and determines if the packet contains data for transmission to packet processing node 9, a system message (e.g., a request for an upstream transmission time), or both. Each analyzer 82 conveys this determination to AN controller 70. As shown schematically with demultiplexers 84-1 through 84-*k* (which could receive signals from a single physical input port), AN controller 70 then transfers PPN-bound data to outgoing queue 86 and management messages to message queue 71 of AN controller 70. In some embodiments, outgoing queue 86 may include multiple queues and/or those multiple queues may correspond to different service flows or otherwise have different priorities. As with packet processing node 9, a service flow may be a DOCSIS service flow or may be a service flow established for VOD or other type of narrowcast and/or unicast service. AN controller 70 may remove data from queue 86 and perform PPN/AN encapsulation in block 87, and then forward PPN/AN-encapsulated data to packet processing node 9 across interface 21. AN controller 70 may periodically remove messages from its message queue 71 and act on those messages, as described in more detail below. AN controller 70 also stores data in, and retrieves data from, a database 89.

[66] As indicated above, FIG. 5 is a functional block diagram. The number and arrangement of blocks in FIG. 5 are merely intended to explain operations performed by access node 6. The blocks of FIG. 5 do not necessarily correspond to individual physical

components. As discussed in more detail in connection with FIG. 16, the operations represented by blocks in FIG. 5 could be distributed across one or more microprocessors and memory devices. Similarly, demultiplexers, multiplexers and signal lines are used in FIG. 5 to schematically show packet handling within access node 6. Many embodiments will not be physically implemented using discrete demultiplexers, multiplexers and signal lines as shown. For example, PPN/AN-encapsulated packets received from packet processing node 9 could alternately be processed using software that stores data using memory addresses mapped to QAM channels and that selectively retrieves data from those assigned addresses for transmission. Similar software could be used to process packets received from subscriber devices. FIG. 5 shows access node 6 having an arbitrary number (p) of downstream QAM channels and arbitrary number (k) of upstream channels. In some embodiments $p > k$; in other embodiments $p = k$; and in still other embodiments $p < k$. As in FIG. 4, the ellipses in FIG. 5 represent additional signal lines, queues and other components corresponding to additional channels.

[67] FIGS. 6-9 are flow charts describing exemplary operations performed by PPN controller 30 of packet processing node 9 according to at least some embodiments. In at least some such embodiments, the sequences of operations shown in FIGS. 6-9 are performed simultaneously in multiple program threads. For example, FIG. 6 shows operations performed by a program thread that processes incoming data packets from network backbone 15. In block 202, a data packet is removed from queue 24 (see FIG. 4). Next, and as shown in block 204 (and as represented by block 25 of FIG. 4), the de-queued packet is analyzed. In block 206 of FIG. 6, PPN controller 30 determines if the packet contains a system management message or other type of message directed to packet processing node 9. If the packet contains such a message, it is forwarded to message queue 31 in block 208. From block 208, controller 30 returns to block 202.

[68] If a de-queued packet contains data intended for a subscriber device, AN controller 30 instead proceeds from block 206 to block 210. In block 210, PPN controller 30 maps the data to the appropriate service flow on the appropriate QAM channel and then forwards that data

to the appropriate downstream QAM channel queue (shown with demultiplexer 34 in FIG. 4). To map data to a service flow and QAM channel, PPN controller 30 compares information regarding the de-queued packet with information in subscriber device database 33 (FIG. 4). For example, the data might be VoIP data corresponding to a service flow that has already been established for a VoIP call. In some embodiments, data for a subscriber device that is not associated with a higher priority service flow is by default placed into a best efforts service flow queue. Pre-processing represented by block 26 of FIG. 4 can also be performed as part of the step(s) corresponding to block 210 of FIG. 6. From block 210, PPN controller 30 returns to block 202 to repeat the algorithm for the next de-queued data packet.

[69] FIG. 7 shows operations performed by a program thread that processes incoming data packets from access nodes. In block 232, PPN controller 30 removes a packet from queue 40 (FIG. 4). In block 234, PPN controller 30 analyzes the de-queued packet using analyzer 42 of FIG. 4. PPN controller 30 then determines in block 236 of FIG. 7 if the packet contains data that is to be forwarded across network backbone 15. If so, PPN controller 30 proceeds on the "yes" branch to block 238. In block 238, PPN controller 30 performs further processing on the outbound data, as represented schematically in FIG. 4 by block 44, and then transmits the data across network backbone 15 (block 240 of FIG. 7). PPN controller 30 then returns to block 232. If a determination is made in block 236 that a packet contains data requiring further action by PPN controller 30 (e.g., a request to establish a particular type of service flow), the packet is forwarded to message queue 31 (FIG. 4). From block 242 of FIG. 7, PPN controller 30 returns to block 232 to repeat the algorithm for the next de-queued packet.

[70] FIG. 8 shows operations performed by a program thread that processes messages in message queue 31 of FIG. 4. In block 250, PPN controller 30 removes the next message from queue 31. In block 252, PPN controller 30 determines if the message is a request by a subscriber device to establish a particular type of service flow. As explained in more detail below, access nodes 4 through 6 control media access across interface 21 for some types of service flows, but other types of service flows are set up and torn down under the control of

packet processing node 9. If the de-queued packet is a request to establish or discontinue a type of service flow managed by packet processing node 9, PPN controller 30 processes that request in block 254. As part of that processing, PPN controller 30 updates subscriber database 33 as necessary. PPN controller 30 may also place a corresponding response message (e.g., a message granting a requested service flow) into the appropriate queue of the appropriate QAM channel. From block 254, PPN controller 30 returns to block 250.

[71] If a de-queued packet does not contain a request to set up or tear down a service flow, PPN controller 30 proceeds on the "no" branch from block 252 to block 256 and determines if the message is a system management message from an access node. Examples of such messages could include messages from an access node seeking configuration and/or provisioning data, messages advising that certain QAM channels are reaching maximum capacity, system failure or error messages, etc. If the de-queued control message is a system management message from an access node, the message is processed in block 258. As part of that processing, PPN controller 30 may place a message into the control message queue or other appropriate queue for one or more QAM channels. From block 258, PPN controller 30 returns to block 250.

[72] If the de-queued control message is not a system administration message from an access node, PPN controller 30 proceeds to block 260 and determines if the message is related to a subscriber device reboot or if the message contains provisioning/configuration data for a subscriber device. If so, the message is processed in block 262. If the reboot/configuration message originated from the subscriber device, the message is forwarded over network backbone 15 to the appropriate configuration server or other destination. If the reboot/configuration message originated from a location in network backbone 15 (e.g., a response from a configuration server), PPN controller 30 places the data in the message into the appropriate queue of the appropriate QAM channel for forwarding to the subscriber device.

[73] The above are just some examples of various types of control messages. Similar processing occurs with regard to other types of messages. The presence of additional decisional and processing steps for such other message types is represented generically in FIG. 8 with broken line blocks 264 and 266 and vertical ellipses. Although not shown in FIG. 8, the algorithm includes a default processing step (e.g., sending an error message and returning to block 250) if a message de-queued in block 250 does not match a known message type.

[74] FIG. 9 is a diagram showing one example of an algorithm by which PPN controller 30 retrieves data from QAM channel queues and forwards that data across interface 21 to access nodes. Beginning in block 280, counters QC and qp are set to zero. In the algorithm of FIG. 9, values of QC identify specific downstream QAM channels and values of qp identify queues for a given QAM channel. Moreover, the identifying values for queues within a given QAM channel are assigned such that lower numbers correspond to higher priority, and the downstream management message queue of each QAM channel is given highest priority (i.e., for each value of QC, qp = 0 represents the "sys." downstream management message queue).

[75] From block 280, PPN controller 30 proceeds to block 281 and determines if there is data in the downstream queue for the current values of QC and qp. If not, PPN controller 30 proceeds on the "no" branch to block 287, which is described below. Otherwise, PPN controller 30 proceeds on the "yes" branch to block 282. In block 282, PPN controller 30 forwards the de-queued packet to block 39 (FIG. 4) for appropriate PPN/AN-encapsulation and transmission. From block 282 of FIG. 9, PPN controller 30 proceeds to block 283 and determines if there is more data in the queue for the current QC and qp values. If not, PPN controller proceeds on the "no" branch to block 287. Otherwise, PPN controller 30 proceeds on the "yes" branch to block 284 and determines if a "send all data until empty" flag is set for the current queue. In particular, the priorities for some queues may be such that all data is to be sent from that queue before any data from a lower priority queue may be transmitted. By way of example, packet processing node 9 may be configured to transmit all management

messages from the "sys." queue before any subscriber data in the SF1 through SF*n* queues can be sent.

[76] If the "send all" flag is not set, PPN controller 30 proceeds on the "no" branch from block 284 to block 285 and determines if the amount of data sent for the current queue during the current loop through the FIG. 9 algorithm has exceeded a predetermined maximum. By way of further example, PPN controller 30 may be configured such that some service flows have priorities that limit the amount of queued data to be transmitted in a particular period of time (e.g., a period of time corresponding to the frequency with which packet processing node 9 sends PPN/AN-encapsulated packets to particular QAM channel). If the amount of data sent from the current queue in the current algorithm loop has exceeded a maximum value, PPN controller 30 proceeds on the "yes" branch to block 289, which is discussed below. If the amount of data sent from the current queue has not exceeded the maximum value, PPN controller 30 proceeds to block 286.

[77] In block 286, PPN controller 30 determines if the amount of data sent for the current QAM channel has exceeded a maximum amount that can be sent in a particular time period. For example, packet processing node 9 may time-multiplex transmissions to multiple access nodes over a single physical media, and transmitting further data to the QAM channel corresponding to the current QC value across interface 21 could require more time than is available in a given time-multiplex period. If the amount of data sent to the current QAM channel has exceeded a maximum amount, PPN controller 30 proceeds on the "yes" branch to block 288, which is discussed below. Otherwise, PPN controller 30 returns to block 282.

[78] If in block 284 PPN controller 30 determines that the "send all flag" is set, PPN controller 30 proceeds on the "yes" branch to block 287. In block 287, and in a manner similar to that described in connection with block 286, PPN controller 30 determines if the amount of data sent for the current QAM channel has exceeded a maximum amount that can be sent in a particular time period. If so, PPN controller 30 proceeds on the "yes" branch to block 288 and determines if the current value of QC is

"QC_last," which value corresponds to the last QAM channel of the last access node (in the present example, QAM channel *n* of access node 6). If QC = QC_last, PPN controller 30 returns to block 280. If QC ≠ QC_last, PPN controller 30 proceeds to block 291. In block 291 PPN controller 30 increments QC by 1 and resets qp to 0. From block 291 PPN controller returns to block 281.

[79] If in block 287 PPN controller 30 determines that the amount of data sent on the current QAM channel has not exceeded a maximum amount, PPN controller 30 proceeds on the "no" branch to block 289. In block 289, PPN controller 30 determines if there are additional queues for the current QAM channel that may have data to transmit. In particular, PPN controller determines if the value of qp is equal to a value ("qp_last") corresponding to the lowest priority queue for the current QAM channel. If qp is not equal to qp_last, PPN controller 30 increments qp in block 290 and then returns to block 281. If qp is equal to qp_last, PPN controller 30 proceeds to block 288.

[80] FIGS. 10-14 are flow charts describing operations performed by access node 6 according to at least some embodiments. Similar operations would be performed by other access nodes in network 1 of FIG. 3. In at least some such embodiments, the sequences of operations shown in FIGS. 10-14 are performed simultaneously in multiple program threads.

[81] FIG. 10 shows operations performed by AN controller 70 of access node 6 (FIG. 5) for packets received from packet processing node 9 across interface 21. In block 340, AN controller 70 removes the next packet from queue 61. In block 341, the de-queued packet is analyzed (block 62 of FIG. 5). In block 342 of FIG. 10, AN controller 70 determines if the packet contains data packets addressed to a particular QAM channel. As indicated above, data packets addressed to a subscriber device on a QAM channel could include user data (e.g., emails, VOD data, data from the Internet, VoIP data, etc.) and/or could include control messages or other control/configuration/provisioning data from packet processing node 9 or from other network elements. If a de-queued packet contains packets addressed to a QAM channel, AN controller 70 transfers those packets to the appropriate QAM channel packet

queue in block 343. As part of that transfer, AN controller 70 strips PPN/AN encapsulation from the packets so as to yield MPEG-encapsulated packets ready for upconversion and modulation. Said stripping can be performed in block 64 (FIG. 5) or in additional blocks interposed between queue 61 and demultiplexer 65 of FIG. 5. From block 343, AN controller 70 returns to block 340. If at block 342 of FIG. 10 a de-queued PPN/AN-encapsulated packet contains a management message for access node 6, the management message is routed to message queue 71 (FIG. 5) at block 344. From block 344, controller 70 returns to block 340.

[82] FIG. 11 shows operations performed by AN controller 70 to process packets received from user or subscriber devices across interface 22 for upstream QAM channel 1. The algorithm of FIG. 11 can be performed simultaneously in multiple program threads for each of the other upstream QAM channels. In block 355, AN controller 70 de-queues the next packet from incoming packet queue 81-1 (FIG. 5). AN controller 70 then analyzes the packet in block 356, said analyzing corresponding to block 82-1 of FIG. 5. If the packet is determined in block 357 to be a system management message (examples of which are provided below), the message is routed to message queue 71 (FIG. 5) at block 358 (FIG. 11). From block 358, AN controller 70 returns to block 355. If the packet is determined in block 357 to be outbound data or a message bound for packet processing node 9, the message is forwarded at block 359 for further processing and transmission across interface 21. Said further processing (e.g., PPN/AN encapsulation) and transmission are performed in block 87 of FIG. 5. From block 359 of FIG. 11, AN controller 70 returns to block 355.

[83] FIG. 12 shows operations by AN controller 70 to process system management messages. In block 370, AN controller 70 de-queues the next management message from queue 71 (FIG. 5). In block 371, AN controller 70 determines if the message is a request from a subscriber device for permission to transmit. If so, AN controller 70 proceeds on the "yes" branch to block 372 and determines if the management message is a request from a subscriber device for an upstream transmission time under a service flow controlled by access node 6.

[84] If a request is for a BEF service flow upstream transmission opportunity, AN controller 70 proceeds to block 373. In block 373, AN controller 70 autonomously determines when the subscriber device may transmit. AN controller 70 can make that autonomous determination based on current upstream traffic conditions, fairness-based algorithms (e.g., to prevent a subscriber device from monopolizing upstream bandwidth), and/or on other bases. AN controller 70 then stores the determined response to the BEF upstream transmission request in database 89 (FIG. 5); the stored response will be included in a downstream MAP message, as described below. In at least some embodiments, access node 6 controls when subscriber devices may send upstream transmissions for a BEF service flow, but upstream transmissions for other service flows are controlled by packet processing node 9. For example, packet processing node 9 could periodically inform access node 6 of the amount of capacity on each upstream QAM channel that access node 6 can assign (or the amount that must be treated as reserved for assignment by packet processing node 9). In other embodiments, an access node could also control upstream transmissions under additional service flows. From block 373 AN controller 70 returns to block 370.

[85] If AN controller 70 determines in block 372 that a request for an upstream transmission opportunity is for a type of service flow managed by packet processing node 9, AN controller 70 forwards that request to packet processing node 9 at block 374, and then returns to block 370. AN controller 70 forwards the message to queue 86 (FIG. 5), from which the message is then PPN/AN encapsulated in block 87 and sent across interface 21.

[86] If the message de-queued at block 370 of FIG. 12 was not a subscriber device request for an upstream transmission time, AN controller 70 proceeds from block 371 on the "no" branch to block 375. AN controller 70 determines in block 375 if the management message is a service flow management message from packet processing node 9 (e.g., a response to a previously forwarded request from a subscriber device). If so, AN controller 70 proceeds on the "yes" branch to block 376 and updates database 89 as necessary. For example, packet processing node 9 may have set up a service flow in response to a subscriber device request and may thus instruct access node 6 to include an appropriate instruction to the requesting

subscriber device in the next MAP message on a particular QAM channel. As another example, packet processing node 9 may have set up a UGS flow or other type of flow providing a regularly recurring upstream transmit time for a particular subscriber device. Accordingly, packet processing node 9 may instruct access node 6 that MAP messages on a designated QAM channel should assign transmit opportunities of a specified duration and at specified intervals until further notice. After updating database 89, AN controller 70 returns to block 370.

[87] Returning to block 375, if the message de-queued in block 370 is not a flow management message from packet processing node 9, AN controller 70 determines if it is another type of management message from packet processing node 9 (e.g., configuration/provisioning data, other type of system administration message, etc.) or from a subscriber device. If so, AN controller 70 takes appropriate action and then returns to block 370. The presence of additional decisional and processing steps for such other message types is represented generically in FIG. 12 with broken line blocks 377 and 378 and vertical ellipses. Although not shown in FIG. 12, the algorithm includes a default processing step (e.g., sending an error message and returning to block 370) if a message de-queued in block 370 does not match a known message type.

[88] The additional types of management messages shown generically in FIG. 12 with blocks 377 and 388 include various messages to and from subscriber devices. Under DOCSIS, subscriber devices transmit messages (known as RNG-REQ messages) that are used to calculate timing offsets, transmit power adjustments, etc., which offsets and adjustments are then sent to subscriber devices in downstream RNG-RSP messages. Unlike a conventional DOCSIS architecture, where such messages, timing adjusts, power adjustments, etc. are handled in an M-CMTS core, such messages and adjustments are handled by access nodes in at least some embodiments of FIGS. 1-14. If access node 6 receives a RNG-REQ message, AN controller 70 processes that message by calculating timing and/or power adjustment corrections for the subscriber device that sent the RNG-REQ

and stores those adjustments in database 89 for subsequent communication to that subscriber device.

[89] FIG. 13 is a flow chart showing operations by AN controller 70 to generate MAP messages and other system management messages for transmission on downstream QAM channels. In at least some embodiments, the algorithm of FIG. 13 is simultaneously performed in multiple program threads, with each of those threads corresponding to a separate downstream QAM channel. In block 401, AN controller 70 determines if it is time to create a new MAP message. If not, AN controller 70 proceeds directly to block 403, which is discussed below. If so, AN controller 70 instead proceeds to block 402 and formulates the MAP message. In formulating that MAP message, AN controller 70 checks database 89 for instructions from packet processing node 9 (e.g., as discussed in connection with blocks 375 and 376 of FIG. 12) and for BEF flow grants determined by AN controller 70 (discussed above in connection with blocks 372 and 373 of FIG. 12). After formulating the MAP message, AN controller 70 places the MAP message into the management message queue for the appropriate downstream QAM channel. As part of block 402 of FIG. 13, AN controller 70 also sends an interrupt to a separate program thread discussed below in connection with FIG. 14.

[90] In block 403, AN controller 70 then determines if it is time for generation of another type of system management message. As indicated above, RNG-REQ messages are sent by subscriber devices on a DOCSIS RF interface, with RNG-RSP messages being sent in response. Other types of system management messages include SYNCH messages to communicate a system timestamp, messages instructing a subscriber device to tune to another (or additional) QAM channel(s), etc. If it is time for the type of system management message corresponding to block 403, the message is formulated and placed into the management message queue for the appropriate downstream QAM channel (block 404). As part of block 404 of FIG. 13, AN controller 70 also sends an interrupt to a separate program thread discussed below in connection with FIG. 14. The algorithm of FIG. 13 continues for

additional types of management messages generated by AN controller 70, as represented generally by a vertical ellipsis and blocks 405 and 406.

[91] FIG. 14 is a flow chart showing operations by AN controller 70 to transmit on a downstream QAM channel. In at least some embodiments, the algorithm of FIG. 14 is simultaneously performed in multiple program threads, with each of those threads corresponding to a separate downstream QAM channel. In block 421, AN controller 70 de-queues the next message in the management message queue for that QAM channel. The management message is sent to the QAM modulator/upconverter in block 421 for transmission across interface 22. In block 423 AN controller 70 determines if there are additional messages in the management message queue. If so, AN controller 70 returns on the "yes" branch to block 422. Otherwise, AN controller 70 proceeds on the "no" branch to block 424. In block 424 AN controller 70 de-queues the next packet in the packet queue for the QAM channel. The packet is forwarded to the QAM modulator/upconverter in block 425. AN controller 70 then proceeds to block 426 and determines if it has received one of the interrupts described in connection with FIG. 13. If not, AN controller 70 returns to block 424 on the "no" branch. If an interrupt has been received, AN controller 70 returns to block 421 on the "yes" branch. If there is no data in the data packet queue for a QAM channel when AN controller 70 reaches block 424 in the algorithm of FIG. 14, an empty "pad" data packet is forwarded to the QAM modulator/upconverter.

[92] FIG. 15 is a block diagram showing, in generally schematic form, hardware for packet processing node 9 according to at least some embodiments. Packet processing node 9 includes a first hardware interface 501 that provides a physical connection to network backbone 15 and a second hardware interface 504 providing a physical connection to the medium over which packet processing node 9 communicates with access nodes 4 through 6. In at least some embodiments, hardware interfaces 501 and 504 are Gigabit Ethernet cards in communication with a separate interface to a fiber optic link (not shown). Packet processing node 9 further includes memory 502 for storing instructions and data and a processor 503 for executing instructions and controlling operation of packet processing node 9. Although a

single block is shown for memory 502 and a single block shown for processor 503, memory and computational operations of packet processing node 9 could respectively be distributed across multiple memory devices and multiple processors located within packet processing node 9. Memory 502 may include volatile and non-volatile memory and can include any of various types of storage technology, including one or more of the following: read only memory (ROM) modules, random access memory (RAM) modules, magnetic tape, magnetic discs, optical disk, flash memory, and EEPROM memory. Processor 503 may be implemented with any of numerous types of devices, including but not limited to one or more general purpose microprocessors, one or more application specific integrated circuits, one or more field programmable gate arrays, and combinations thereof. In at least some embodiments, processor 503 carries out operations described in connection with PPN controller 30 of FIG. 4 according to machine readable instructions stored in memory 502 and/or stored as hardwired logic gates within processor 503. In at least some embodiments, memory 502 carries out operations described in connection with database 33, queue 31, and other queues described in connection with FIG. 4.

[93] FIG. 16 is a block diagram showing hardware for access node 6 according to at least some embodiments. Access nodes 4 and 5 are similar. Access node 6 includes a first hardware interface 521 that provides a physical connection to the medium over which access node 6 communicates with packet processing node 9. In at least some embodiments, hardware interface 521 is a Gigabit Ethernet card in communication with a separate interface to a fiber optic link (not shown). QAM modulator/upconverters 69-1 through 69-*p* and QAM demodulator/downconverters 80-1 through 80-*k* form a hardware interface providing a physical connection to the medium over which access node 6 communicates with subscriber devices in its access sub-network. In some embodiments, either or both of *p* and *k* could be 16, 32, 64 or more, or some other value. QAM modulator/upconverters and QAM demodulator/downconverters are known in the art and thus not described further herein. QAM modulator/upconverters 69-1 through 69-*p* output signals that emanate from a single physical output port 526 and QAM demodulator/downconverters 80-1 through 80-*k* receive signals through a single physical input port 527. Access node 6 node further includes

memory 522 for storing instructions and data and a processor 523 for executing instructions and controlling operation of access node 6. Although a single block is shown for memory 522 and a single block shown for processor 523, memory and computational operations of access node 6 could respectively be distributed across multiple memory devices and multiple processors located within access node 6. Memory 522 and processor 523 can be implemented with the same types of technologies described in connection with memory 502 and processor 503, respectively, of FIG. 15. In at least some embodiments, processor 523 carries out operations described in connection with AN controller 70 of FIG. 5 according to machine readable instructions stored in memory 522 and/or stored as hardwired logic gates within processor 523. In at least some embodiments, memory 522 carries out operations described in connection with database 89, queue 71, and other queues described in connection with FIG. 5.

[94] FIG. 17 is a diagram of a subscriber network 600 according to at least some additional embodiments. Network 600 includes an access network that includes a packet processing node 609 similar to packet processing node 9 (of FIGS. 3 and 15), as well as access nodes 607, 608 and 699. Packet processing node 609 communicates with a network backbone 615 over an interface 619 (e.g., one or more 40 Gbps or 100 Gbps Ethernet interfaces) via one or more routers 611. Packet processing node 609 also communicates over an interface 621 (e.g., one or more Gigabit Ethernet links) with access nodes 607, 608 and 699 in the domain of packet processing node 609. Access node 607 is similar to access node 6 of FIGS. 3 and 16, and communicates with a plurality of subscriber devices (not shown) in its access sub-network over an interface 614. Each of access nodes 607, 608 and 699 conducts upstream and downstream communications with devices at multiple different subscriber premises (e.g., multiple homes and/or offices) in the access sub-network of that access node. Except as may be described below for certain embodiments, packet processing node 609 and access node 607 operate in a manner similar to that described above in connection with FIGS. 3-16.

[95] Unlike the embodiments of FIGS. 3-16, some access nodes in the domain of packet processing node 609 communicate with subscriber devices in access sub-networks that

employ non-DOCSIS interfaces and/or that may rely on non-HFC media. For example, access node 608 communicates with subscriber devices (not shown) in its access sub-network over a passive optical network (PON) interface 615. Access node 699 communicates with subscriber devices (not shown) in its access sub-network over a wireless interface 616. Additional access nodes could be included to communicate with subscriber devices over other types of interfaces. Although FIG. 17 shows packet processing node 609 serving a domain having only three access nodes, a single packet processing node could serve a domain having more or fewer access nodes.

[96] Similar to network 1 of FIG. 3, packet processing node 609 and access nodes 607, 608 and/or 699 may be separated by over one mile (as shown in FIG. 17), or may be located within the same room or within the same chassis. Packet processing node 609 may also configure and manage access nodes 607, 608 and 699 and other access nodes in the domain of packet processing node 609. Packet processing node 609 may proxy IPDR (Internet Protocol Data Record), SNMP (Simple Network Management Protocol) and/or provisioning functions to one or more servers 602, but may maintain a local database of information regarding subscriber premises devices in the premises of subscribers within the access sub-networks of access nodes 607, 608 and 699. As with packet processing node 9 and access nodes 4 through 6 in the embodiment of FIG. 3, packet processing node 609 and access nodes 607, 608 and/or 699 could communicate over various types of interfaces. When integrated into a common chassis, a packet processing node and access nodes can communicate over an internal backplane. When a packet process node and access nodes are in separate chassis or in separate locations, communications could, e.g., be over one or more 40 Gbps or 100 Gbps Ethernet interfaces.

[97] FIG. 18 is a block diagram showing hardware for packet processing node 609 according to at least some embodiments. Packet processing node 609 includes one or more first hardware interfaces 701 that provide a physical connection to network backbone 615 and one or more second hardware interfaces 704 providing a physical connection to the medium over which packet processing node 609 communicates with access nodes 607 through 699.

In at least some embodiments, hardware interfaces 701 and 704 are Gigabit Ethernet cards that facilitate communication over a fiber optic link. Packet processing node 609 further includes memory 702 for storing instructions and data and a processor 703 for executing instructions and controlling operation of packet processing node 609. Although a single block is shown for memory 702 and a single block shown for processor 703, memory and computational operations of packet processing node 609 could respectively be distributed across multiple memory devices and multiple processors. Memory 702 and processor 703 can be implemented with the same types of technologies described in connection with memory 502 and processor 503, respectively, of FIG. 15. In at least some embodiments, processor 703 carries out the herein described operations of packet processing node 609 according to machine readable instructions stored in memory 702 and/or stored as hardwired logic gates within processor 703. Memory 702 may also be used to store data used by processor 703 as part of carrying out various operations. For example, packet processing node 609 may also maintain a local database in memory 702 that stores MAC and IP address data for all subscriber premises devices (including computers, STBs and other customer equipment) in the access network served by packet processing node 609, as well as information regarding the access sub-network in which each subscriber premises device is located. That local database may also store data indicating services that individual subscribers are authorized to receive, as well as provisioning data and other information regarding individual subscriber premises devices.

[98] FIG. 19 is a block diagram showing hardware for access node 607 according to at least some embodiments. Access node 607 includes one or more hardware interfaces 711 that provide a physical connection to the medium over which access node 607 communicates with packet processing node 609. In at least some embodiments, hardware interface 711 includes one or more Gigabit Ethernet cards that facilitate communication over a fiber optic link. QAM modulators/upconverters 714-1 through 714-*p* and QAM demodulators/ downconverters 715-1 through 715-*k* form a hardware interface providing a physical connection to the HFC medium over which access node 607 communicates with subscriber premises devices in the access node 607 access sub-network. QAM modulator/upconverters

714-1 through 714-*p* are similar to QAM modulator/upconverters 69-1 through 69-*p* and output QAM modulated data over a port 716. Processor 713 encapsulates data according to the MAC protocol portion of the applicable standards, for example, HFC standard(s) (e.g., DOCSIS), applicable to access node 607 and then causes one or more of QAM modulator/upconverters 724-1 through 724-*p* to transmit that MAC-encapsulated data according to the physical layer (PHY) requirements of the applicable HFC standard(s). QAM demodulators/downconverters 715-1 through 715-*k* are similar to QAM demodulator/ downconverters 80-1 through 80-*k* described in connection with FIG. 16 and receive QAM modulated communications over a port 717. In particular, QAM modulated upstream RF communications from subscriber premises received via port 717 are demodulated in demodulator/downcoverters 715-1 through 715-*k*. Processor 713 analyzes MAC headers in demodulated upstream data packets and processes those packets as required. For example, system messages may be processed in accordance with the HFC standards applicable to access node 607, outbound user data may be forwarded to packet processing node 609, etc.

[99] In some embodiments, an access node such as access node 607 may employ other types of modulation for upstream and/or downstream communications. Examples of such other modulation types include, but are not limited to, QPSK modulation and modulation used as part of Orthogonal Frequency Division Multiplexing (OFDM). In embodiments using other types of modulation for downstream communications, one or more of QAM modulators/upconverters 714-1 through 714-*p* is replaced with a modulator/upconverter appropriate to the other modulation type. In embodiments using other types of modulation for upstream communications, one or more of QAM demodulators/downconverters 715-1 through 715-*k* is replaced with a demodulator/downconverter appropriate to the other modulation type.

[100] Access node 607 node further includes memory 712 for storing instructions and data and a processor 713 for executing instructions and controlling operation of access node 607. Memory and computational operations of access node 607 could respectively be distributed across multiple memory devices and multiple processors. Memory 712 and processor 713

can be implemented with the same types of technologies described in connection with memory 502 and processor 503, respectively, of FIG. 15. In at least some embodiments, processor 713 carries out the herein described operations of access node 607 according to machine readable instructions stored in memory 712 and/or stored as hardwired logic gates within processor 713. Memory 712 may also be used to store data used by processor 713 as part of carrying out various operations.

[101] FIG. 20 is a block diagram showing hardware for access node 608 according to at least some embodiments. Access node 608 includes one or more hardware interfaces 721 (e.g., one or more Gigabit Ethernet cards that facilitate communication over a fiber optic link) that provide a physical connection to the medium over which access node 608 communicates with packet processing node 609. Memory 722 and processor 723 of access node 608 are similar to memory 712 and processor 713 of access node 607 (FIG. 19), but contain data and execute instructions to carry out operations of access node 608 in accordance with the transport technology and corresponding standards applicable to access node 608. For example, access node 608 may employ a fiber-to-the-home (FTTH) digital optical transport technology and operate in accordance with Ethernet PON (EPON) standards (described, e.g., by IEEE 802.3-2008 and/or other IEEE standards). Access node 608 might alternatively be operated in accordance with Gigabit PON (GPON) standards (as described, e.g., by ITU-T G.984 and/or other ITU-T standards) or other protocols. Access node 608 may communicate with subscriber premises over optical links that terminate at optical network terminals (ONTs) located at those subscriber premises, with those ONTs performing optical/electrical conversion for communication of data to and from routers, computers, STBs and other customer equipment (CE) devices in the subscriber premises.

[102] Unlike access node 607, which creates separate physical channels through different frequencies for QAM modulation and/or through time or code multiplexing within a particular frequency, access node 608 creates separate physical channels by employing different optical wavelengths and/or through time multiplexing within a particular wavelength. Each of optical transmitter/receiver (TX/RX) units 724-1 through 724-*p*

includes a hardware interface providing a physical connection to the FTTH fiber medium over which access node 608 communicates with subscriber premises devices in the access node 608 access sub-network. Each of TX/RX units 724-1 through 724-*p* includes a laser and photodiode configured to perform upstream and downstream communications over a common physical interface. Access node 608 communicates downstream data to subscriber premises ONTs or other termination points using the transmission lasers in one or more of TX/RX units 724-1 through 724-*p*. Processor 723 encapsulates data according to the MAC protocol portion of the PON standard(s) applicable to access node 608 and then causes one or more of the lasers in TX/RX units 724-1 through 724-*p* to optically transmit that MAC-encapsulated data according to the physical layer (PHY) requirements of the applicable PON standard(s). Some of the lasers in TX/RX units 724-1 through 724-*p* may operate on different wavelengths and be combined for transmission through a single fiber (not shown). Others of those lasers may transmit over separate fibers. Access node 608 receives upstream data from subscriber premises ONTs through receiving photodiodes at TX/RX units 724-1 through 725-*p*. Upstream optical communications to access node 608 may use different wavelengths and be over the same fibers used for downstream communications to subscribers. The photodiodes at TX/RX units 724-1 through 725-*p* convert optical signals into electrical signals and provide the upstream data packets communicated by those signals to processor 723. Processor 723 analyzes MAC headers in received upstream data packets and processes those packets as required. For example, system messages may be processed in accordance with the PON standards applicable to access node 608, outbound user data may be forwarded to packet processing node 609, etc.

[103] FIG. 21 is a block diagram showing hardware for access node 699 according to at least some embodiments. Access node 699 includes one or more hardware interfaces 731 (e.g., one or more Gigabit Ethernet cards that facilitate communication over a fiber optic link) that provide a physical connection to the medium over which access node 699 communicates with packet processing node 609. Memory 732 and processor 733 of access node 699 are similar to memory 712 and processor 713 of access node 607 (FIG. 19), but contain data and execute instructions to carry out operations of access node 699 in accordance with the

transport technology and corresponding standards applicable to access node 699. For example, access node 699 may employ a wireless transport technology and operate in accordance with WiMAX standards (described, e.g., by IEEE 802.16 and/or other IEEE standards) or in accordance with another set of wireless standards. Access node 699 may communicate with subscriber premises over wireless links that terminate at wireless transceivers located at those subscriber premises, with those transceivers demodulating downstream data for communication to CE devices in the subscriber premises and modulating data received from those CE devices for upstream communication to access node 699.

[104] Access node 699 creates separate physical channels, in both the upstream and downstream directions, by using multiple different frequencies and/or through time (and/or code) multiplexing. Transmitters (TX) 734-1 through 734-*p* and receivers (RX) 735-1 through 735-*k* form a hardware interface providing a physical connection to the wireless medium over which access node 699 communicates with subscriber premises devices in the access node 699 access sub-network. Access node 699 communicates downstream data to subscriber premises transceivers (which in turn communicate with CE devices at a premises) using one or more of transmitters 734-1 through 734-*p* to modulate data onto RF frequencies and send that modulated data over one or more antennas 736-1 through 736-*p*. Each of transmitters 736-1 through 736-*p* communicates over a different frequency range. Processor 733 encapsulates data according to the MAC protocol portion of the wireless standard(s) applicable to access node 699 and then causes one or more of transmitters 734-1 through 734-*p* to modulate and transmit that MAC-encapsulated data according to the physical layer (PHY) requirements of the applicable wireless standard(s).

[105] Access node 699 receives upstream data from subscriber premises using one or more of receivers 735-1 through 735-*k* and antennas 737-1 through 737-*k*. Each of receivers 735-1 through 735-*k* is tuned to a different frequency range. Modulated upstream wireless communications from transceivers at various subscriber premises received via antennas 737-1 through 737-*k* are demodulated in receivers 735-1 through 735-*k*. Processor 733 analyzes

MAC headers in demodulated upstream data packets and processes those packets as required. For example, system messages may be processed in accordance with the wireless standards applicable to access node 699, outbound user data may be forwarded to packet processing node 609, etc.

[106] Although separate transmitter and receiver blocks are shown in FIG. 21, some wireless access nodes may employ transceiver components able to both transmit and receive. Similarly, a more than one transmitter or receiver (or transceiver) may share a common antenna.

[107] In at least some embodiments, packet processing node 609 receives IP packets containing service data from network backbone 615 over interface 619 and forwards those service data IP packets to subscribers via one or more of access nodes 607, 608 and 609. Packet processing node 609 similarly receives data from subscribers via access nodes 607, 608 and 609 and forwards that data to the appropriate destinations. As with packet processing node 9 of FIG. 3, packet processing node 609 may also establish and tear down service flows and perform system management operations.

[108] FIG. 22 is a flow chart showing operations performed by packet processing node 609, in at least some embodiments, upon receiving a service data IP packet that is destined for one or more subscriber premises. FIG. 22 assumes that a service flow has already been established. Although service flows were discussed in the context of DOCSIS service flows in connection with the embodiments of FIGS. 3-16, the concept of service flows is not unique to DOCSIS. In the embodiment of FIGS. 17-24, various service flows are also used to associate different qualities of service and/or priorities with various types of service data. In block 801, packet processing node 609 receives a service data IP packet over interface 619. That packet may be destined for a single CE device at a single subscriber premises, may be destined for multiple CE devices within a single premises, or may be destined for multiple CE devices located at multiple premises.

[109] In block 802, packet processing node 609 determines the service flow with which the received service data is associated. For example, packet processing node 609 may establish one or more types of best efforts service flows for service data related to subscriber web surfing and emails, one or more types of guaranteed bandwidth service flows (e.g., an unsolicited grant service flow) for telephony service data, still other types of service flows for multicast IPTV service data, yet other service flows for narrowcast IPTV service data, etc. In at least some embodiments, packet processing node determines a service flow associated with an IP packet based on the source address and/or destination address in the IP packet header. Packet processing node 609 also determines the destination(s) for the service data IP packet in block 802. As part of this determination, packet processing node also determines the access node(s) serving the destination devices(s) to which the service data will be forwarded.

[110] In block 803, packet processing node 609 encapsulates the service data IP packet within a frame according to the network side data interface requirements of the access node serving the subscriber premises to which the service data will be sent. For example, a WiMAX access node may receive data from a network according to the WiMAX R6 interface requirements, an EPON access node may receive data from a network according to an EPON interface requirements (as defined, e.g., by IEEE 802.3-2008 or by another applicable standard), and a DOCSIS access node may receive data from a network according to DOCSIS interface requirements (as defined, e.g., by the aforementioned DOCSIS Downstream RF Interface Specification and DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification). If the service data is multicast data to be sent to subscriber premises located in the access sub-networks of more than one access node, packet processing node 609 may create a separate frame for each of those access nodes encapsulating the service data IP packet according to the individual network side interface requirements of those access nodes.

[111] As part of encapsulating the service data according a specific access node's network side interface, packet processing node 609 may include data in that encapsulation that links the service data to the service flow associated with that service data. Alternatively, packet processing node 609 may mark the service data (e.g., by setting one or more IP header flags

of the service data) so as to link the service data with its associated service flow. As yet another alternative, packet processing node 609 may have already instructed one or more of access nodes 607, 608 and 699 that service data having a particular value in an IP header field (e.g., IP source address), and/or in a field of a frame encapsulating a service data IP packet, is linked with a particular service flow.

[112] In block 804, packet processing node 609 further encapsulates the encapsulated service data IP packet into an Ethernet frame and forwards that Ethernet frame over interface 621 to the appropriate access node. If there are multiple frames sending the same service data to different access nodes, each is placed into a separate Ethernet frame and transmitted across interface 621. In some embodiments, packet processing node 609 may use framing other than Ethernet framing (e.g., a layer 2 tunneling protocol framing) for communication with access nodes 607, 608 and 699.

[113] In at least some embodiments, the data forwarded by packet processing node 609 in block 804 does not specify a physical channel that an access node should use to forward that data to a subscriber. In some such embodiments, an access node is allowed to allocate physical channels in response to load conditions at that access node. In other such embodiments, an access node is allowed to allocate physical channels subject to certain restraints imposed by packet processing node 609 or by another network element. In still other such embodiments, packet processing node 609 may send separate signaling messages to an access node to allocate different physical channels among various service flows.

[114] FIG. 23 is a flow chart showing operations performed by an access node in response to receiving a service data IP packet forwarded by packet processing node 609 in block 804 of FIG. 22. Each of access nodes 607, 608 or 699 will generally perform the same types of operations in connection with receipt of service data, but according to the protocols associated with the transport technology and corresponding standards applicable to that access node. FIG. 23 is thus generic to any of access nodes 607, 608 or 699. In block 851 the access node receives the Ethernet frame from packet processing node 609 across interface

621. In block 852, the access node identifies the service flow associated with the service data IP packet and the destination(s) for that service data. In block 853, the access node determines the appropriate physical channel(s) that should be used to forward the service data IP packet to the appropriate destination(s) in one or more subscriber premises. For example, and in accordance with the protocol utilized by the access node, the access node may have previously instructed some subscriber premises devices to receive data in one physical channel, instructed other devices to receive data in a different physical channel, etc. As another example, an access node may designate one physical channel for certain types of data and/or service flows and designate a different physical channel for other types of data or service flows.

[115] In block 854, the access node encapsulates the service data IP packet using a MAC header and/or other encapsulation of the access network protocol employed by the access node. For example, access node 607 might encapsulate the service data using a DOCSIS MAC header. As other examples, access node 608 might encapsulate the service data using an EPON MAC header and access node 699 might encapsulate the service data using a WiMAX MAC header.

[116] In block 855, the MAC header-encapsulated service data is transmitted to the subscriber premises. Depending on the access sub-network protocols and/or transport technology the access node may queue downstream data based on service flow and/or physical channel. If the service data is being multicast to multiple subscriber premises receiving data over different physical channels, the access node may replicate MAC-encapsulated service data for those multiple physical channels.

[117] FIG. 24 shows another embodiment of a network 600' having different types of access nodes. Network 600' is generally similar to network 600 of FIG. 17, except that access node 607 also communicates with a femtocell WiMAX access node 698. Access node 607 forwards service data to some subscribers directly through the access network of access node 607 and forwards service data to other subscribers via link 697 to access node 698.

Service data sent over link 697 is MAC-encapsulated, which encapsulation is removed by access node 698 and replaced with WiMAX MAC encapsulation before being transmitted to subscribers over the wireless interface of access node 698. In some embodiments, service flow information can be provided to access node 698 by separate signaling messages. In other embodiments, access node 698 can maintain data that maps service flow identifiers in a MAC header to WiMAX service flows. Although FIG. 24 only shows an access node communicating with subscribers via an intermediate WiMAX access node, other embodiments include other combinations of one type of access node communicating with subscriber premises via an intermediate access node of a different type.

[118] As previously indicated in connection with FIG. 17, additional access nodes could be included in network 600 to communicate with subscriber devices over different types of access network interfaces. This is also the case for access network 600' and is represented generically in FIG. 24 by access node 900. Access node 900 communicates with a plurality of subscriber devices over an interface 901 that corresponds to the transport technology and protocol(s) implemented by access node 900. As with access nodes 607, 608 and 699, access node 900 may include a processor controlling operation of the access node, a memory storing data and/or instructions for operation of the access node, one or more hardware interfaces for communicating with packet processing node 609, and one or more hardware interfaces for creating separate physical channels for communications with subscriber devices according to the MAC and other protocols applicable to the access sub-network of access node 900. Access node 900 may carry out operations similar to those described in connection with FIG. 23. Packet processing node 609 may interface with access node 900 according to FIG. 22 and otherwise in a manner similar that in which packet processing node 609 communicates with other access nodes, but using an interface applicable to the transport technology and protocols implemented by access node 900.

[119] As previously indicated, embodiments such as are shown in FIGS. 3, 17 and 24 could be implemented using various deployment scenarios. As one example, a packet processing node and one or more access nodes could be deployed in the same location, e.g., a hub where

fiber optic cables are terminated. In embodiments such as in FIG. 17 where wireless service is provided, the packet processing node and access node(s) could be located in base stations. Alternatively, a packet processing node could be deployed in a central location relative to the access nodes, e.g., a packet processing node could be located in a headend and access nodes could be located in the same headend for fiber or distribution cable that is locally terminated from the headend. Additional access nodes could be located in connected hubs for fiber or distribution cable terminating at those hubs. As yet another alternative, access nodes could be miniaturized and hardened and located in service group nodes. In such a scenario the fiber interface could be changed from an analog modulated fiber interface (as is currently used in existing architectures such as in FIG. 1) to a digital interface (e.g., 10 Gbps downstream and 1 Gbps upstream). The service group node would then contain the various components of the access node, such as the downstream narrowcast QAM modulators, upstream demodulators, and other ancillary interface components. In some embodiments such as in FIG. 17, a packet processing node could be located in a central location, with HFC access nodes located in service group nodes, PON access nodes located in hubs with (or as part of) optical loop termination (OLT) equipment, and with wireless access nodes located in base stations.

[120] In some embodiments, all narrowcast traffic in an access network could be transmitted via a packet processing node and access nodes. This narrowcast traffic could include all HSD, voice and narrowcast video (e.g., unicast over MPEG-TS (MPEG transport stream) for VOD or "start over" service, multicast for SDV over MPEG-TS, and IP). The packet processing node and access nodes could handle each of those flows according to its specific needs and in a correspondingly efficient manner. For example, the packet processing node may receive IP-frame encapsulated data for voice, video and HSD services. The packet processing node could then multiplex SPTS video streams into MPTS video streams and MPEG-TS encapsulate those MPTS streams. The packet processing node could process HSD and voice data traffic according to one or more applicable protocols, such as DOCSIS protocols. The packet processing node could encapsulate video over IP streams into DOCSIS frames, but at the same time avoid "high-touch" packet manipulation associated with DOCSIS HSD traffic. As indicated above, a processing node and access nodes could be

implemented in a single chassis or in separate chassis. In either case, significantly denser QAM-channel-to-RF-port ratios (e.g., 16, 32, 64 or more) could be implemented, thereby facilitating assignment of some QAM channels of an RF port to DOCSIS streams and other QAM channels of that port to video over MPEG-TS. As but one example, a portion of the QAM channels on a single downstream RF port could be assigned as DOCSIS QAM channels for HSD, voice and IP video, and other QAM channels on that port assigned to video over MPEG-TS (e.g., VOD, SDV, "start over" service).

[121] Transmitting narrowcast traffic via a packet processing node and access nodes can allow consolidation of flows for all (or many) services into fewer (and unified) interfaces. Such an architecture also facilitates leveraging of increased channel-to-port density, can increase reliability, and can lower capital and operating expenditures. Such an architecture can also allow simpler and more streamlined operations, alleviate HUB/OTN constraints, and allow continued provision of existing services and/or use of existing subscriber devices.

[122] Although various portions of the foregoing description refer to multiple QAM channels used to communicate between access nodes and subscriber devices, in some embodiments other types of modulation techniques are used to communicate between access nodes and subscriber devices. In some such embodiments, multiple channels according to such other modulation techniques may likewise be combined for transmission via a single physical port.

[123] The foregoing description of embodiments has been presented for purposes of illustration and description. The foregoing description is not intended to be exhaustive or to limit embodiments of the present invention to the precise form disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of various embodiments. The embodiments discussed herein were chosen and described in order to explain the principles and the nature of various embodiments and their practical application to enable one skilled in the art to utilize the present invention in various embodiments and with various modifications as are suited to the particular use contemplated.

The features of the embodiments described herein may be combined in all possible combinations of methods, apparatus, modules, systems, and computer program products. Although certain processes having various steps have been described herein, other embodiments may include processes in which one or more of said steps are omitted or are performed in a different order relative to other steps. Similarly, other embodiments may include devices in which one or more components described herein are omitted or are rearranged relative to other components.

CLAIMS:

1. A method comprising:

receiving a data packet at a packet processing node of a network, the packet processing node being in communication with a first group of user devices through a first access node employing a first transport technology to communicate with the user devices in the first group, the packet processing node also being in communication with a second group of user devices through a second access node employing a second transport technology to communicate with the user devices in the second group, the first and second transport technologies being different;

determining at the packet processing node that the data packet is to be forwarded to the first access node; and

transmitting the data packet from the packet processing node to the first access node.

2. The method of claim 1, further comprising:

determining, at the packet processing node, a service flow associated with the data packet; and

providing information from the packet processing node to the first access node linking the data packet to the associated service flow.

3. The method of claim 1, wherein the information linking the data packet to the associated service flow is provided with the transmitted data packet.

4. The method of claim 1, wherein

the first access node employs a first media access control (MAC) protocol to communicate with the user devices in the first group,

the second access node employs a second MAC protocol to communicate with the user devices in the second group, and

the first and second MAC protocols are different.

5. The method of claim 4, further comprising:

receiving the transmitted data packet at the first access node;

encapsulating the data packet in a MAC frame according to the first MAC protocol; and

forwarding the encapsulated data packet to a destination user device of the first group.

6. The method of claim 4, further comprising:

receiving a second data packet at the packet processing node;

determining at the packet processing node that the second data packet is to be forwarded to the second access node;

transmitting the second data packet from the packet processing node to the second access node;

receiving the transmitted second data packet at the second access node;

encapsulating the second data packet in a MAC frame according to the second MAC protocol; and

forwarding the encapsulated second data packet to a destination user device of the second group.

7. The method of claim 1, wherein the first access node employs a hybrid fiber coaxial transport technology and the second access node employs a passive optical network transport technology.

8. The method of claim 1, wherein the first access node employs a hybrid fiber coaxial transport technology and the second access node employs a wireless transport technology.

9. The method of claim 1, wherein the first access node employs a passive optical network transport technology and the second access node employs a wireless technology.

10. The method of claim 1, wherein the service data packet is an internet protocol packet containing video data.

11. A method comprising:

receiving a data packet at a packet processing node of a network, the packet processing node being in communication with a first group of user devices through a first access node employing a first media access control (MAC) protocol to communicate with the user devices in the first group, the packet process node also being in communication with a second group of user devices through a second access node employing a second MAC protocol to communicate with the user devices in the second group, the first and second MAC protocols being different;

determining at the packet processing node that the data packet is to be forwarded to the first access node; and

transmitting the data packet from the packet processing node to the first access node.

12. A system comprising:

a packet processing node, the packet processing node having a first hardware interface for communication with a first portion of a network, a second hardware interface for communication with a second portion of the network, at least one memory and at least one processor configured to control operations of the packet processing node so as to cause the packet processing node to

receive a data packet over the first hardware interface,

communicate over the second hardware interface with a first group of user devices through a first access node employing a first transport technology to communicate with the user devices in the first group,

communicate over the second hardware interface with a second group of user devices through a second access node employing a second transport technology to communicate with the user devices in the second group, wherein the first and second transport technologies are different,

determine that the data packet is to be forwarded to the first access node, and

transmit the data packet to the first access node.

13. The system of claim 12, wherein the at least one processor is configured to control operations of the packet processing node so as to cause the packet processing node to

determine a service flow associated with the data packet, and

provide information to the first access node linking the data packet to the associated service flow.

14. The system of claim 12, wherein the information linking the data packet to the associated service flow is provided with the transmitted data packet.

15. The system of claim 12, wherein

the first access node employs a first media access control (MAC) protocol to communicate with the user devices in the first group,

the second access node employs a second MAC protocol to communicate with the user devices in the second group, and

the first and second MAC protocols are different.

16. The system of claim 15, further comprising:

the first access node, the first access node including a first hardware interface for communication with the second portion of the network, a first access sub-network hardware interface for communication with the first group of user devices, at least one memory and at least one processor configured to control operations of the first access node so as to cause the first access node to

receive the transmitted data packet,

encapsulate the data packet in a MAC frame according to the first MAC protocol, and

forward the encapsulated service data packet to a destination user device of the first group.

17. The system of claim 16, wherein the at least one processor of the packet processing node is configured to control operations of the packet processing node so as to cause the packet processing node to

receive a second data packet over the first hardware interface of the packet processing node,

determine that the second data packet is to be forwarded to the second access node,

transmit the second data packet to the second access node, and further comprising

the second access node, the second access node including a first hardware interface for communication with the second portion of the network, a second access sub-network hardware interface for communication with the second group of user devices, at least one memory and at least one processor configured to control operations of the second access node so as to cause the second access node to

receive the transmitted second data packet,

encapsulate the second data packet in a MAC frame according to the second MAC protocol, and

forward the encapsulated second data packet to a destination user device of the second group.

18. The system of claim 12, wherein the first access node employs a hybrid fiber coaxial transport technology and the second access node employs a passive optical network transport technology.

19. The system of claim 12, wherein the first access node employs a hybrid fiber coaxial transport technology and the second access node employs a wireless transport technology.

20. The system of claim 12, wherein the first access node employs a passive optical network transport technology and the second access node employs a wireless technology.

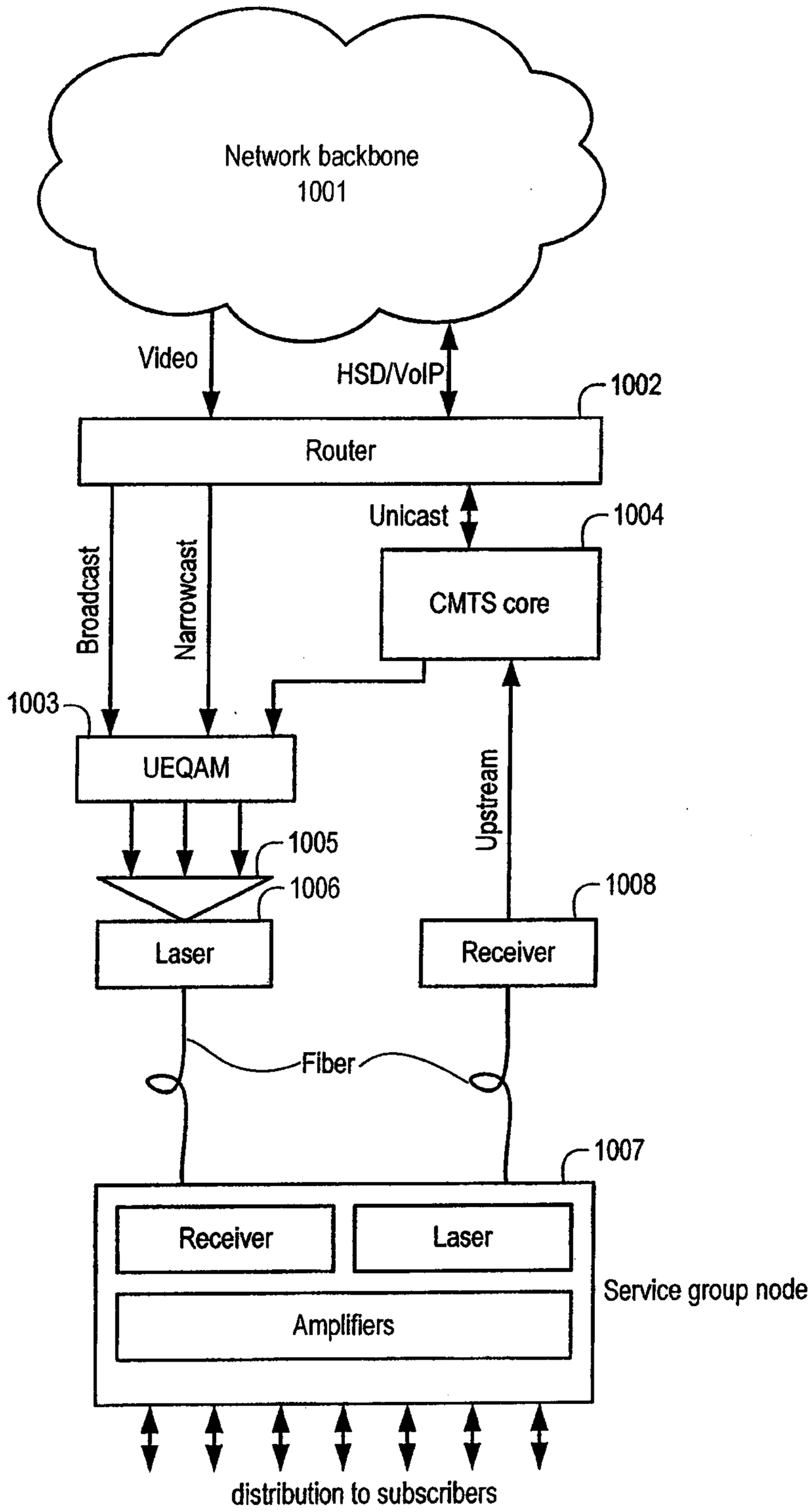

Network backbone
1001
Video
HSD/VoIP
1002
Router
Unicast
1004
CMTS core
Broadcast
Narrowcast
1003
UEQAM
Upstream
1005
1006
Laser
1008
Receiver
Fiber
1007
Receiver
Laser
Amplifiers
Service group node
distribution to subscribers

FIG. 1

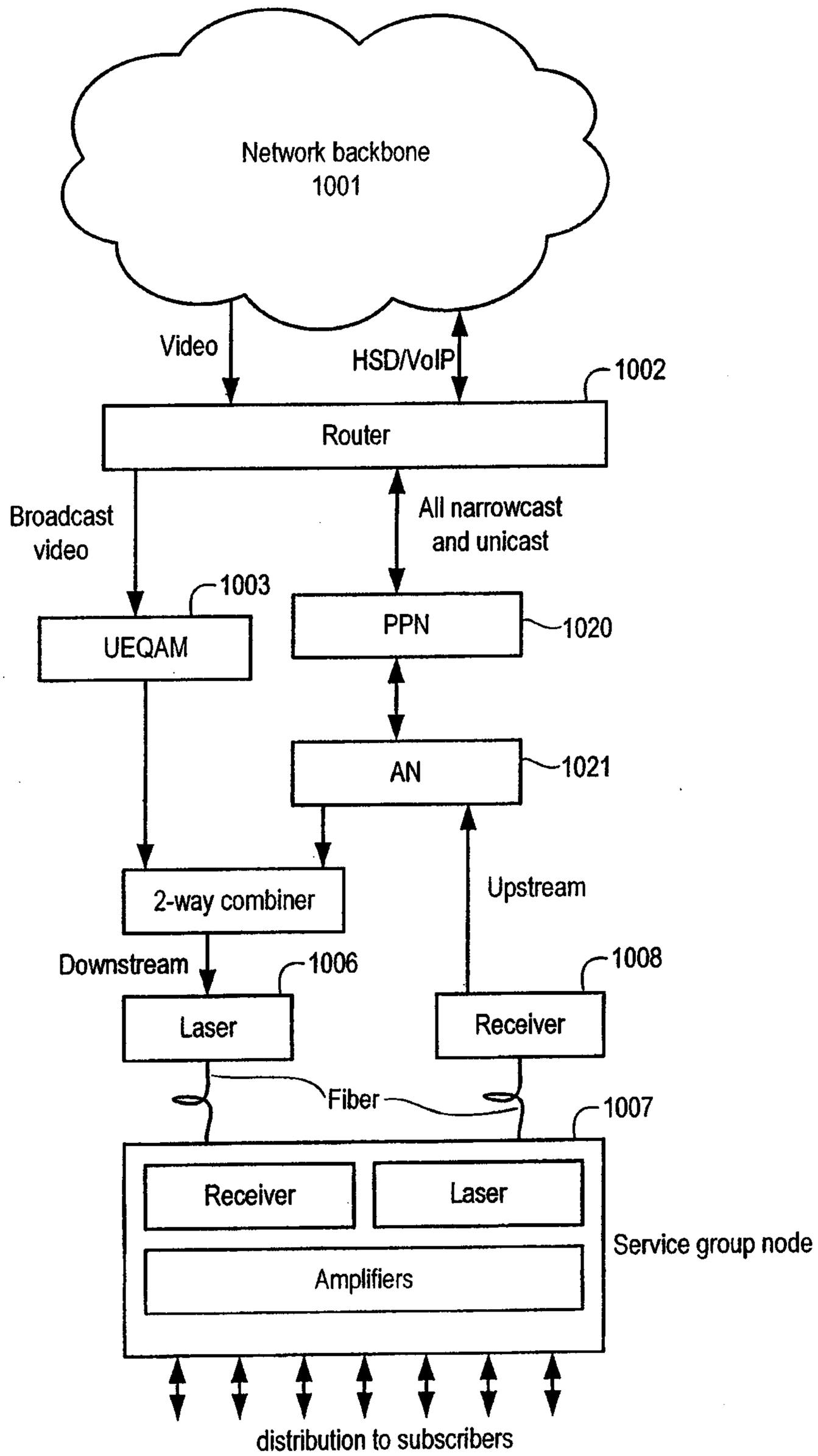

Network backbone
1001
Video
HSD/VoIP
1002
Router
Broadcast video
All narrowcast and unicast
1003
UEQAM
PPN
1020
AN
1021
2-way combiner
Upstream
Downstream
1006
Laser
1008
Receiver
Fiber
1007
Receiver
Laser
Amplifiers
Service group node
distribution to subscribers

FIG. 2

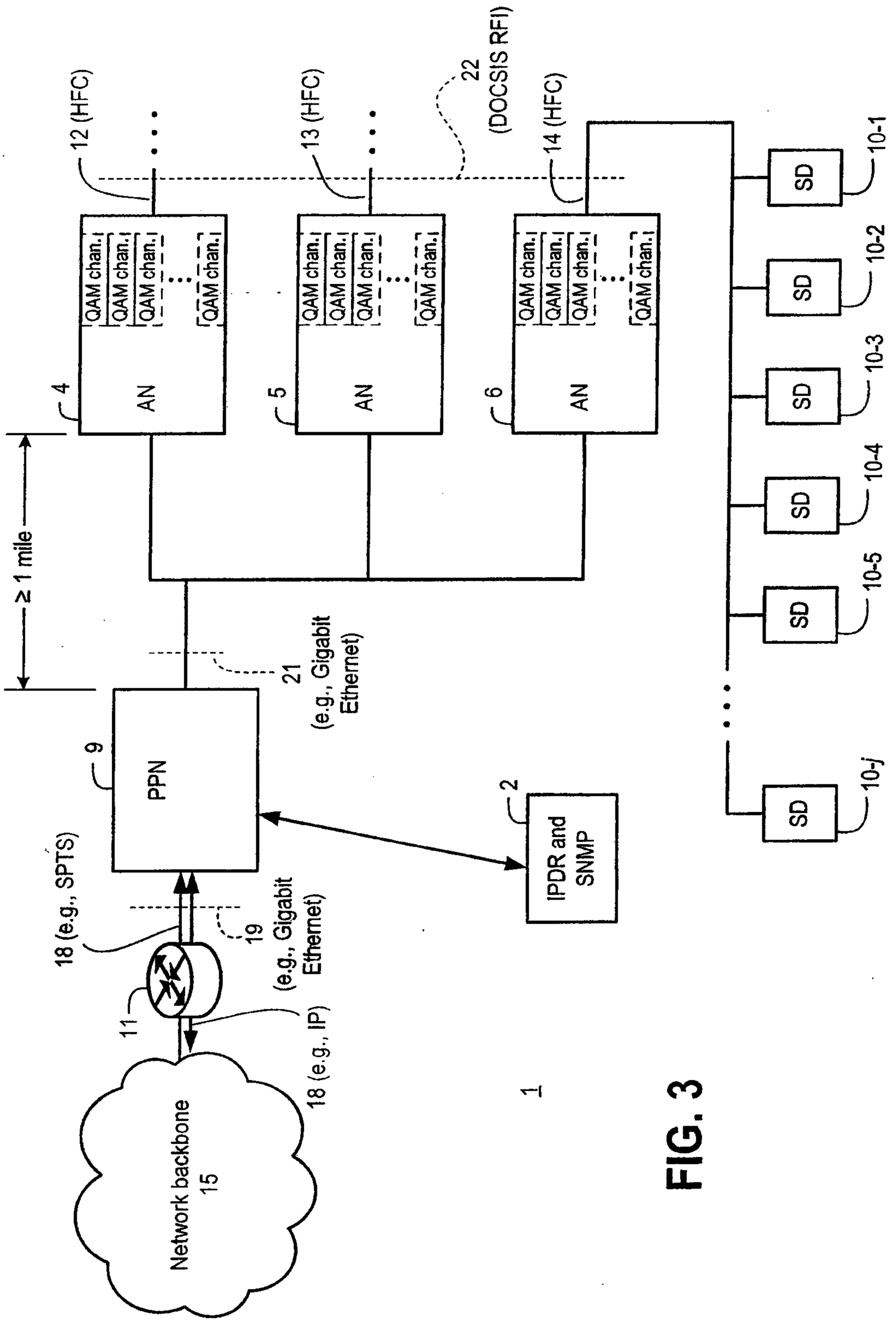

Network backbone
15
18 (e.g., IP)
11
18 (e.g., SPTS)
19
(e.g., Gigabit Ethernet)
PPN
9
IPDR and SNMP
2
21
(e.g., Gigabit Ethernet)
≥ 1 mile
4
5
6
AN
AN
AN
12 (HFC)
13 (HFC)
14 (HFC)
2
(DOCS RFI)
QAM chan.
SD
10-1
10-2
10-3
10-4
10-5
10-j

FIG. 3

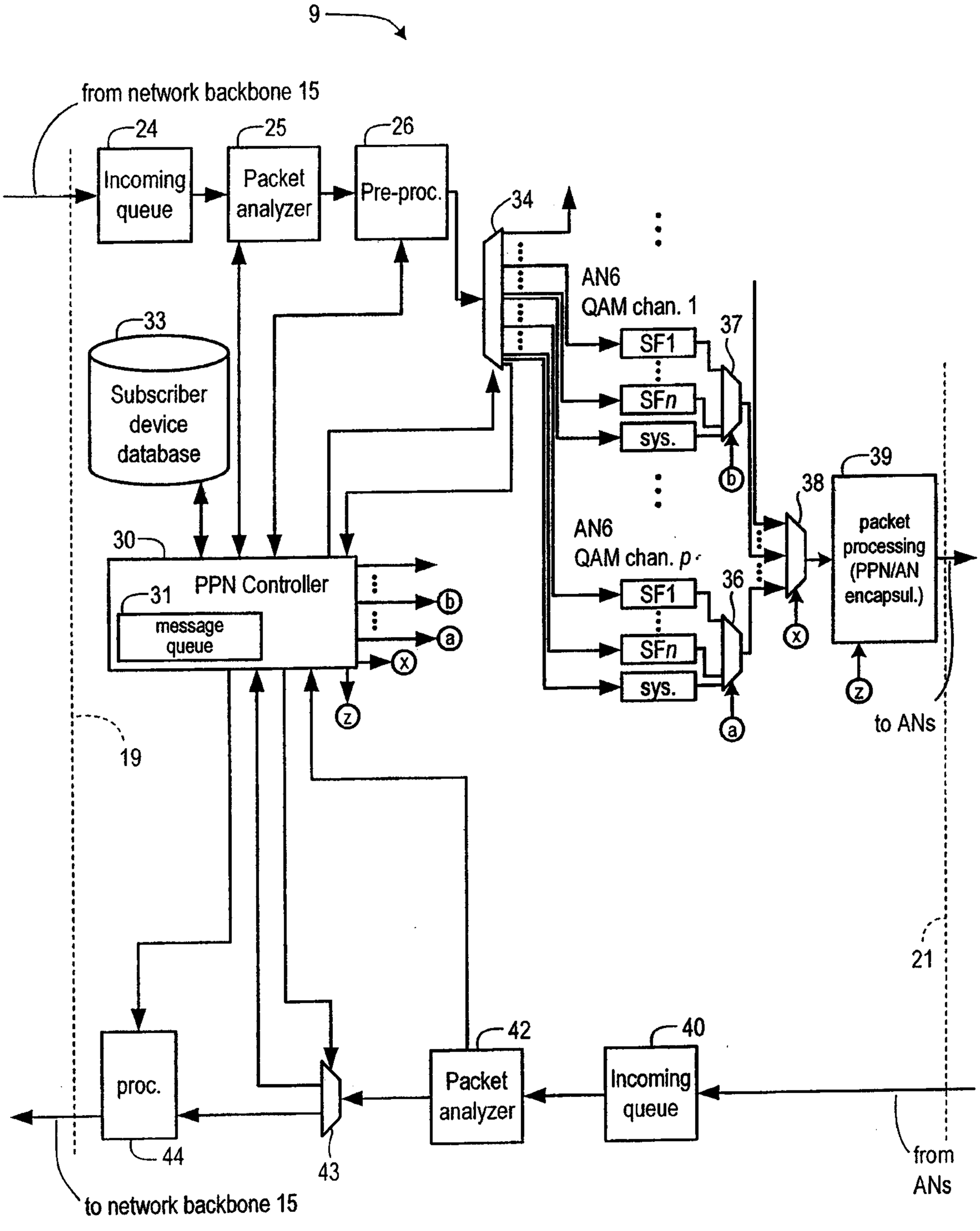

9
from network backbone 15
24
Incoming queue
25
Packet analyzer
26
Pre-proc.
34
AN6 QAM chan. 1
SF1
SFn
sys.
37
33
Subscriber device database
30
PPN Controller
31
message queue
AN6 QAM chan. p
SF1
SFn
sys.
36
38
39
packet processing (PPN/AN encapsul.)
b
a
x
z
to ANs
19
21
42
40
proc.
Packet analyzer
Incoming queue
44
43
from ANs
to network backbone 15

FIG. 4

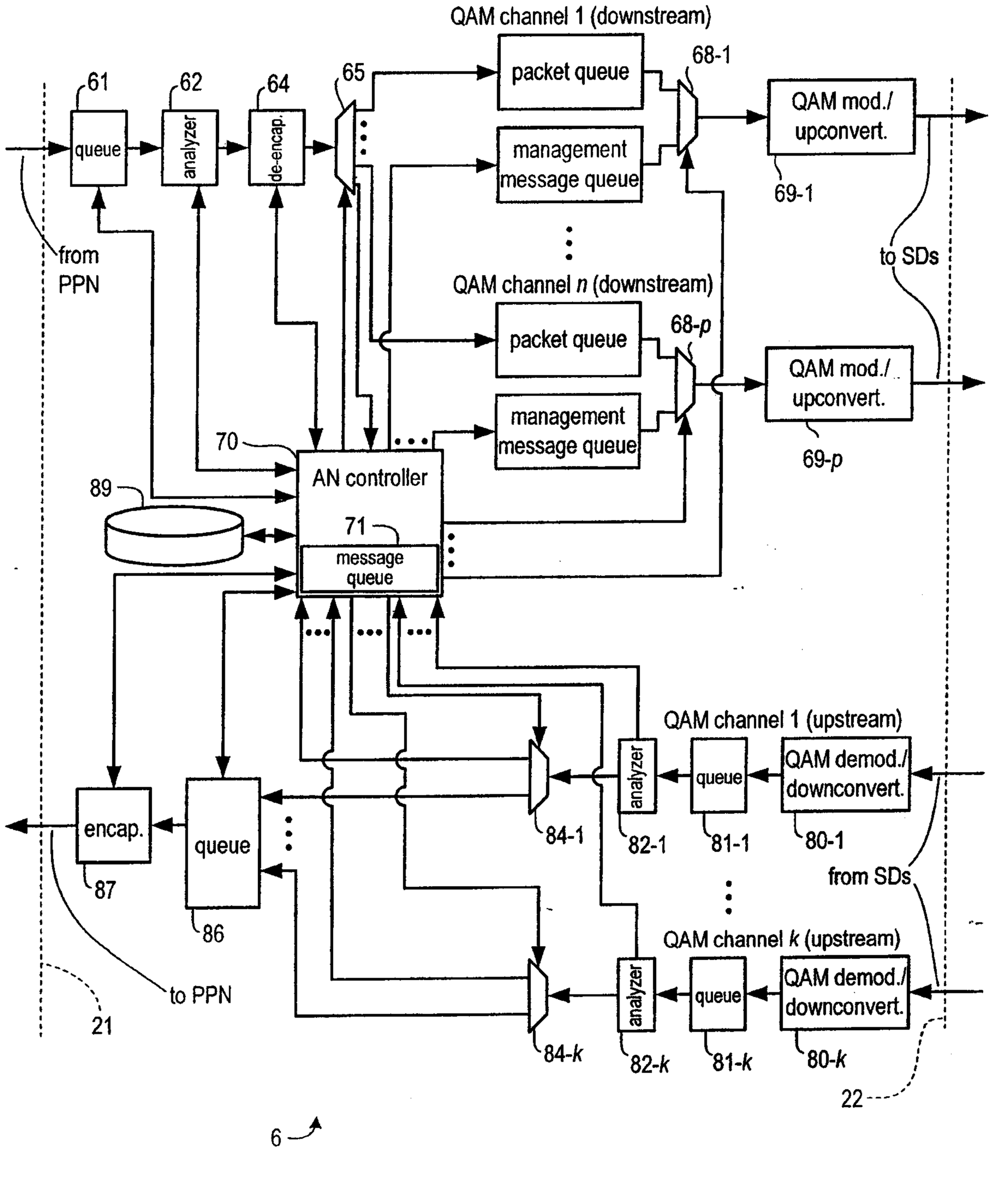

QAM channel 1 (downstream)
packet queue
68-1
61
62
64
65
queue
analyzer
de-encap.
management message queue
QAM mod./ upconvert.
69-1
from PPN
to SDs
QAM channel n (downstream)
packet queue
68-p
management message queue
QAM mod./ upconvert.
69-p
70
AN controller
89
71
message queue
QAM channel 1 (upstream)
encap.
queue
87
86
84-1
analyzer
queue
QAM demod./ downconvert.
82-1
81-1
80-1
from SDs
QAM channel k (upstream)
to PPN
21
84-k
82-k
81-k
80-k
22
6

FIG. 5

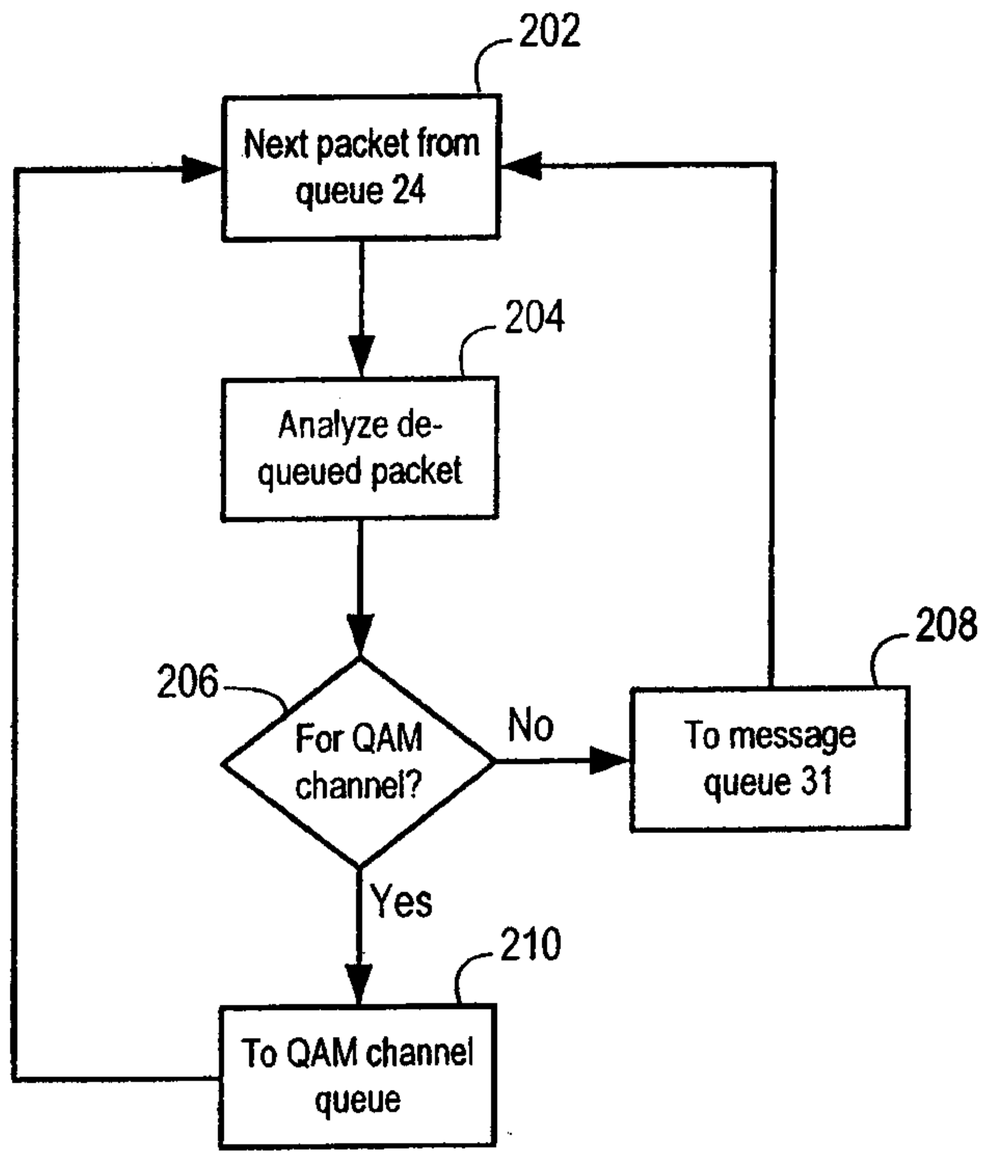

202
Next packet from queue 24
204
Analyze de-queued packet
206
For QAM channel?
No
208
To message queue 31
Yes
210
To QAM channel queue

FIG. 6

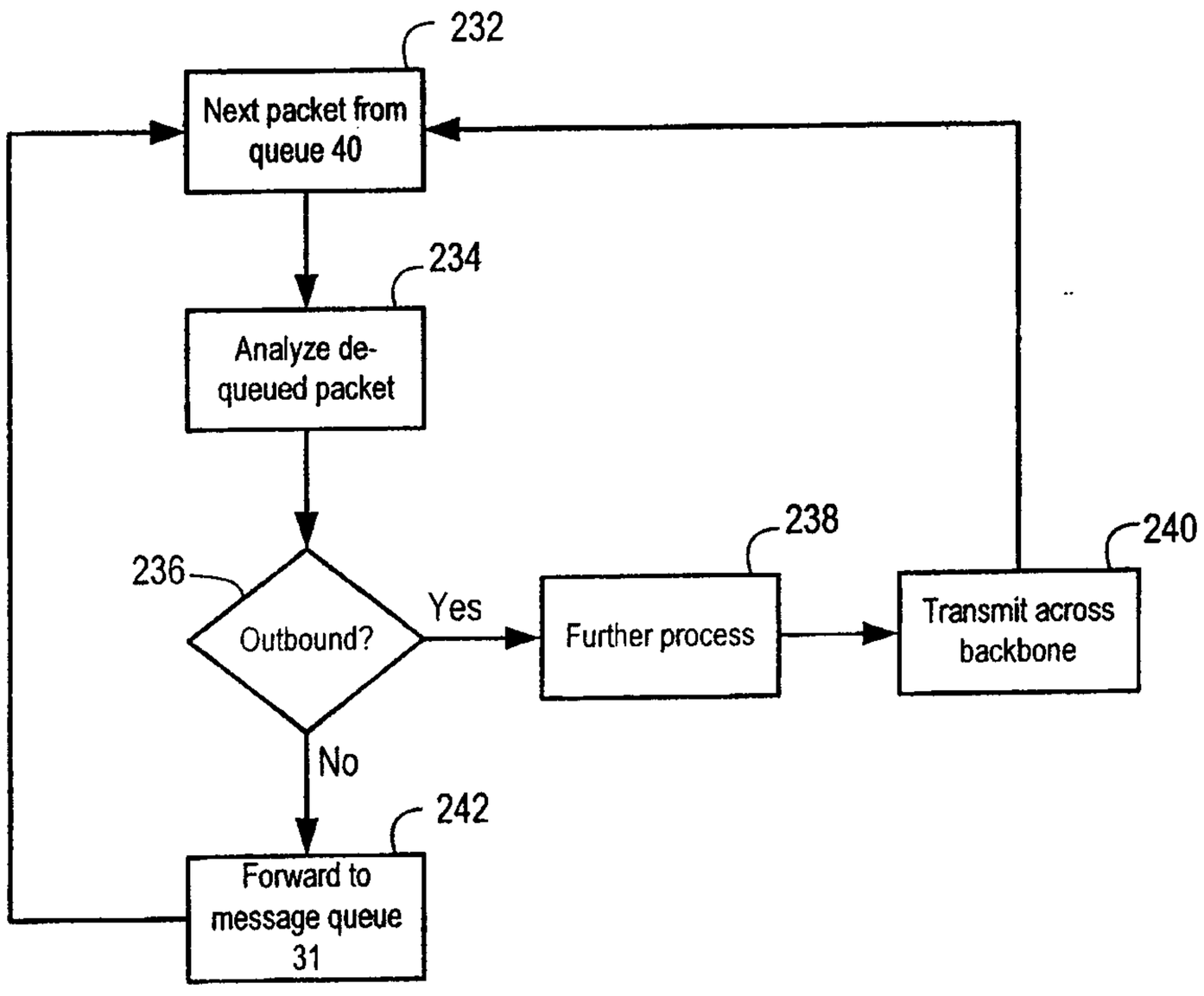

232
Next packet from queue 40
234
Analyze de-queued packet
236
Outbound?
Yes
238
Further process
240
Transmit across backbone
No
242
Forward to message queue 31

FIG. 7

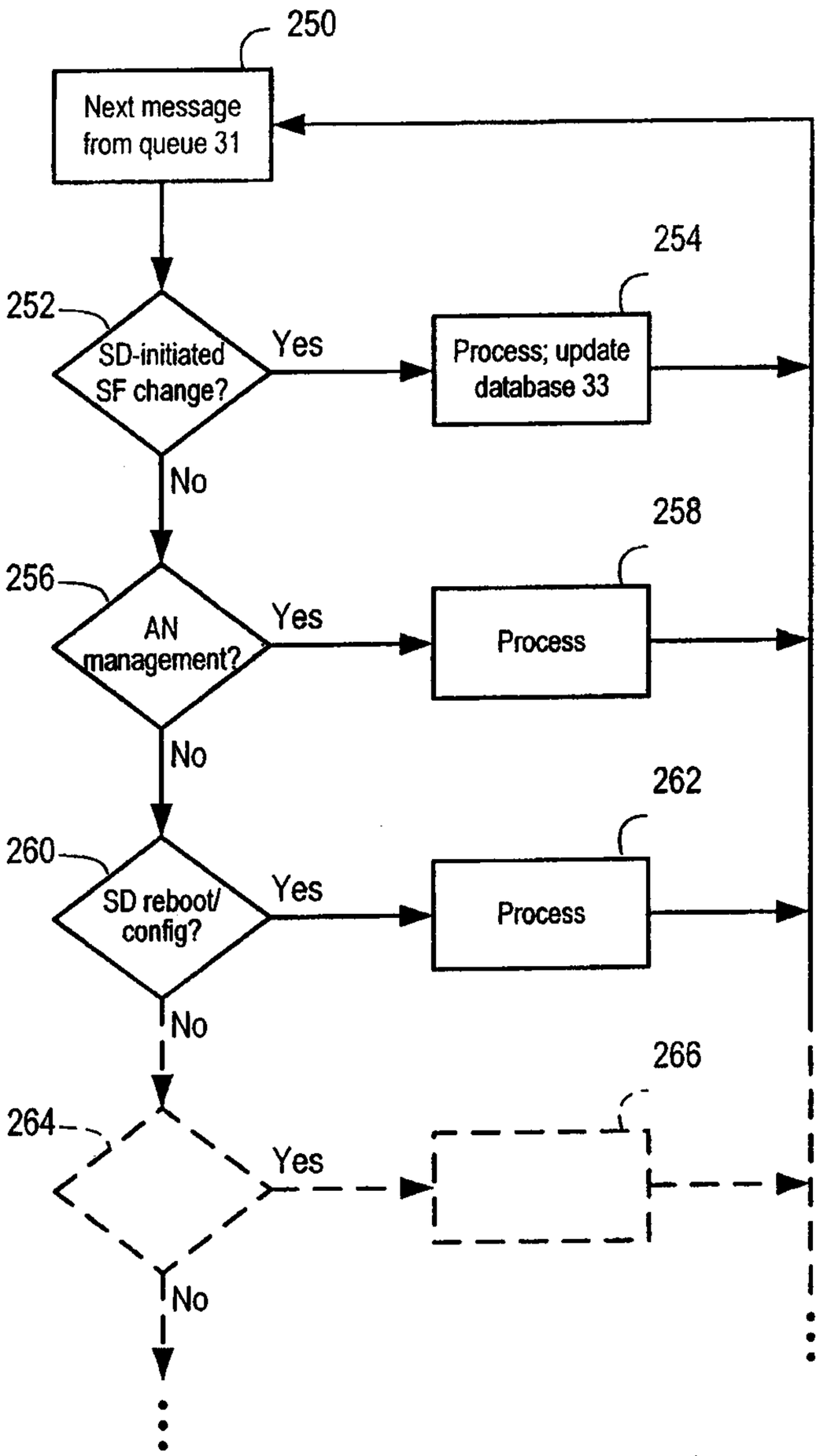

250
Next message from queue 31
252
SD-initiated SF change?
Yes
254
Process; update database 33
No
256
AN management?
Yes
258
Process
No
260
SD reboot/ config?
Yes
262
Process
No
264
Yes
266
No

FIG. 8

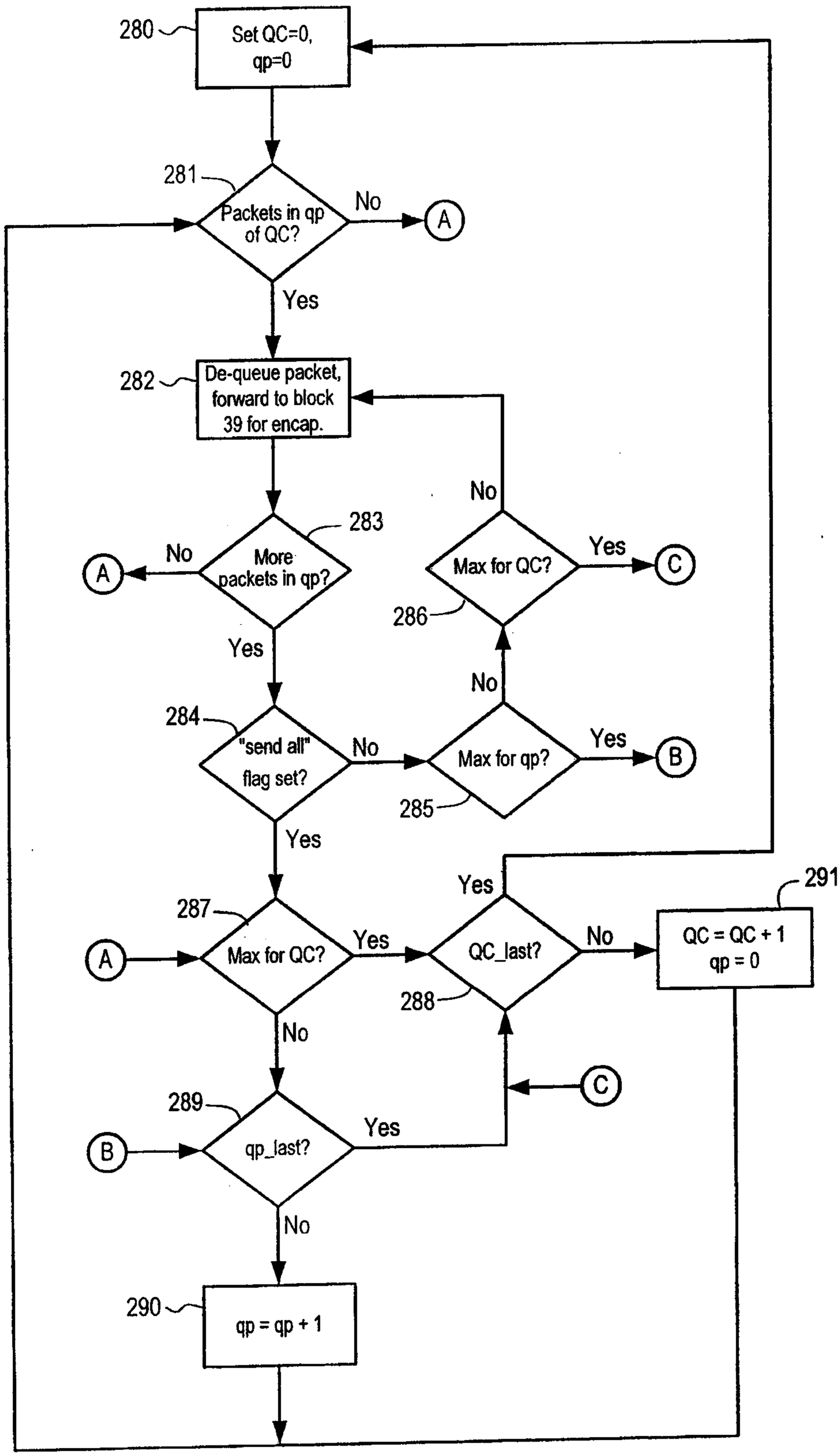

280
Set QC=0,
qp=0
281
Packets in qp
of QC?
No
A
Yes
282
De-queue packet,
forward to block
39 for encap.
283
More
packets in qp?
No
A
Yes
284
"send all"
flag set?
No
Max for qp?
Yes
B
285
No
286
Max for QC?
Yes
C
No
287
Max for QC?
Yes
A
No
288
QC_last?
No
Yes
291
QC = QC + 1
qp = 0
289
qp_last?
Yes
B
C
No
290
qp = qp + 1

FIG. 9

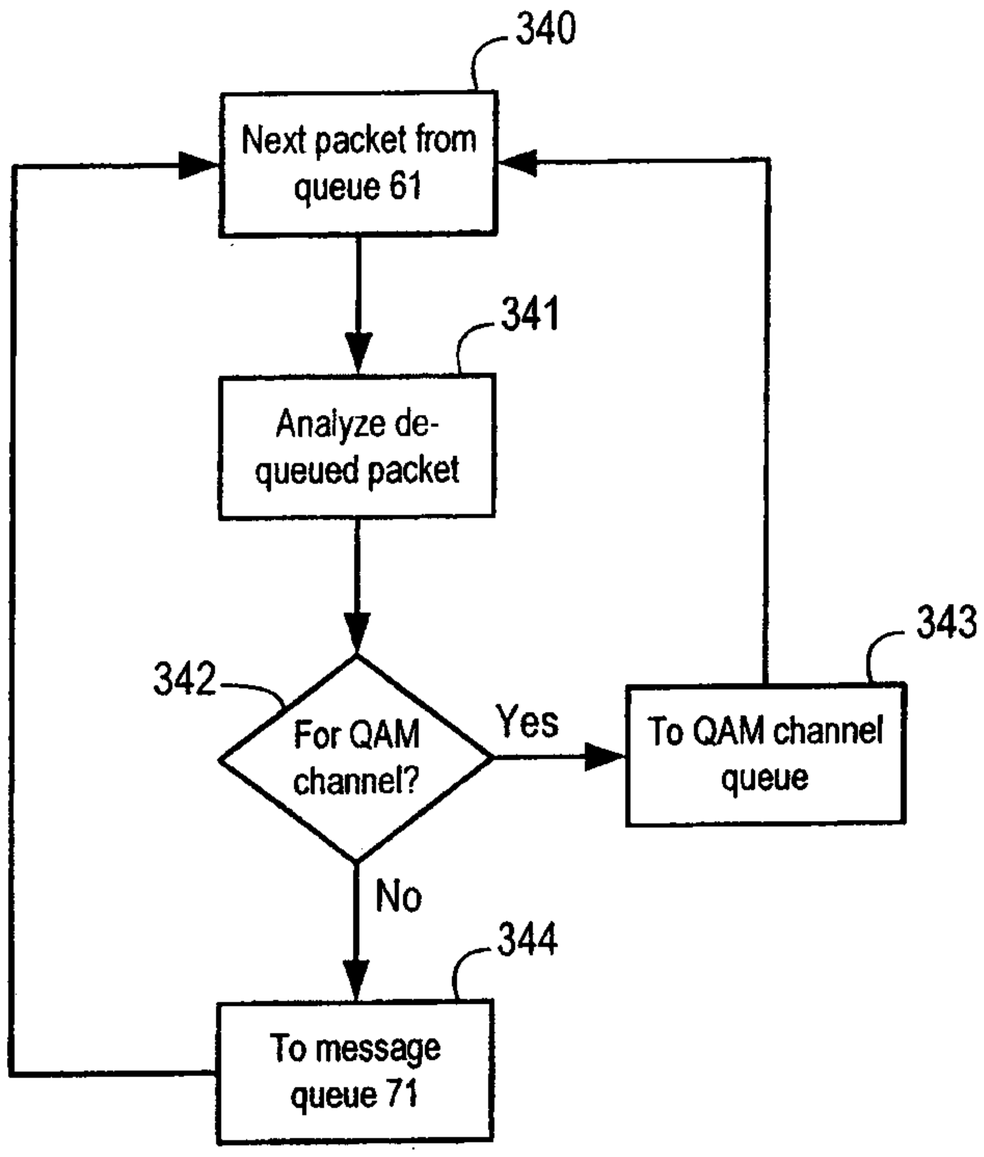

340
Next packet from queue 61
341
Analyze de-queued packet
342
For QAM channel?
Yes
343
To QAM channel queue
No
344
To message queue 71

FIG. 10

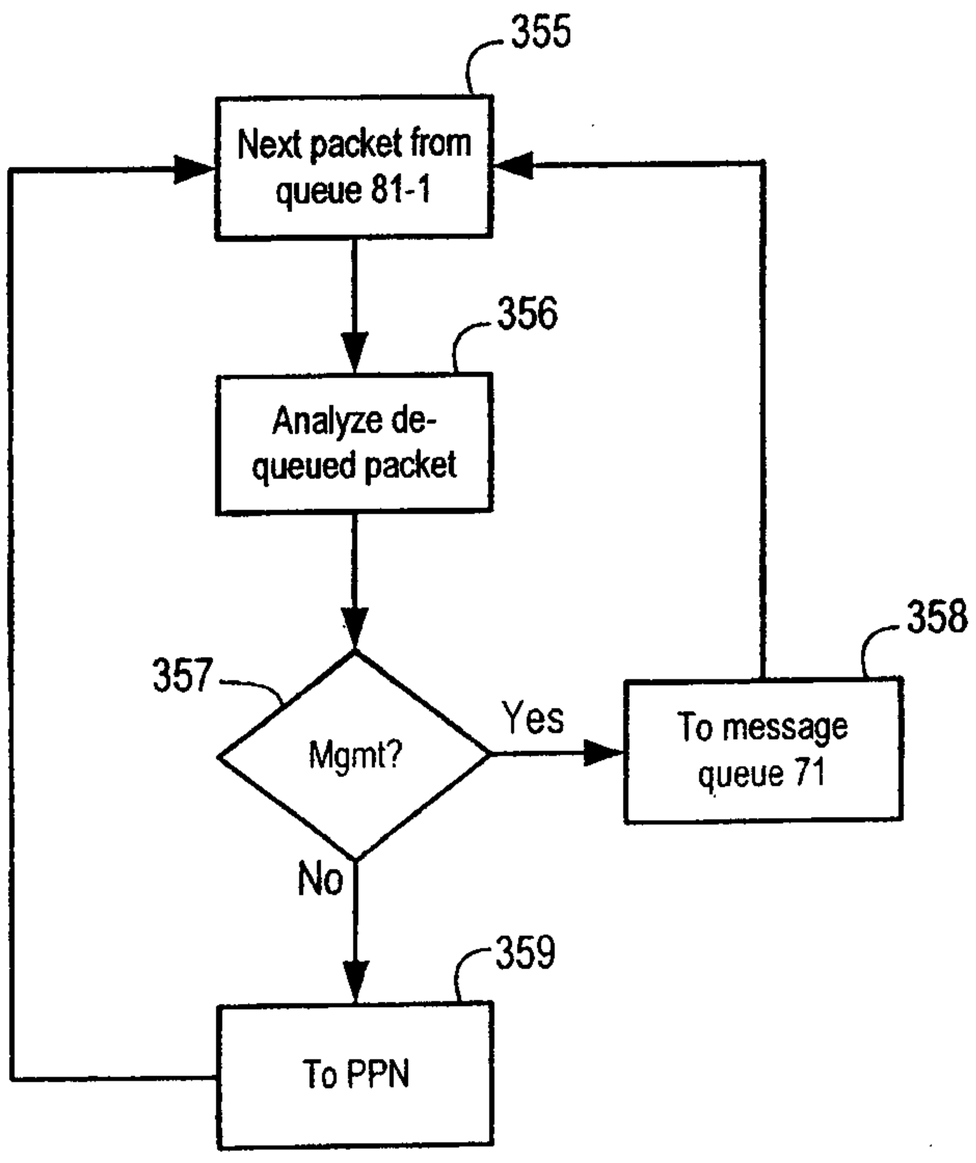

355
Next packet from queue 81-1
356
Analyze de-queued packet
357
Mgmt?
Yes
358
To message queue 71
No
359
To PPN

FIG. 11

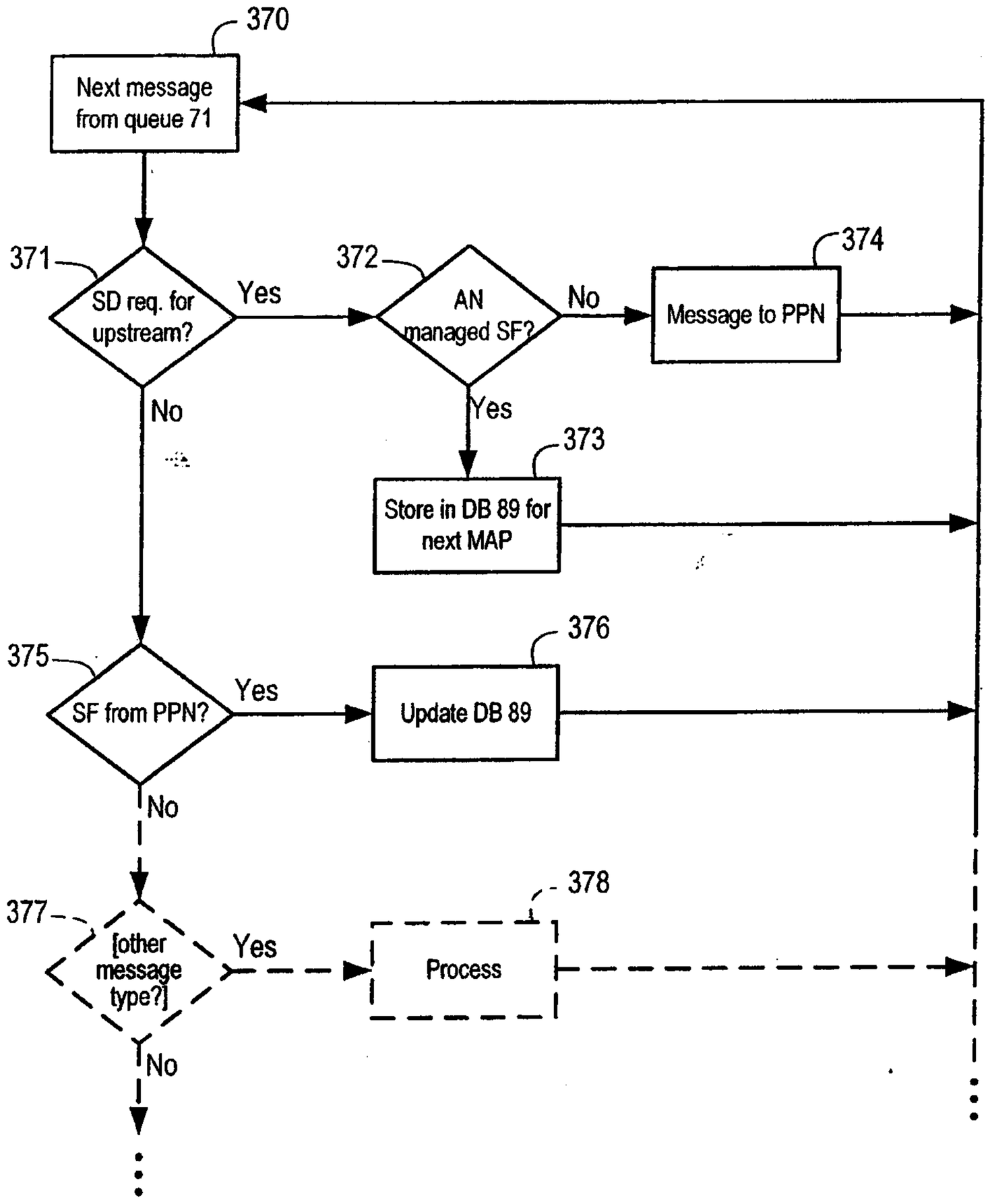

370
Next message from queue 71
371
SD req. for upstream?
Yes
372
AN managed SF?
No
374
Message to PPN
Yes
373
Store in DB 89 for next MAP
No
375
SF from PPN?
Yes
376
Update DB 89
No
377
[other message type?]
Yes
378
Process
No

FIG. 12

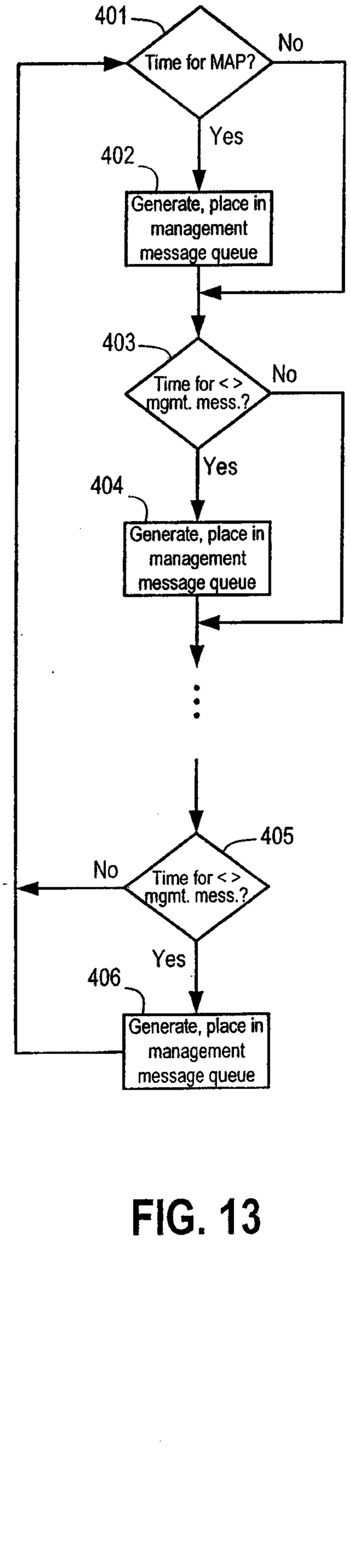

401
Time for MAP?
No
Yes
402
Generate, place in management message queue
403
Time for < > mgmt. mess.?
No
Yes
404
Generate, place in management message queue
405
Time for < > mgmt. mess.?
No
Yes
406
Generate, place in management message queue

FIG. 13

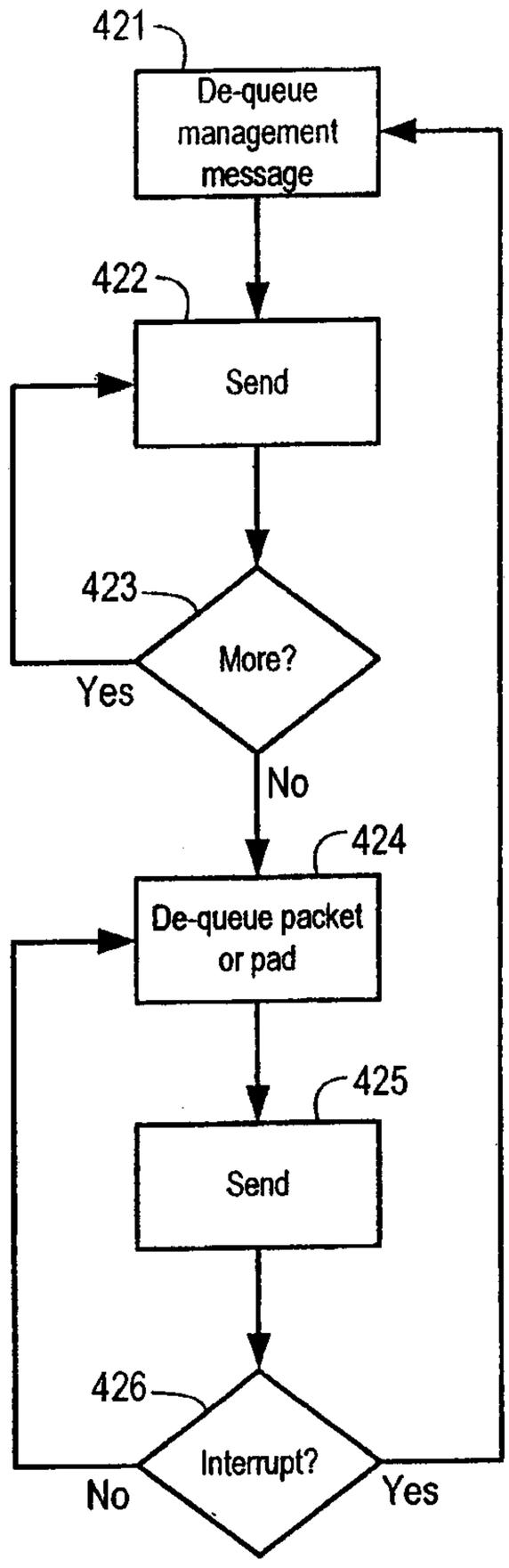

421
De-queue management message
422
Send
423
More?
Yes
No
424
De-queue packet or pad
425
Send
426
Interrupt?
No
Yes

FIG. 14

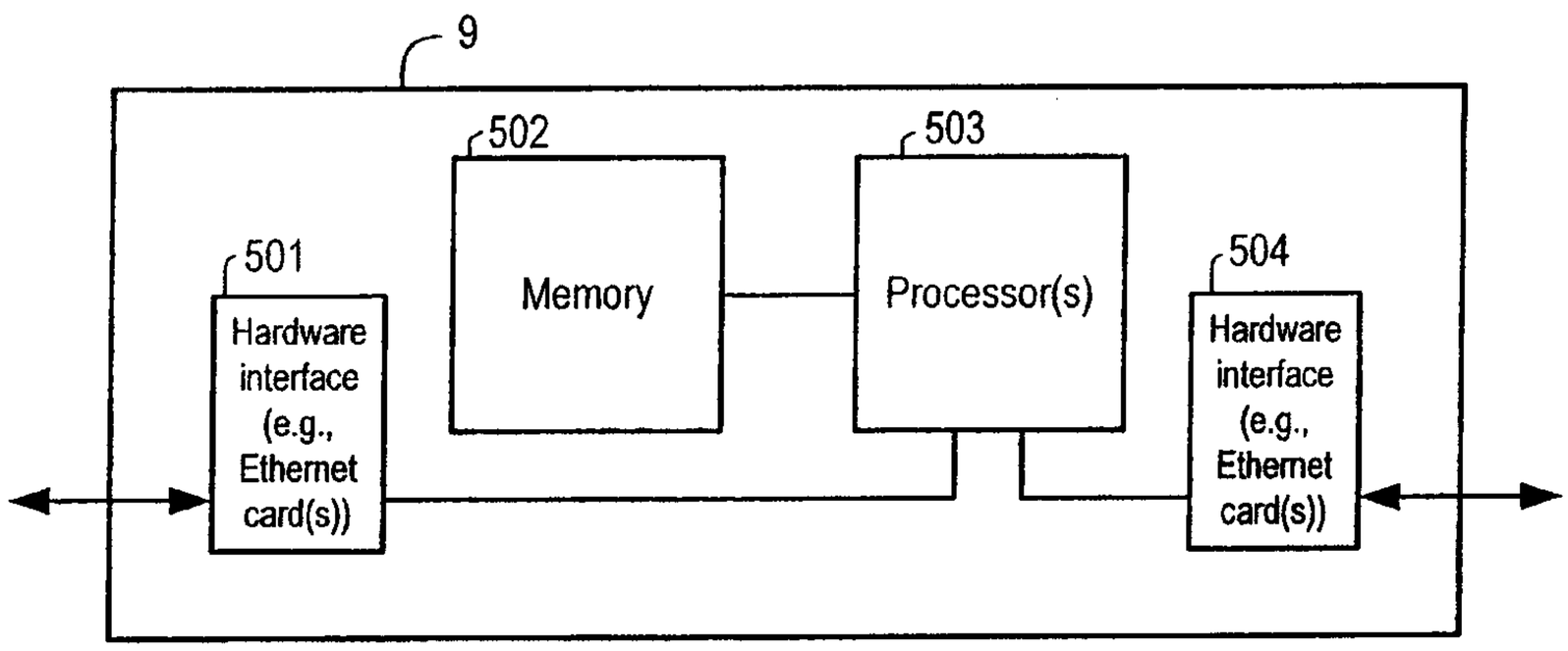

9
502
503
501
Memory
Processor(s)
504
Hardware interface (e.g., Ethernet card(s))
Hardware interface (e.g., Ethernet card(s))

FIG. 15

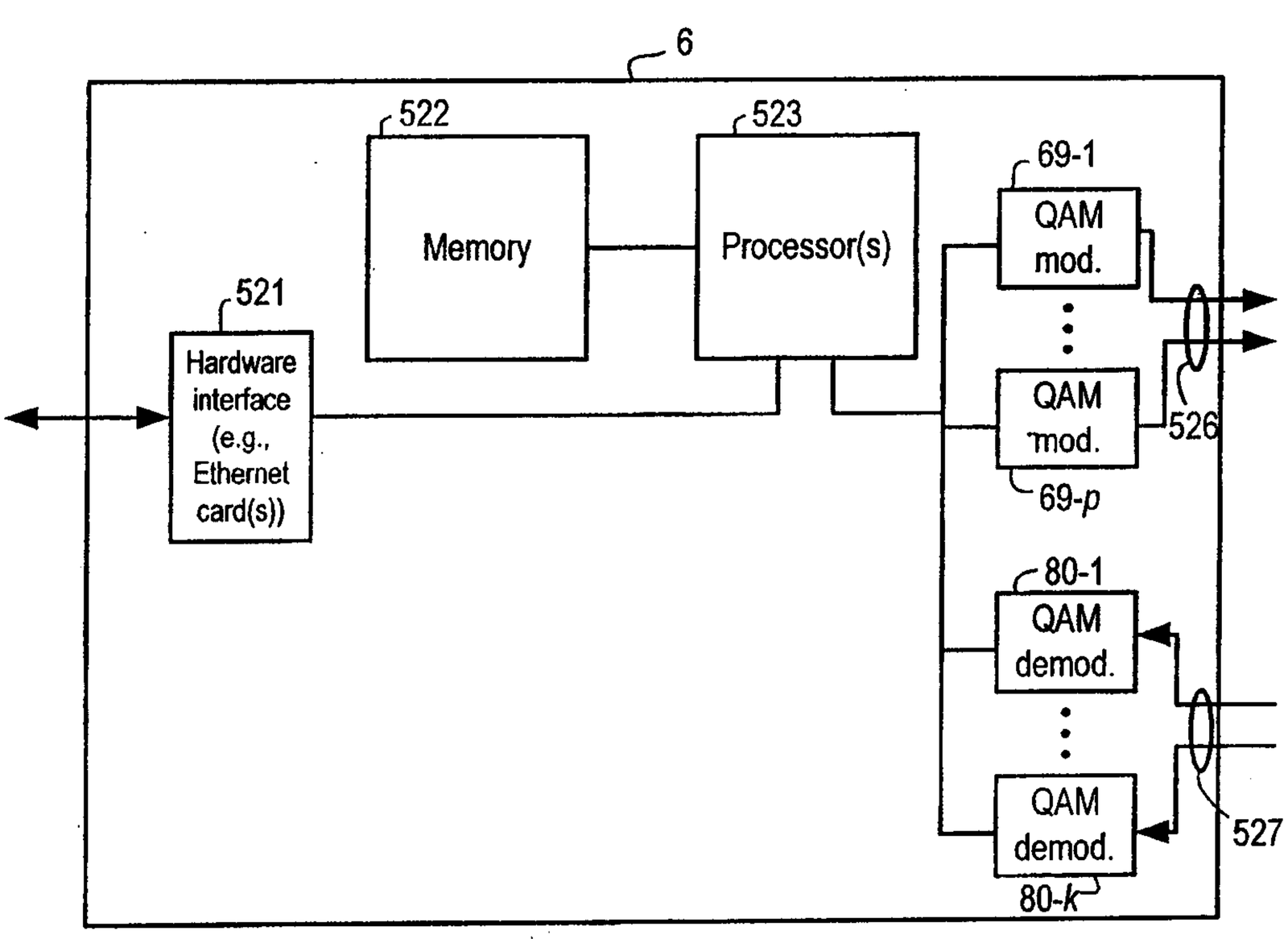

6
522
523
69-1
Memory
Processor(s)
QAM mod.
521
Hardware interface (e.g., Ethernet card(s))
QAM mod.
526
69-p
80-1
QAM demod.
QAM demod.
527
80-k

FIG. 16

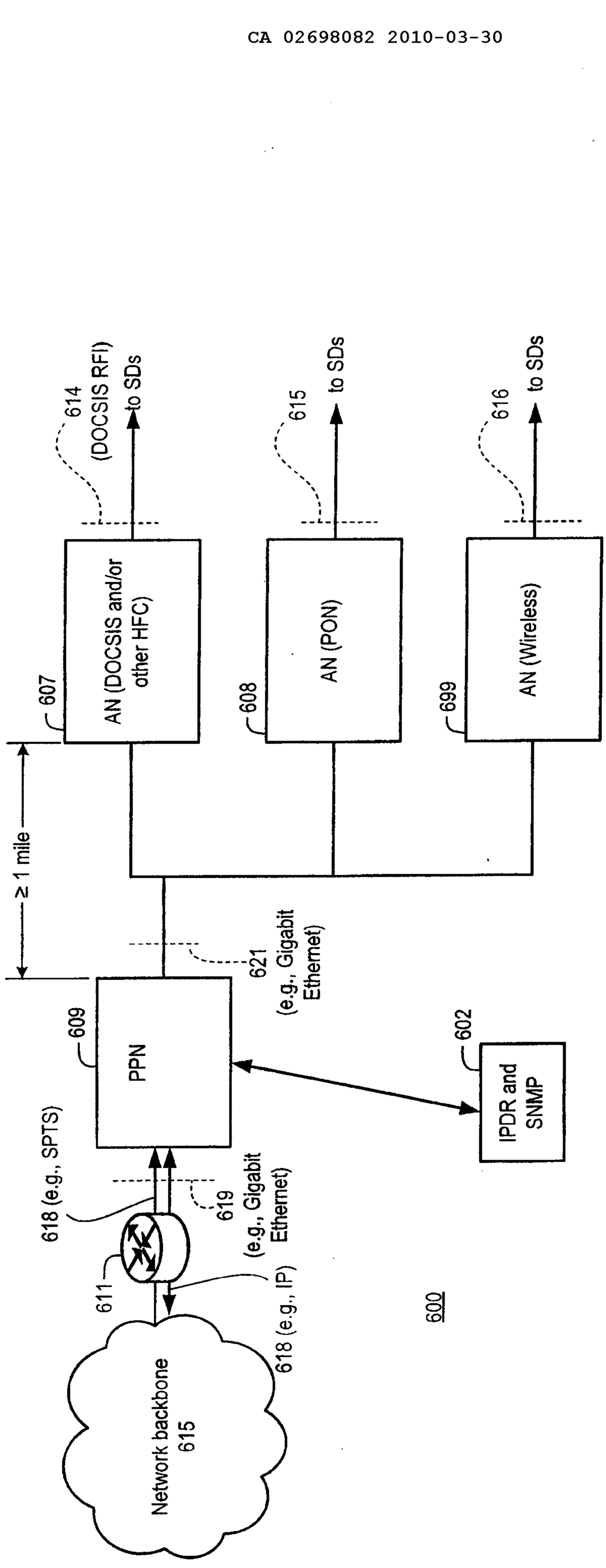

614
(DOCSIS RFI)
to SDs
615
to SDs
616
to SDs
607
AN (DOCSIS and/or other HFC)
608
AN (PON)
699
AN (Wireless)
≥ 1 mile
621
(e.g., Gigabit Ethernet)
609
PPN
602
IPDR and SNMP
618 (e.g., SPTS)
619
(e.g., Gigabit Ethernet)
611
618 (e.g., IP)
Network backbone
615
600

FIG. 17

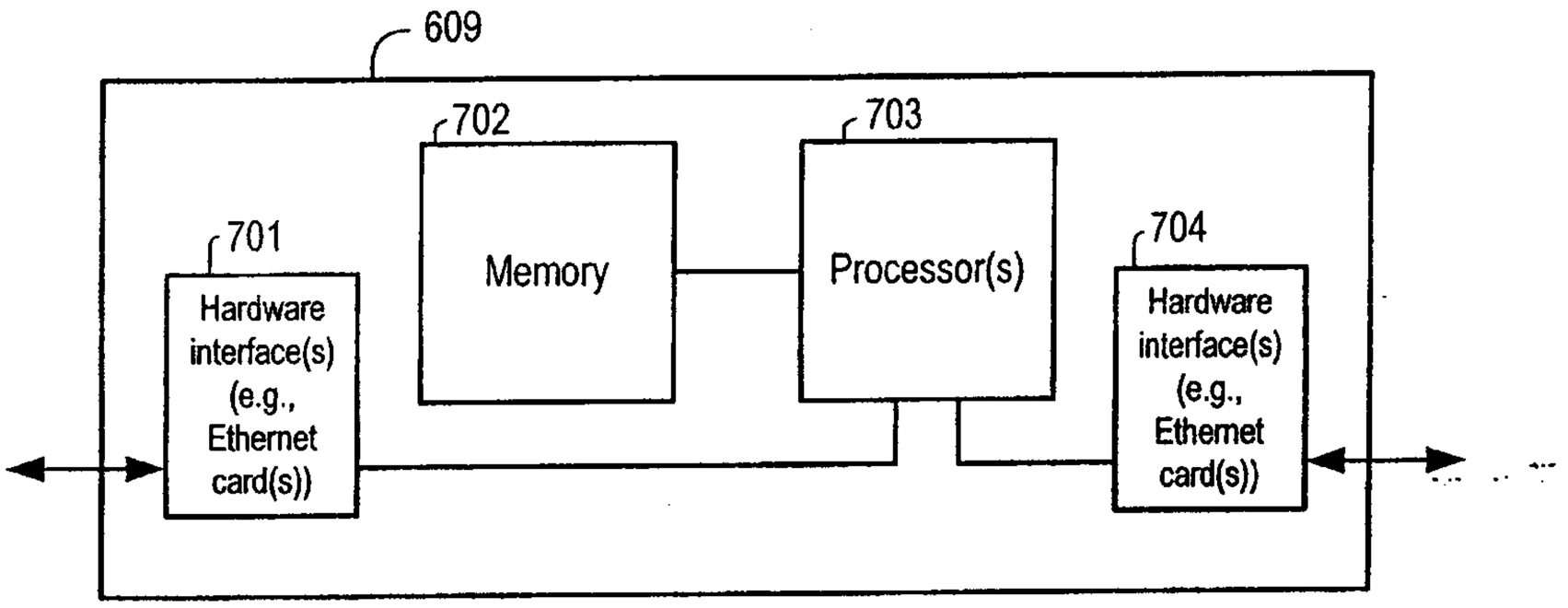

609
702
703
701
Memory
Processor(s)
704
Hardware interface(s) (e.g., Ethernet card(s))
Hardware interface(s) (e.g., Ethernet card(s))

FIG. 18

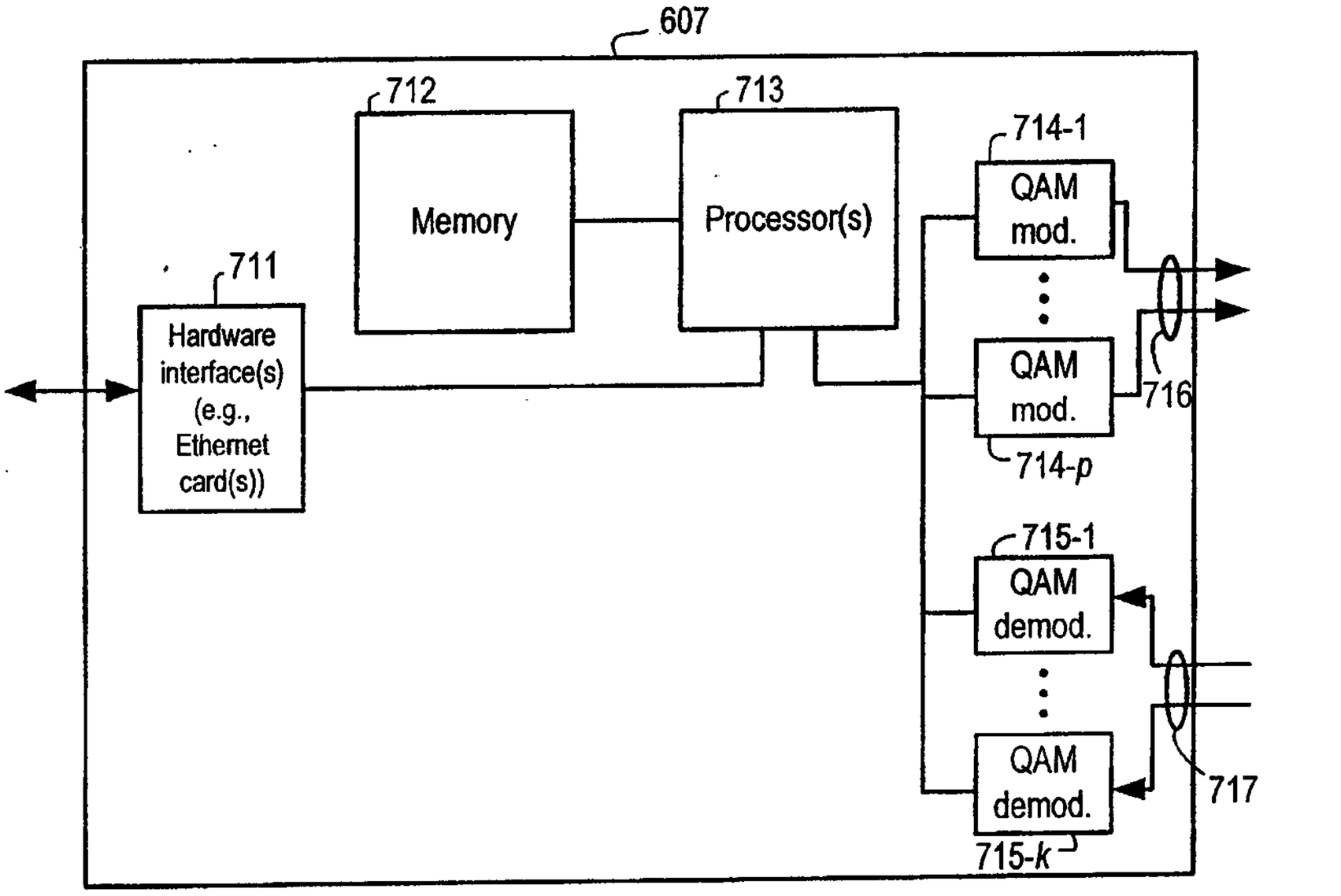

607
712
713
714-1
Memory
Processor(s)
QAM mod.
711
Hardware interface(s) (e.g., Ethernet card(s))
QAM mod.
716
714-p
715-1
QAM demod.
QAM demod.
717
715-k

FIG. 19

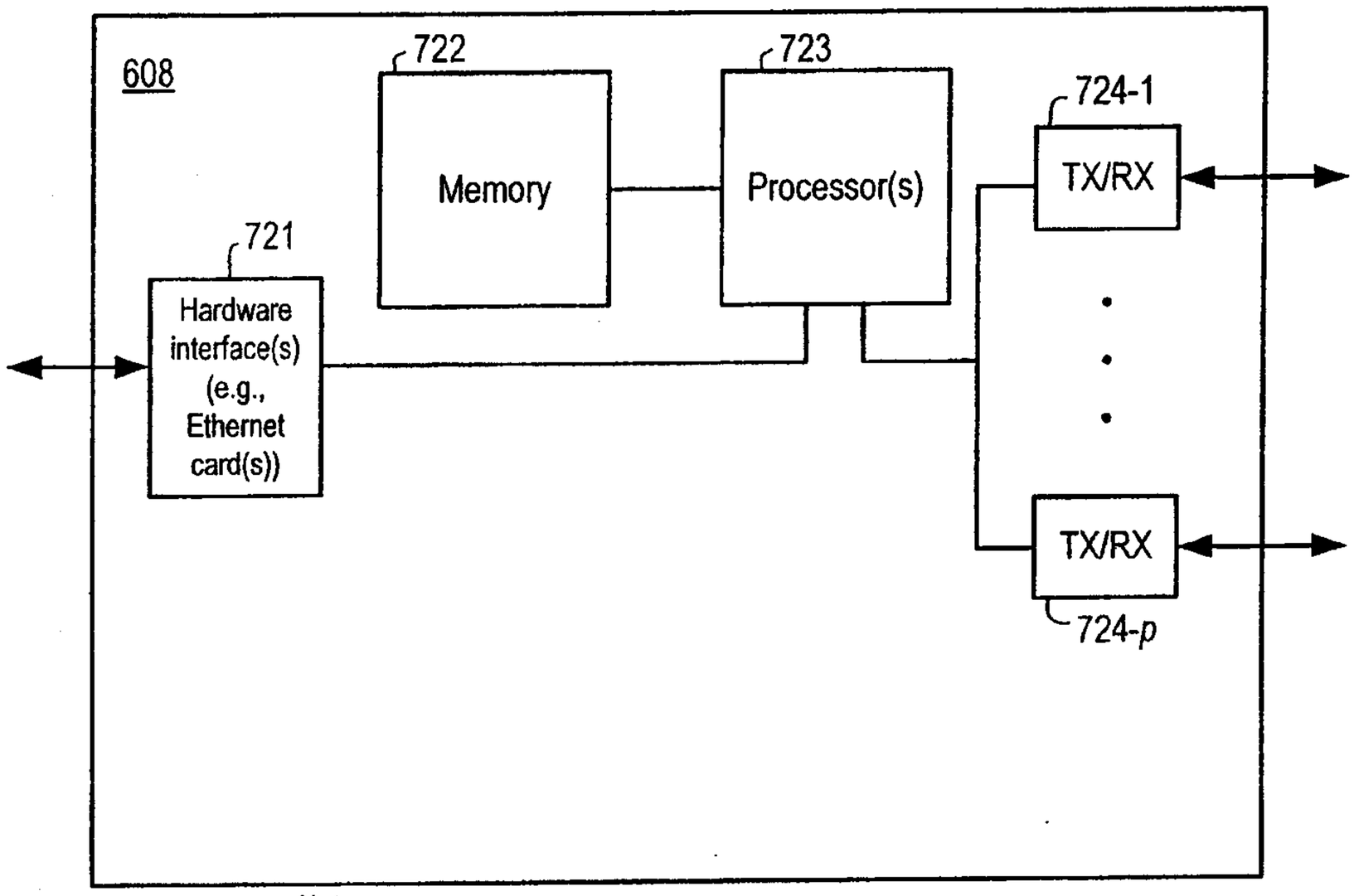

608
722
Memory
723
Processor(s)
724-1
TX/RX
721
Hardware interface(s) (e.g., Ethernet card(s))
TX/RX
724-p

FIG. 20

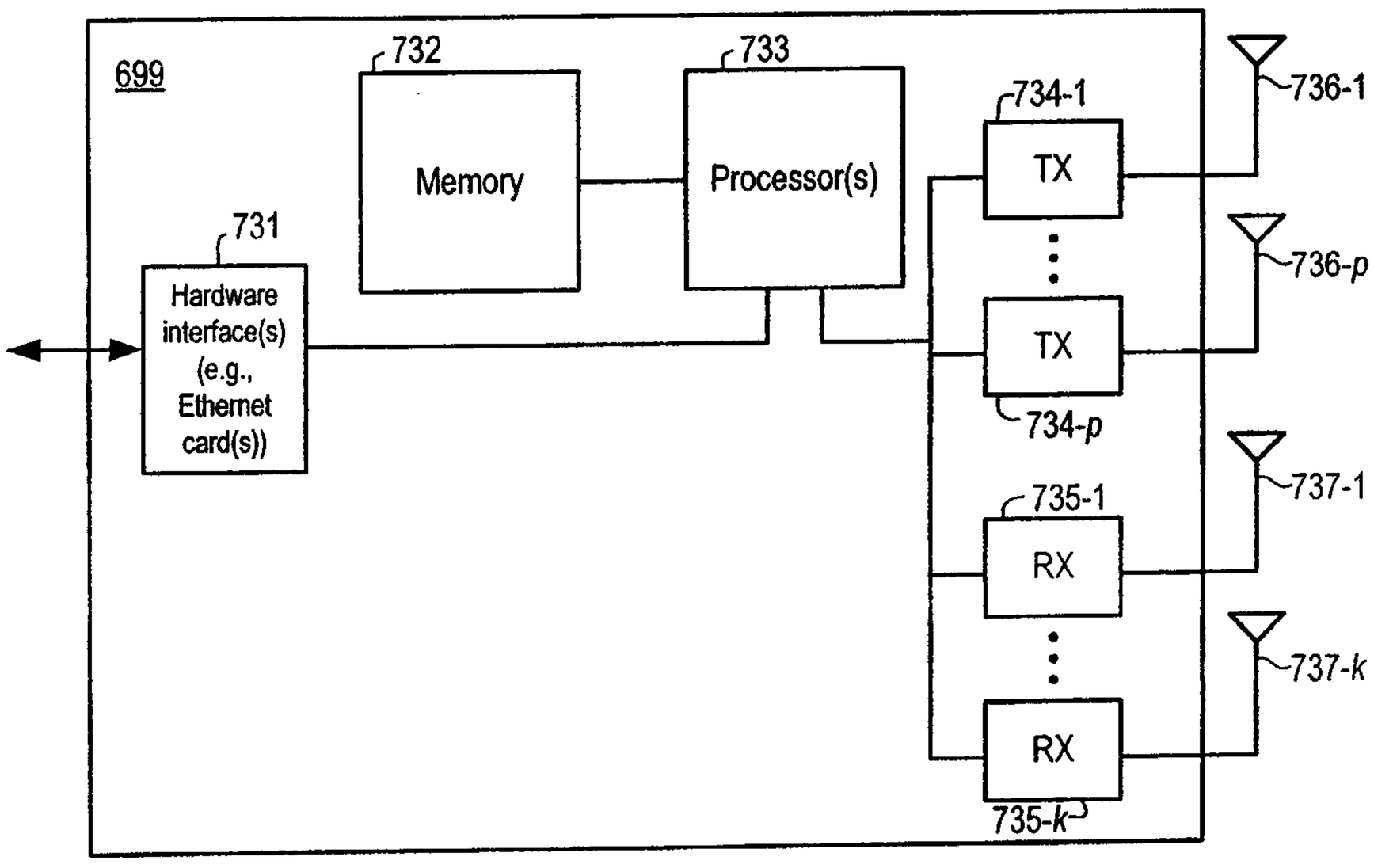

699
732
Memory
733
Processor(s)
734-1
TX
736-1
731
Hardware interface(s) (e.g., Ethernet card(s))
TX
736-p
734-p
735-1
RX
737-1
737-k
RX
735-k

FIG. 21

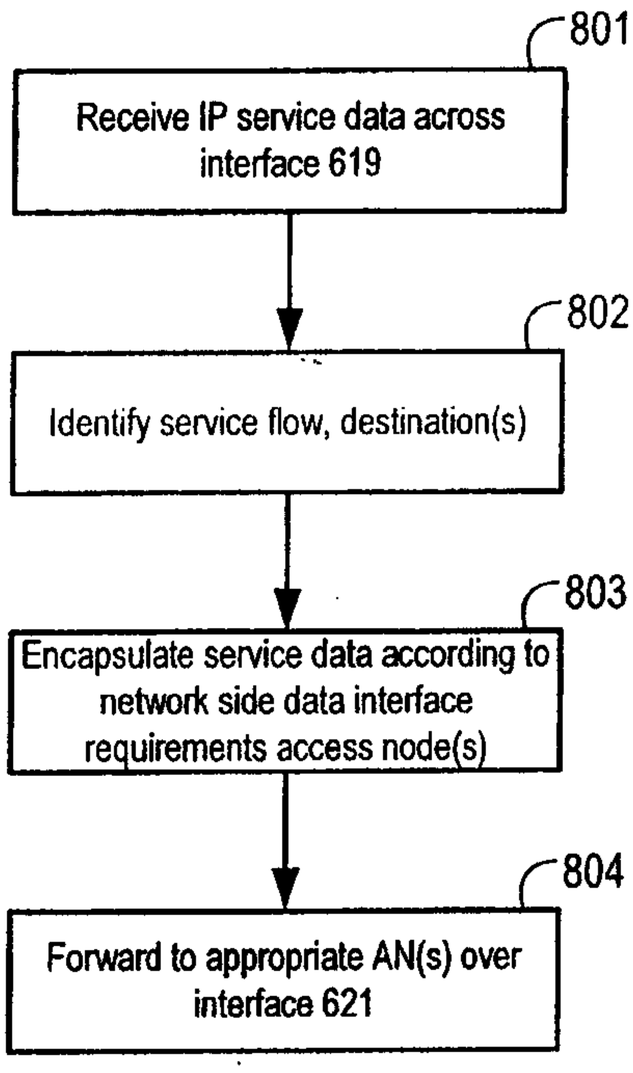

801
Receive IP service data across interface 619
802
Identify service flow, destination(s)
803
Encapsulate service data according to network side data interface requirements access node(s)
804
Forward to appropriate AN(s) over interface 621

FIG. 22

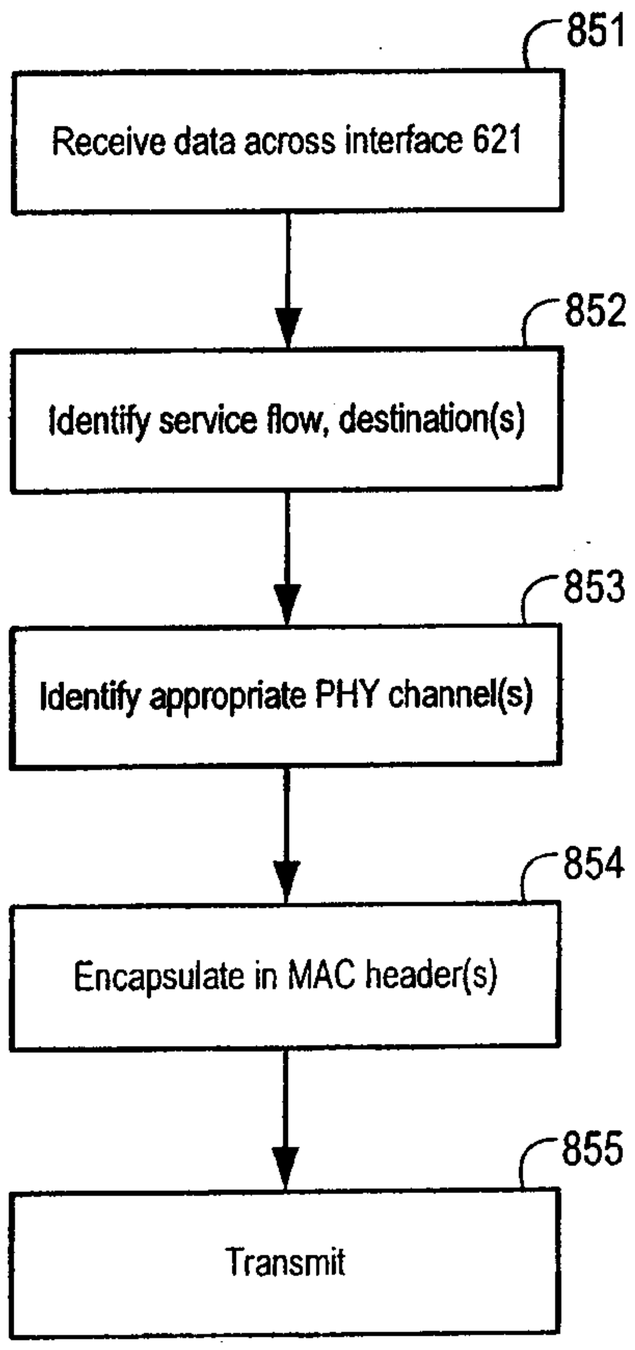

851
Receive data across interface 621
852
Identify service flow, destination(s)
853
Identify appropriate PHY channel(s)
854
Encapsulate in MAC header(s)
855
Transmit

FIG. 23

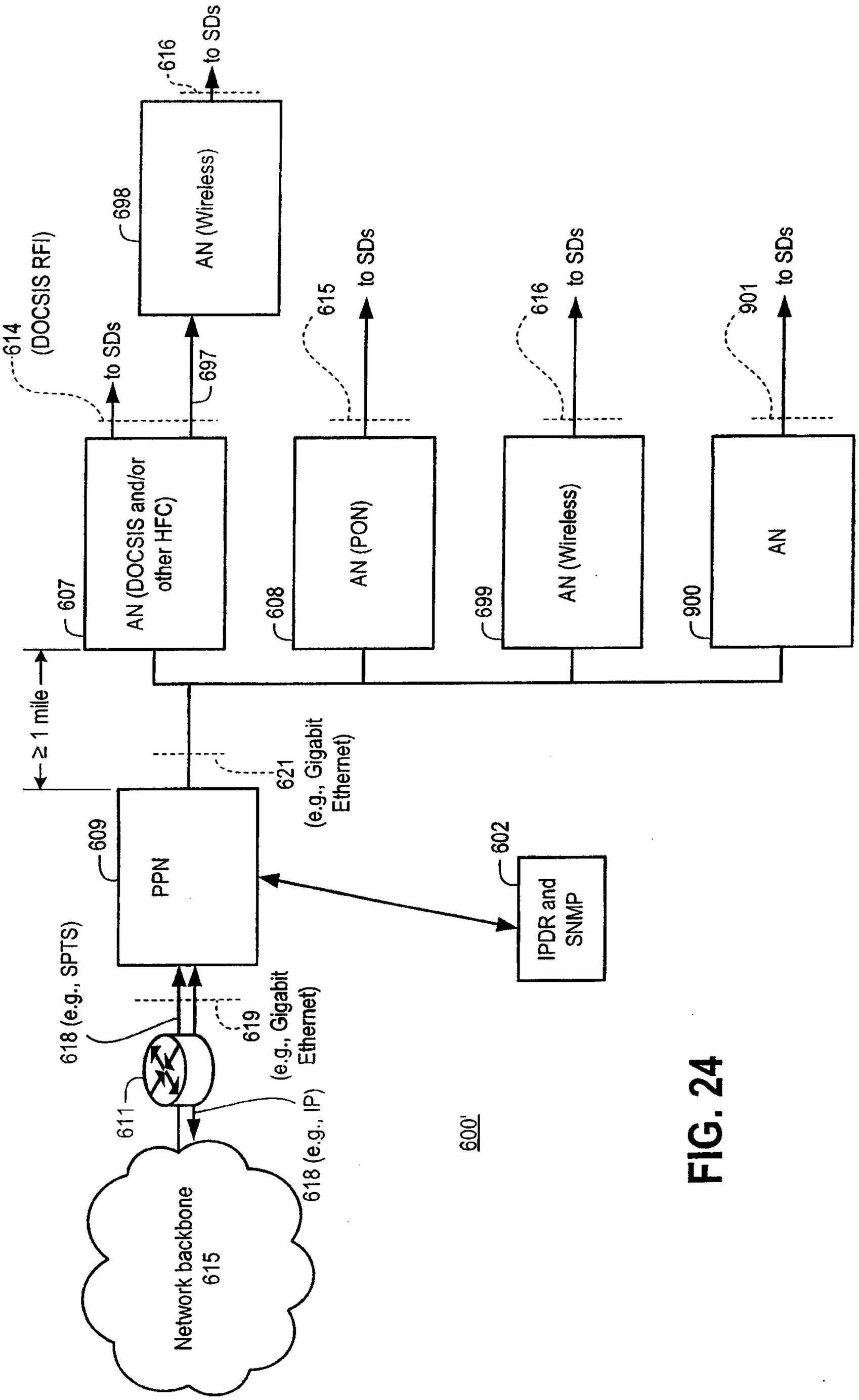
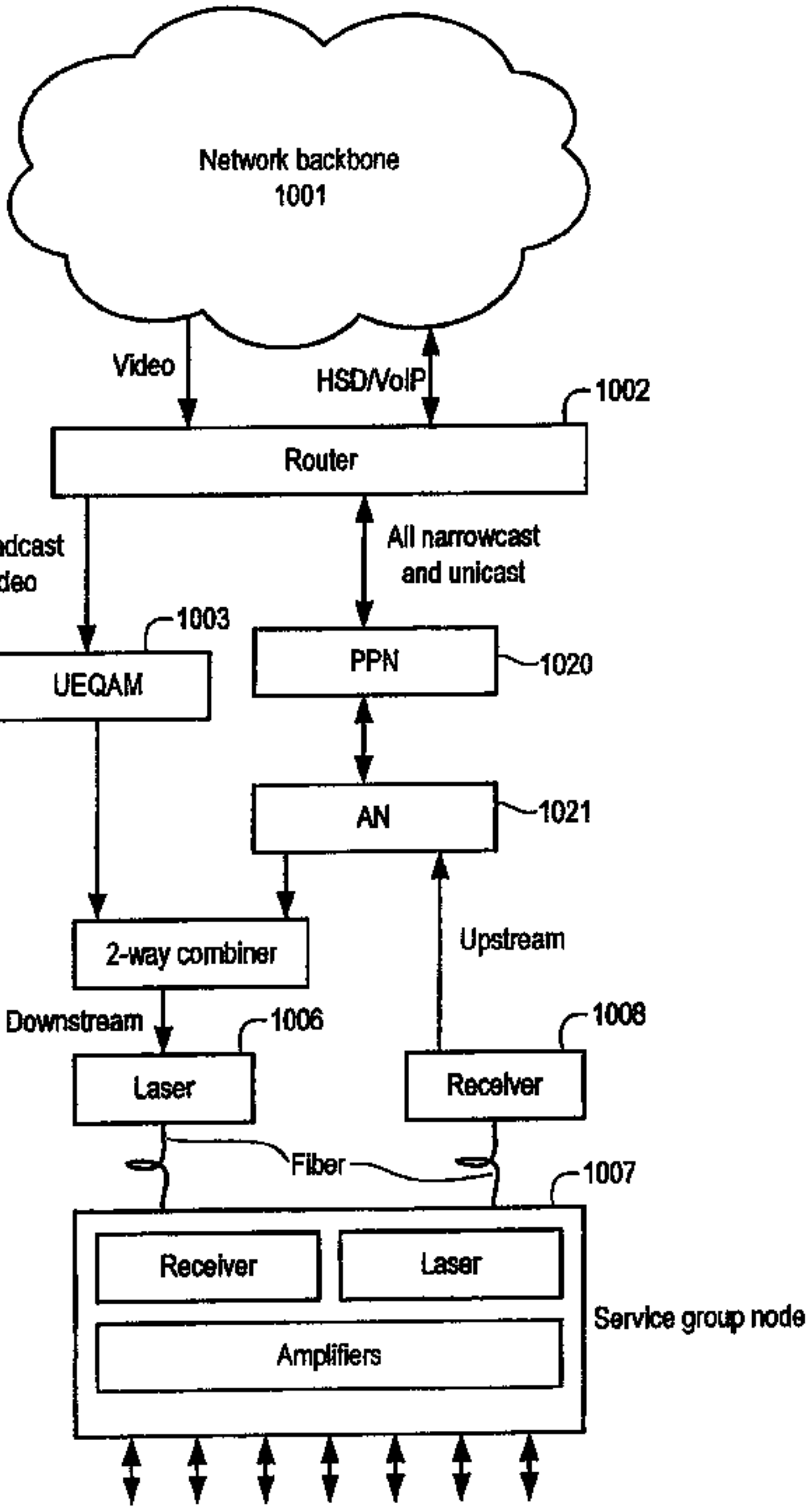

600'
Network backbone
615
618 (e.g., IP)
611
618 (e.g., SPTS)
619
(e.g., Gigabit Ethernet)
PPN
609
621
(e.g., Gigabit Ethernet)
≥ 1 mile
IPDR and SNMP
602
AN (DOCSIS and/or other HFC)
607
AN (PON)
608
AN (Wireless)
699
AN
900
AN (Wireless)
698
614
(DOCSIS RFI)
697
615
616
901
to SDs

FIG. 24


Network backbone
1001
Video
HSD/VoIP
1002
Router
Broadcast
video
All narrowcast
and unicast
1003
UEQAM
PPN
1020
AN
1021
2-way combiner
Upstream
Downstream
1006
1008
Laser
Receiver
Fiber
1007
Receiver
Laser
Amplifiers
Service group node
distribution to subscribers