US009270616B1

(12) **United States Patent**
Cotter

(10) **Patent No.: US 9,270,616 B1**
(45) **Date of Patent: Feb. 23, 2016**

(54) **LOW-LATENCY QUALITY OF SERVICE**

(71) Applicant: **ARRIS Group, Inc.**, Suwanee, GA (US)

(72) Inventor: **Anthony Cotter**, Ballincollig (IE)

(73) Assignee: **ARRIS Enterprises, Inc.**, Suwanee, GA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 143 days.

(21) Appl. No.: **13/772,486**

(22) Filed: **Feb. 21, 2013**

(51) **Int. Cl.**
H04L 12/911 (2013.01)

(52) **U.S. Cl.**
CPC ***H04L 47/821*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,558,247	B2 *	7/2009	Lee et al.	370/351
7,688,853	B2 *	3/2010	Santiago et al.	370/468
7,911,956	B2 *	3/2011	Schmidt	370/235
7,948,883	B1 *	5/2011	Croft et al.	370/230
8,045,561	B1 *	10/2011	Varma et al.	370/395.4
2001/0040697	A1 *	11/2001	Wu et al.	358/1.15
2002/0186661	A1 *	12/2002	Santiago et al.	370/252
2003/0161303	A1 *	8/2003	Mehrvar et al.	370/386
2005/0100022	A1 *	5/2005	Ramprashad	370/395.42
2005/0249114	A1 *	11/2005	Mangin et al.	370/229
2006/0087969	A1 *	4/2006	Santiago et al.	370/229
2007/0058677	A1 *	3/2007	Kwon et al.	370/468
2008/0056125	A1 *	3/2008	Kneckt et al.	370/229
2009/0028151	A1 *	1/2009	Schmidt	370/392
2009/0323525	A1 *	12/2009	Chen et al.	370/230
2011/0194411	A1 *	8/2011	Croft et al.	370/235
2013/0003545	A1 *	1/2013	Li	370/230.1
2013/0066674	A1 *	3/2013	Vasters	705/7.29
2013/0077486	A1 *	3/2013	Keith	370/230.1
2013/0128735	A1 *	5/2013	Li et al.	370/230
2013/0182573	A1 *	7/2013	Soppera et al.	370/235
2015/0009826	A1 *	1/2015	Ma et al.	370/235

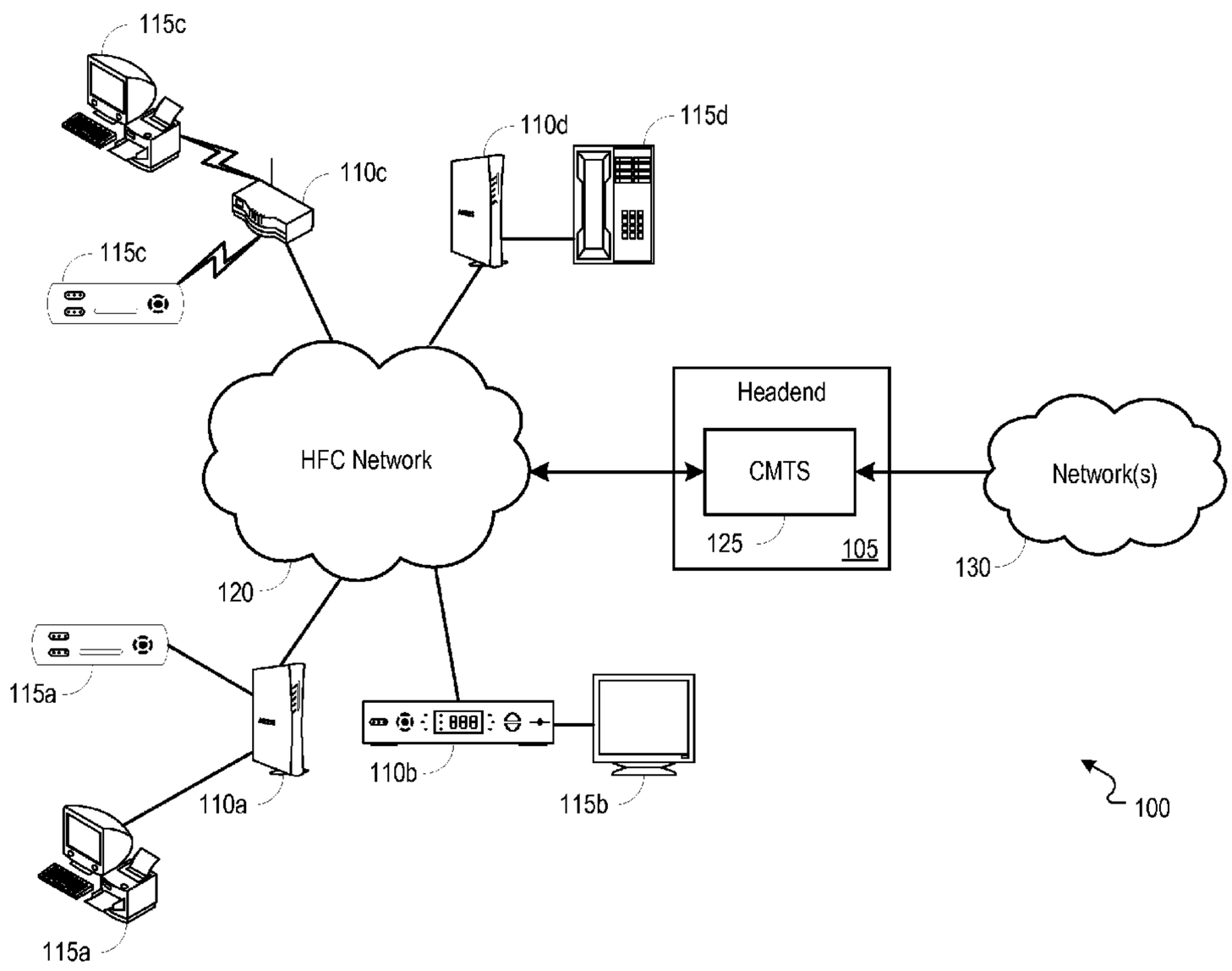
* cited by examiner

Primary Examiner — Ayaz Sheikh
Assistant Examiner — Peter Chau
(74) *Attorney, Agent, or Firm* — Bart A. Perkins

(57) **ABSTRACT**

Systems, methods and computer readable media can be operable to apportion a transmission medium's available resources between multiple subscribers and to allocate a subscriber's apportioned resources between a plurality of subflows. The available resources of a transmission medium can be apportioned between subscribers receiving data through the transmission medium and the resources apportioned to a subscriber can be allocated to different data flows by giving priority to low-latency data packets over normal-latency data packets.

16 Claims, 5 Drawing Sheets



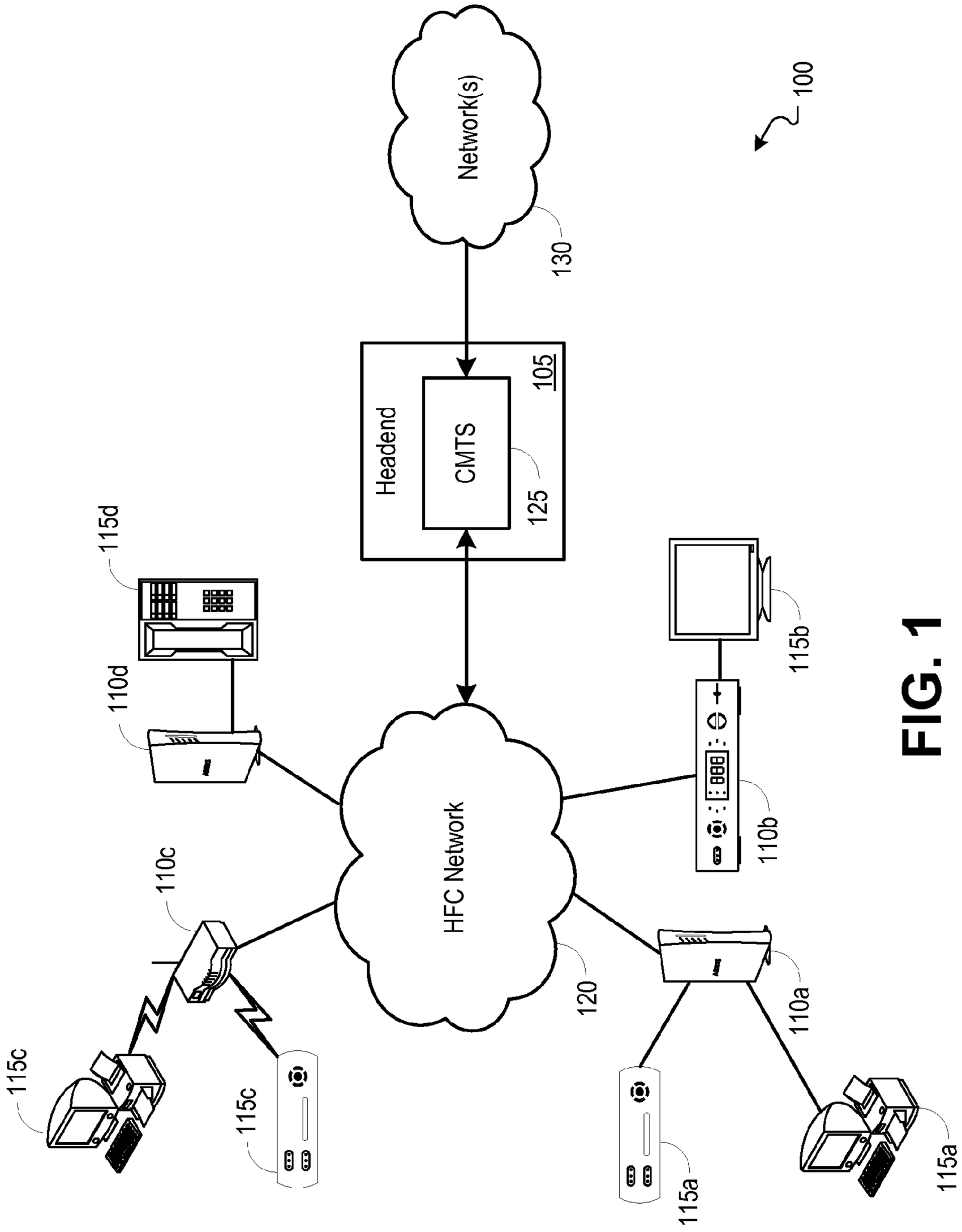

100
Network(s)
130
Headend
CMTS
105
125
115d
110d
110c
HFC Network
120
115b
110b
110a
115c
115c
115a
115a

FIG. 1

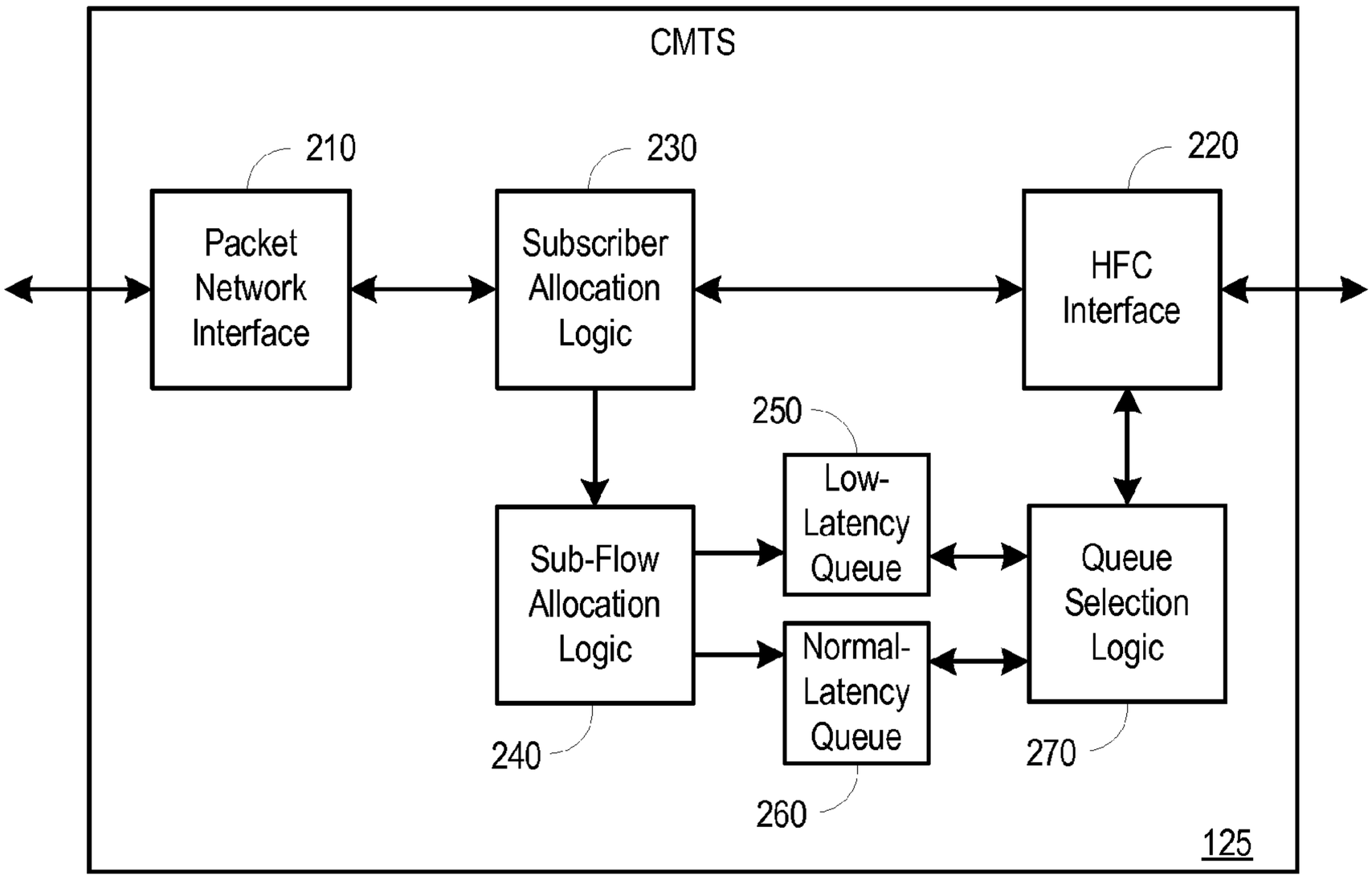

CMTS
210
Packet Network Interface
230
Subscriber Allocation Logic
220
HFC Interface
250
Low-Latency Queue
Normal-Latency Queue
Sub-Flow Allocation Logic
Queue Selection Logic
240
260
270
125

FIG. 2

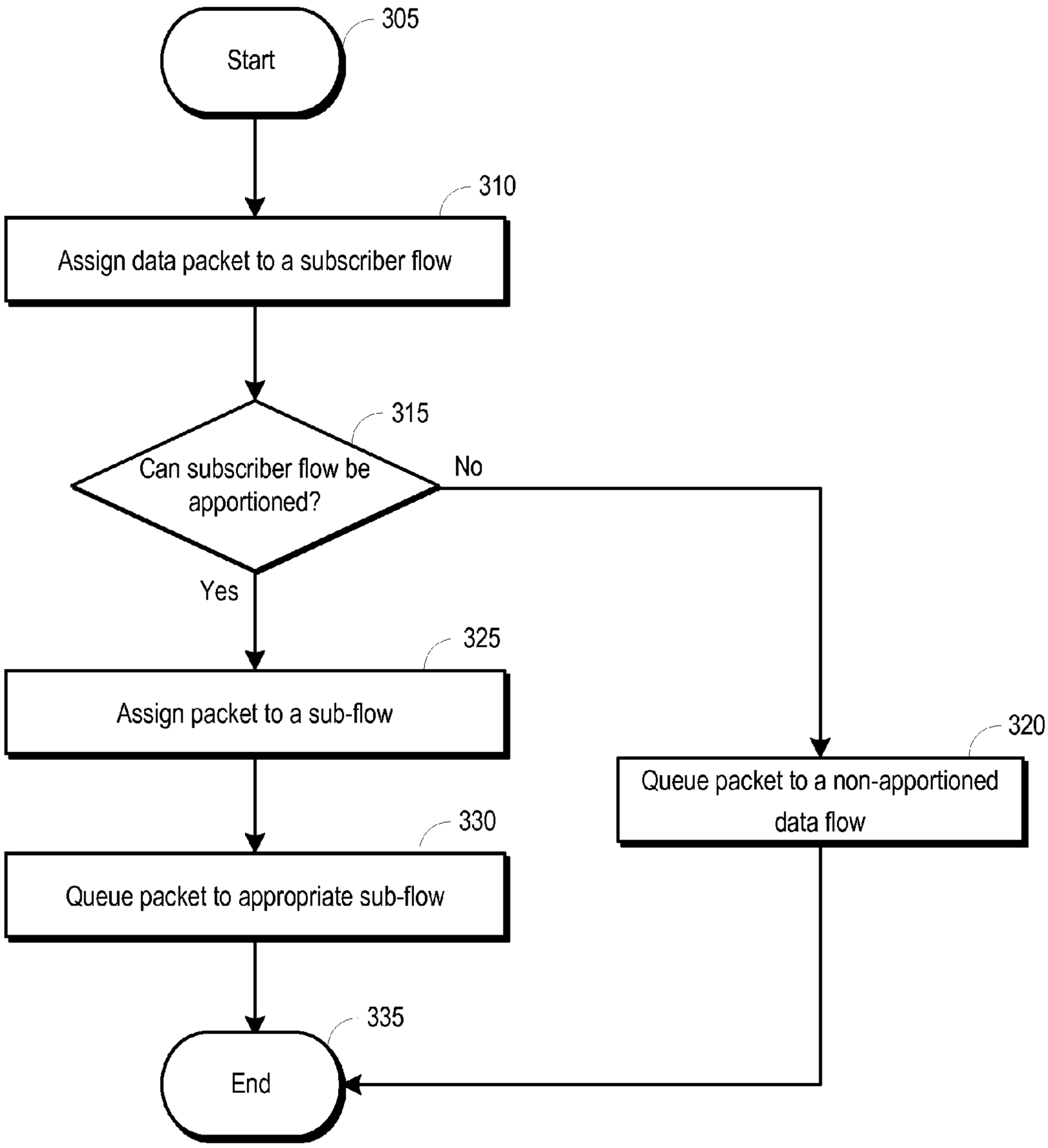

Start
305
Assign data packet to a subscriber flow
310
Can subscriber flow be apportioned?
315
No
Yes
Assign packet to a sub-flow
325
Queue packet to appropriate sub-flow
330
Queue packet to a non-apportioned data flow
320
End
335
300

FIG. 3

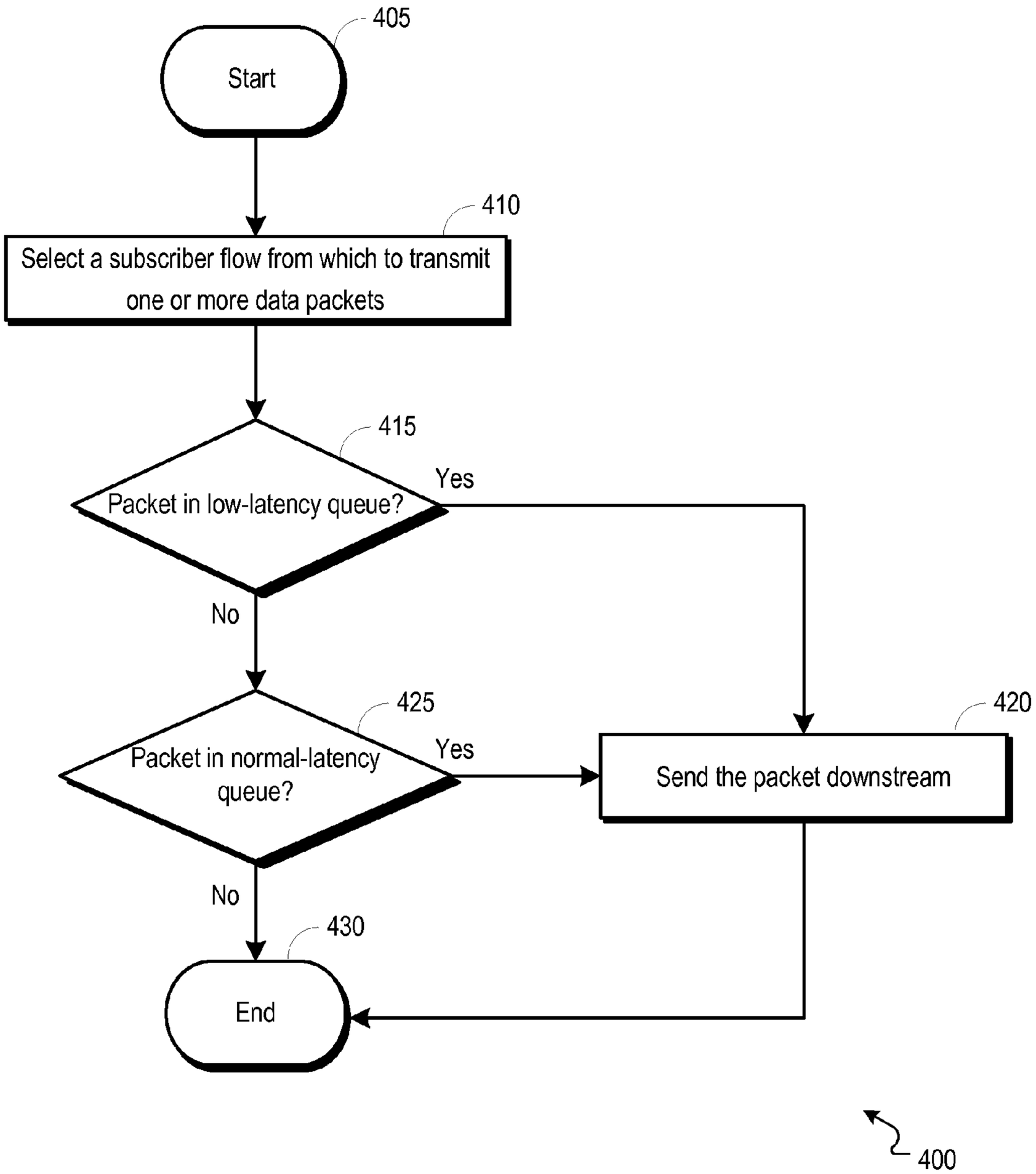

405
Start
410
Select a subscriber flow from which to transmit one or more data packets
415
Packet in low-latency queue?
Yes
No
425
Packet in normal-latency queue?
Yes
420
Send the packet downstream
No
430
End
400

FIG. 4

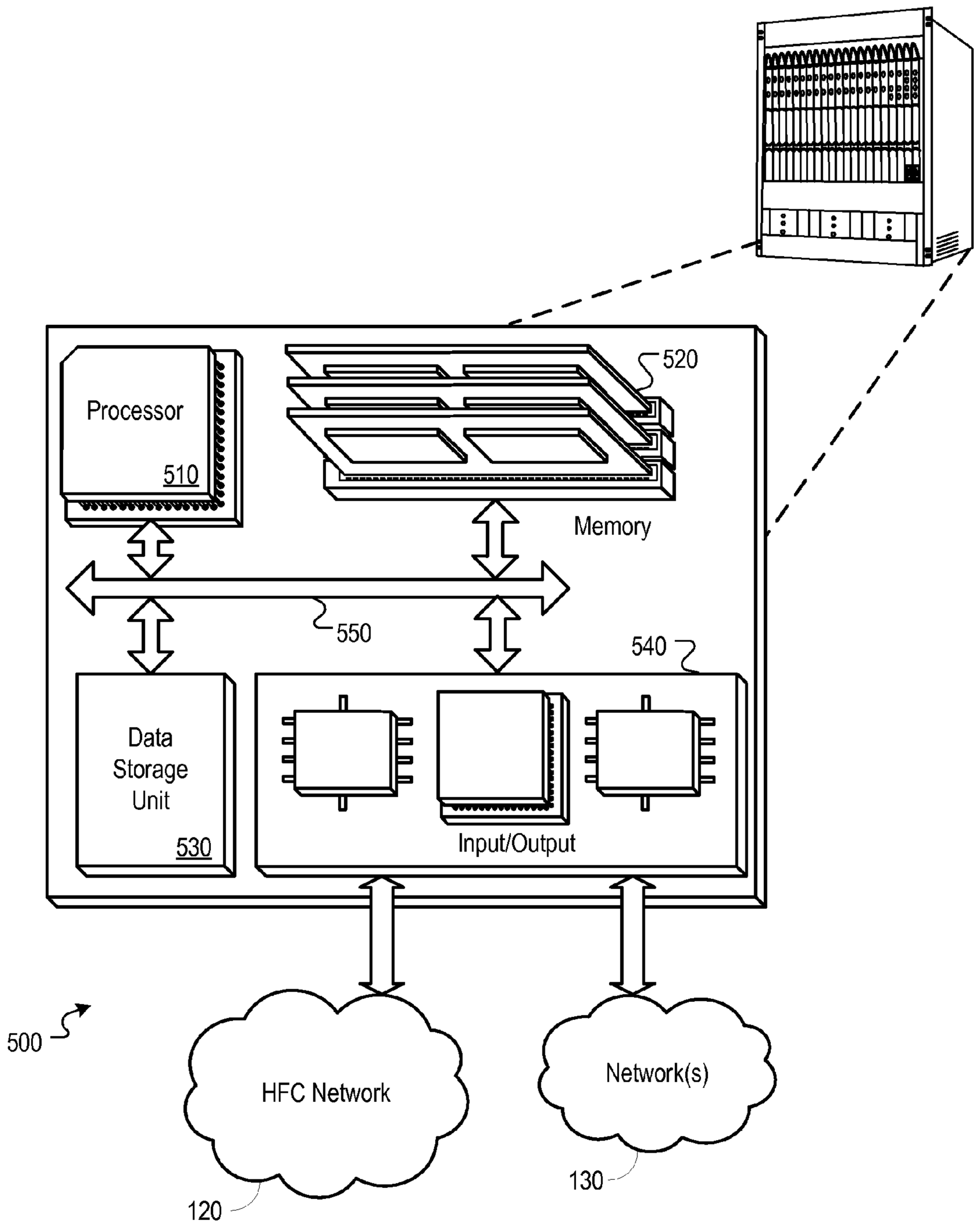

Processor
510
520
Memory
550
540
Data
Storage
Unit
530
Input/Output
500
HFC Network
120
Network(s)
130

FIG. 5

LOW-LATENCY QUALITY OF SERVICE

TECHNICAL FIELD

This disclosure relates to quality of service associated with low-latency packet flows.

BACKGROUND

The Data-Over-Cable Service Interface Specification (DOCSIS) was established by cable television network operators to facilitate transporting data traffic, primarily Internet traffic, over existing community antenna television (CATV) networks. In addition to transporting data traffic, as well as television content signals over a CATV network, multiple services operators (MSO) also use their CATV network infrastructure for carrying voice, video on demand (VoD) and video conferencing traffic signals, among other types.

Data traffic and signals are typically transported to subscribers in the form of internet protocol (IP) packets. Generally, when a transmission medium (e.g., cable, wireless router, EPON, etc.) is shared by a plurality of subscribers to access one or more of the various services (e.g., data traffic, television or voice signals, etc.) offered by a MSO, the resources associated with the transmission medium (e.g., maximum bitrate) are allocated between the one or more services being accessed by the plurality of subscribers at a specific time.

The need for low-latency packet flows varies with the type of application requesting a packet flow from a MSO. For example, online gaming generally requires a low-latency packet flow, while data transmissions via file transfer protocol (FTP) can be effectively completed through a normal or high-latency packet flow. Generally, when a transmission medium becomes congested with a plurality of data flows, the available resources of the transmission medium are allocated between the data flows, thus the bitrate available for each flow to utilize is typically reduced. When the bitrate available for a data flow is reduced to a rate lower than the rate at which data is requested, data packets are generally either dropped from the data flow or are buffered, thus increasing the latency of the data flow. A general prioritization of data requests from applications needing low-latency data flows can lead to an unfair disparity between the bitrate available to heavy users of low-latency data flows and the bitrate available to light users of low-latency data flows. Broad use of such a general prioritization over a transmission medium allows a subscriber's use of low-latency data flow to negatively impact the bitrate available to other subscribers accessing the same transmission medium. Thus, there is a need for improved quality of service for low-latency data flows.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a network environment operable to apportion available resources on a transmission medium between subscribers and to allocate apportioned resources between multiple sub-flows.

FIG. **2** is a block diagram illustrating an example cable modem termination system operable to apportion available resources for a transmission medium between subscribers and to allocate apportioned resources between multiple sub-flows.

FIG. **3** is a flowchart illustrating a process operable to assign a data packet to one of a plurality of data sub-flows.

FIG. **4** is a flowchart illustrating a process operable to apportion available resources for a transmission medium between subscribers and to allocate apportioned resources between a plurality of data sub-flows.

FIG. **5** is block diagram of a hardware configuration operable to apportion available resources for a transmission medium between subscribers and to allocate apportioned resources between multiple sub-flows.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

In some implementations of this disclosure, systems, methods, and computer readable media can operate to apportion available resources on a transmission medium between subscribers and to allocate the apportioned resources between various sub-flows. In general, when a transmission medium becomes congested, some data packets requested by subscribers are buffered or dropped rather than being transmitted in due course to the subscribers. Latency, or delay, created by the buffering of data packets can impede the use of applications that depend on low-latency packet flows (e.g., online gaming, video, voice, etc.), while other applications (e.g., FTP data transmissions) can run effectively with a buffered data flow.

In some implementations of this disclosure, queuing all packets directly to sub-flows, without dropping or delaying any packets, can solve the problem of dropping or delaying low-latency packets that exists in typical priority schedulers. In various implementations of this disclosure, a division of bandwidth between subscriber-flows can be accomplished by pulling packets from the subscriber-flows at an allowed rate (e.g., by pulling packets for each subscriber-flow from a low-latency queue first and a normal latency queue afterwards).

Data packets transmitted over the Internet can be encoded using a variety of signaling and transmission formats. For example, one format for carrying data packets is known as the internet protocol (IP). IP data is typically transported over the Internet using a transmission protocol known as the transmission control protocol (TCP). The transmission of IP data using TCP is often referred to as TCP/IP. Routers typically play a significant role in receiving TCP/IP data packets and directing individual data packets to their appropriate destinations based on information carried in the header portion of each data packet. The information in the header portion can be used by routers not only for routing data packets but can also be used for enforcement of a desired quality of service (QoS).

Generally, a certain type of header byte can be used for identifying various types of QoS services. This header byte has been defined and refined under different versions of the Internet Engineering Task Force (IETF) standards. Specifically, when used in IP version 4 (IPv4) applications, this header byte is often referred to as a type of service (TOS) packet header. When used in IP version 6 (IPv6) applications, the header byte may be referred to as a traffic class (TC) packet header. However, regardless of the labeling, the individual bits contained in the header byte are used in the same way in both IPv4 and IPv6. Generally, six of the high-order bits (bits **0-5**) are referred to as the differentiated services code point (DSCP) field, while the remaining two low-order bits (bits **5** and **6**) are either left unused or sometimes used for explicit congestion notification (ECN) purposes.

The DSCP field can be used for conveying QoS related information to one or more routers for purposes of carrying out tasks such as managing traffic congestion queues. In some

implementations, a router can use the DSCP field to identify and provide precedence to certain types of data packets while dropping certain other types of data packets. For example, a DSCP field can identify a data packet as either a low-latency packet or a normal-latency packet, and low-latency packets can be given priority over normal-latency packets during data transmission.

FIG. **1** is a block diagram illustrating a network environment **100** operable to apportion a transmission medium's available resources between multiple subscribers and to allocate a subscriber's apportioned resources between a plurality of sub-flows. In some implementations, a headend **105** can provide video, data and/or voice service(s) to customer premise equipment (CPE) devices **110***a-d* in one or more subscriber groups (e.g., service group(s)). The CPE devices can include, for example, a cable modem **110***a*, a set top box **110***b*, a wireless router including an embedded cable modem **110***c*, or a media terminal adapter (MTA) **110***d*, among others. A cable modem **110***a* can facilitate communications between the headend **105** and a computer and/or a gaming console **115***a*. A set top box **110***b* can facilitate communications between the headend **105** and a television or a digital video recorder **115***b*. A wireless router **110***c* can facilitate wireless communications between a headend **105** and a computer and/or a gaming console **115***c*. An MTA **110***d* can facilitate communications between a headend **105** and a telephone **115***d*.

In some implementations, the CPE devices **110***a-d* can communicate with the headend **105** via a hybrid fiber-coax (HFC) network **120** or any shared access broadband network. The headend **105** can include devices such as a cable modem termination system (CMTS) **125** and/or an edge quadrature amplitude modulation (EQAM) device (not shown), or a combined or converged device (not shown) including multiple edge and/or video or data processing functionalities. Headend devices can operate to facilitate communications between a network **130** and the CPE devices **110***a-d*. In various implementations, the network **130** can include one or more networks internal to the headend and/or one or more networks external to the headend (e.g., one or more extranets, the Internet, etc.).

Data services (e.g., voice over internet protocol (VoIP), internet protocol television (IPTV), online gaming, data transfer, etc.) can be handled by the headend **105** through a CMTS **125**. The CMTS **125** can receive data signals from external device(s) or nodes through network(s) **130**. The network(s) **130**, for example, can operate using internet protocol (IP), sending data packets to and receiving data packets from the headend **105**.

In some implementations, the CMTS **125** can perform packet conversion and packet addressing functions. For example, the CMTS **125** can identify the destination of a data packet (e.g., the subscriber addressed by the data packet) and can direct the data packet to an appropriate data flow (e.g., the data flow destined for the subscriber addressed by the data packet). In various implementations, the CMTS can identify the level of QoS treatment to be received by a data packet and the CMTS can flag the data packet accordingly. For example, using differentiated services code point (DSCP), the CMTS can classify each received data packet as a packet to receive low-latency treatment or a packet to receive best-effort treatment. The CMTS can, for example, classify a received data packet by changing a specific header bit in the data packet.

In some implementations, the CMTS **125** can perform packet latency functions (e.g., buffering or dropping packets). For example, when a transmission medium becomes congested to the point that it is incapable of transmitting a downstream data flow at a rate equal to or greater than the rate at which the data flow is arriving at the CMTS, the CMTS can delay the transmission of data packets in the data flow. In various implementations, the CMTS can apportion the resources of a transmission medium between subscribers accessing data via the transmission medium. For example, the CMTS can apportion the resources of a transmission medium into equal shares based on the number of subscribers using the medium or can apportion the resources according to one or more service levels purchased by subscribers (e.g., one level of subscribers can pay more to receive data at a rate of 10 megabits per second (Mbps) while another level of subscribers can pay less to receive data at a rate of 7 Mbps, etc.). In some implementations of this disclosure, the CMTS can select data packets to delay based on the level of QoS associated with the data packets. For example, when a transmission medium becomes congested, the CMTS can delay the transmission of data packets that are to receive normal-latency treatment (e.g., low QoS treatment) rather than delay the transmission of data packets that are to receive low-latency treatment (e.g., higher QoS treatment).

In some implementations, the CMTS **125** can determine how much bandwidth each downstream subscriber flow (e.g., subscriber, customer, etc.) is entitled to during periods in which the transmission medium becomes congested. In various implementations, the amount of bandwidth each downstream subscriber flow is entitled to can vary according to the demands from each of the subscriber flows. In some implementations, the allowed bitrate can be reduced to a percentage of the configured DOCSIS maximum-sustained-rate so that the total bandwidth stays within the capacity of the channel (s). For example, if the CMTS allows each subscriber flow 50% of its configured maximum-sustained-rate and if a particular subscriber flow has a configured maximum-sustained-rate of 6 megabits per second (Mbps), the particular subscriber flow will be allowed to transmit data at 3 Mbps based on the current level of congestion on the channel(s). Supposing that the particular subscriber flow consists of both a 2 Mbps video stream and an online gaming service requiring 2 Mbps, data will accumulate in the buffers at a rate of 1 Mbps (representing the difference between the 4 Mbps size of the particular subscriber flow and the 3 Mbps size allowed by the CMTS) until the subscriber flow reacts to the increasing latency (as packets build up due to arrival in the CMTS at 4 Mbps but exiting at only 3 Mbps).

By separating the subscriber flow into two sub-flows, one sub-flow for gaming and the other sub-flow for the remaining data traffic, the gaming traffic can be sent first so that the gaming traffic experiences no latency, and the video traffic experiences the full latency. Supposing that the channel congestion worsens and that the CMTS reduces the allowed bitrate of each subscriber flow further to 30% of the maximum-sustained-rate of each flow, then the particular subscriber flow will be allowed only 1.8 Mbps with which to transmit data packets in a downstream subscriber flow. If the gaming service still requires 2 Mbps, latency (the excess of the 2 Mbps requested for gaming over the 1.8 Mbps allowed) will begin to build up on the gaming flow and the video flow will stop functioning because the gaming flow is using up all of the available bitrate for the particular subscriber. In this example, subscriber flows other than this particular subscriber flow will not be impacted because the prioritization of a low-latency data flow is only applicable up to the bitrate allowed to the particular subscriber flow, based on the current level of congestion on the transmission medium.

FIG. **2** is a block diagram illustrating an example cable modem termination system **125** operable to apportion a trans-

mission medium's available resources between multiple subscribers and to allocate a subscriber's apportioned resources between a plurality of sub-flows. In some implementations, the CMTS **125** can include a packet network interface **210**, an HFC network interface **220**, subscriber allocation logic **230**, sub-flow allocation logic **240**, a low-latency queue **250**, a normal-latency queue **260**, and queue selection logic **270**. The packet network interface **210** can be operable, for example, to receive IP packets from an IP network (e.g., the internet). The received packets can be transmitted to an entity (e.g., CPE device) connected to an HFC network (e.g., HFC network **120** of FIG. **1**) or to the CMTS **125** itself.

The HFC network interface **220** can be operable to receive and transmit data over an HFC network. An HFC network typically operates by communicating DOCSIS packets from the CMTS to CPE devices using radio frequency (RF) signals. The HFC interface **220** can include components operable to encapsulate IP packets to produce DOCSIS packets. The HFC interface **220** can also operate to modulate the DOCSIS packets associated with one or more flows onto a RF channel and to multiplex or combine multiple RF channels for transmission over the HFC network. The HFC interface **220** can also operate to demodulate RF channels in order to recover DOCSIS packets received from subscriber devices (e.g., CPE devices **110***a-d*). In other implementations, the modulation and demodulation functions can be performed by an external device (e.g., a quadrature amplitude modulator (QAM) device).

Subscriber allocation logic **230** can be operable to assign a data packet to a specific data flow. For example, subscriber allocation logic can identify the destination (e.g., the subscriber or CPE device) associated with a data packet and can place the data packet into a data flow having the same destination as the data packet. In some implementations, subscriber allocation logic can determine whether the data flow can be apportioned into sub-flows. For example, some types of data packets can be subject to apportionment (e.g., data packets associated with online gaming, browser downloads/applications, etc.) while other types of data packets can be subject to transmission via a primary data flow rather than one of a plurality of sub-flows (e.g., VoIP, IPTV, etc.). Further, as an example, the service of assigning data packets to one or more of a plurality of sub-flows associated with a data flow destined for a subscriber can be offered to the subscriber at a premium. In some implementations, if the determination is made that a data packet is of the type that can be transmitted via a sub-flow and that the destination associated with the data packet is associated with a subscriber of the data flow apportionment service, the data packet can be sent from the subscriber allocation logic **230** to sub-flow allocation logic **240**. If the determination is made that a data packet is of the type that cannot be transmitted via a sub-flow or that the destination associated with the data packet is not associated with a subscriber of the data flow apportionment service, the data packet can be transmitted to the subscriber in a primary data flow via the HFC interface **220**.

Subscriber allocation logic **230** can, in some implementations, process data requests from one or more subscribers and allocate available resources on a transmission medium between the data requests. For example, subscriber allocation logic can establish a data flow associated with each of the subscribers and can apportion available resources associated with a transmission medium between the data flows based on the number of subscribers accessing the medium (e.g., for a transmission medium capable of transmitting data at a rate of 100 Mbps that is servicing **10** subscribers, subscriber allocation logic can establish a data flow capable of transmitting data at a rate of 10 Mbps for each subscriber accessing the medium).

Sub-flow allocation logic **240** can be operable to assign a data packet to a specific data sub-flow. In some implementations, sub-flow allocation logic can receive a primary data flow from subscriber allocation logic **230**, and the sub-flow allocation logic can apportion the primary data flow into a plurality of sub-flows. For example, sub-flow allocation logic can split a primary data flow into a sub-flow containing low-latency data packets and another sub-flow containing normal-latency data packets.

In some implementations, sub-flow allocation logic **240** can be operable to classify and/or flag a data packet based on the latency that is to be received by the data packet (e.g., classified or flagged as a low-latency packet or a normal-latency packet). For example, sub-flow allocation logic can identify the type of application requesting the data packet (e.g., online game, file download, etc.), determine whether the identified type is dependent on receiving a data flow with minimal delays, and can classify and/or flag the data packet appropriately. In various implementations, the determination whether a type of application is dependent on receiving a data flow with minimal delays can be based on a variety of factors and/or parameters including whether the application involves interactions between multiple locations or users, a threshold rate at which data needs to be received to support the application, or a level of priority given to the type of application, as well as others. In some implementations, a data flow can be classified as low-latency or normal-latency based upon the type of application requesting the data flow and/or the type of device requesting the data flow. Thus, data flows can be dynamically classified and reclassified in and out of a low-latency or normal-latency classification based on a change in the device requesting the data flow or a change in the application requesting the data flow. In other implementations, a data flow can be statically assigned to be a low-latency or normal-latency flow.

Sub-flow allocation logic **240** can also be operable to send a data packet to an appropriate queue based on a data sub-flow assigned to the data packet and/or latency classification. In some implementations, sub-flow allocation logic can direct separate sub-flows into separate queues. For example, sub-flow allocation logic can direct a sub-flow containing low-latency data packets into a low-latency queue **250** and can direct a sub-flow containing normal-latency data packets into a normal-latency queue **260**. While a low-latency queue **250** and a normal-latency queue **260** are shown, it should be understood that the CMTS **125** can include a multitude of various levels of queues that are associated with a variety of data packet classifications (e.g., latency, bitrate, etc.).

Queue selection logic **270** can include a packet transmission scheduling algorithm, and can be operable to select data packets from one or more queues and to transmit selected data packets downstream to a subscriber. In some implementations, queue selection logic can prioritize the one or more queues and can select and transmit data packets in a higher priority queue before selecting and transmitting data packets in a lower priority queue. For example, queue selection logic can prioritize the low-latency queue **250** over the normal-latency queue **260** and can accordingly select and transmit data packets from the low-latency queue before selecting and transmitting data packets from the normal-latency queue.

In some implementations, queue selection logic **270** can apportion a subscriber's allocated resources (e.g., resources allocated to the subscriber by subscriber allocation logic **230**) between sub-flows containing data packets addressed to that

subscriber. For example, when data addressed to a subscriber arrives at a rate greater than the bitrate that is allocated to the subscriber, the queue selection logic can use the subscriber's allowed bitrate to transmit data packets from the low-latency queue and can delay the transmission of data packets in the normal-latency queue where the subscriber's remaining bitrate, if any, is less than the rate at which normal-latency data packets are arriving for the subscriber. In various implementations, queue selection logic **270** can cap the bitrate available to transmit data packets in the low-latency queue **250**. For example, queue selection logic can cap the apportionment to the low-latency queue at a percentage of the total bitrate allocated to a subscriber. The cap implemented by the queue selection logic can be based on predetermined or variable factors or parameters.

FIG. **3** is a flowchart illustrating a process **300** operable to assign a data packet to one of a plurality of data sub-flows. The process **300** can start at stage **305** when data packets addressed to a certain subscriber are received by a CMTS (e.g., CMTS **125** of FIG. **1**). In some examples, the data packets addressed to a certain subscriber can be data packets that are requested by the subscriber. In other examples, the data packets addressed to a certain subscriber can be generated and addressed to the subscriber without being requested by the subscriber (e.g., multicast data packets).

After the data packets are received, the process **300** proceeds to stage **310**. At stage **310**, the data packets are assigned to a subscriber flow. The data packets can be assigned to a subscriber flow, for example, by subscriber allocation logic (e.g., subscriber allocation logic **230** of FIG. **2**). The data packets can be received from a network (e.g., IP network) via a packet network interface (e.g., packet network interface **210** of FIG. **2**). In some implementations, subscriber allocation logic can identify or otherwise direct the received data packets to the appropriate subscriber.

After data packets are assigned to a subscriber data flow, the process **300** proceeds to stage **315**. At stage **315**, a determination is made whether the subscriber flow can be apportioned into sub-flows. This determination can be made, for example, by subscriber allocation logic (e.g., subscriber allocation logic **230** of FIG. **2**). In some implementations, the determination whether a subscriber flow can be apportioned into sub-flows can be based on the type of application generating the data packets, the type of device to which the data packets are destined, the particular subscriber addressed by the data packets, and/or other factors or parameters. As an example, the service of prioritizing low-latency data flows for a subscriber can be offered by an MSO to subscribers who pay a premium for the service. As another example, the service of prioritizing low-latency data flows can be offered by an MSO for certain applications or devices. If a determination is made that the subscriber flow cannot be apportioned, the process **300** proceeds to stage **320**. At stage **320**, the data packets in the subscriber flow are queued in a non-apportioned data flow (e.g., a single data flow comprising all of the data packets addressed to a subscriber, regardless of differences between the latency treatment associated with each data packet). After the data packets in the subscriber flow are queued in the non-apportioned data flow, the process **300** can end at stage **335**.

If at stage **315**, a determination is made that the subscriber flow can be apportioned, the process **300** proceeds to stage **325**. At stage **325**, each data packet in the subscriber flow is assigned to one of a plurality of data sub-flows. For example, each data packet can be flagged or classified as either a low-latency or normal-latency data packet. Data packets can be flagged or classified as either low-latency or normal-latency, for example, by sub-flow allocation logic (e.g., sub-flow allocation logic **240** of FIG. **2**). In some implementations, sub-flow allocation logic can be operable to classify data packets into a variety of classifications other than low-latency or normal-latency. For example, data packets can be classified based on the QoS to be given to the data packets, the bitrate associated with the request for the data packets, as well as other classifications associated with data packets. In some implementations, data packets can be classified into three or more levels of priority, such that the highest priority receives low-latency treatment during extreme congestion of the transmission medium.

After data packets are assigned to one of a plurality of data sub-flows, the process **300** proceeds to stage **330**. At stage **330**, each data packet is sent to an appropriate queue. Each data packet can be sent to a queue, for example, by sub-flow allocation logic (e.g., sub-flow allocation logic **240** of FIG. **2**). In some implementations, sub-flow allocation logic can send data packets of the same type (e.g., low-latency or normal-latency) to the same queue. For example, sub-flow allocation logic can send data packets classified as low-latency data packets to a low-latency queue (e.g., low-latency queue **250** of FIG. **2**) and can send data packets classified as normal-latency data packets to a normal-latency queue (e.g., normal-latency queue **260** of FIG. **2**). It should be understood that data packets can be sent to a variety of queues and can be classified based on a variety of factors or parameters (e.g., narrowed latency priority levels, QoS treatment, requested bitrate, etc.). It should also be understood that separate queues can be established for each subscriber flow (e.g., each subscriber flow can have a low-latency and a normal-latency queue). After each data packet is sent to the appropriate queue, the process **300** ends at stage **335**.

FIG. **4** is a flowchart illustrating a process **400** operable to apportion available resources for a transmission medium between subscribers and to allocate apportioned resources between a plurality of data sub-flows. The process **400** can start at stage **405** when a channel from which to transmit one or more data packets is selected. A channel from which to transmit one or more data packets can be selected, for example, by a CMTS (e.g., CMTS **125** of FIG. **1**).

After a channel is selected for use in transmitting one or more data packets, the process **400** proceeds to stage **410**. At stage **410**, a subscriber flow is selected for use in transmitting one or more data packets. A subscriber flow can be selected, for example, by queue selection logic (e.g., queue selection logic **270** of FIG. **2**). In some implementations, the subscriber flow can be selected based on the bitrate allowed for the subscriber flow and the current level of congestion of the transmission medium servicing the subscriber flow.

After a subscriber flow from which to transmit one or more data packets is selected, the process **400** proceeds to stage **415**. At stage **415**, a determination is made whether a low-latency packet is present in the low-latency queue. This determination can be made, for example, by queue selection logic (e.g., queue selection logic **270** of FIG. **2**). If there is a low-latency packet in the low-latency queue, the process **400** proceeds to stage **420**. At stage **420**, the low-latency packet is sent downstream to the appropriate subscriber. The low-latency packet can be sent downstream, for example, by queue selection logic via an interface (e.g., HFC interface **220** of FIG. **2**). After the low-latency packet is sent downstream, the process **400** ends at stage **430**.

If at stage **415**, a determination is made that there is not a low-latency packet in the low-latency queue, the process **400** proceeds to stage **425**. At stage **425**, a determination is made whether a normal-latency packet is present in the normal-

latency queue. The determination whether a normal-latency packet is present in the normal-latency queue can be made, for example, by queue selection logic (e.g., queue selection logic **270** of FIG. **2**). If there is a normal-latency packet in the normal-latency queue, the process **400** proceeds to stage **420**. At stage **420**, the normal-latency packet is sent downstream to the appropriate subscriber. The normal-latency packet can be sent downstream, for example, by queue selection logic via an interface (e.g., HFC interface **220** of FIG. **2**). After the normal-latency packet is sent downstream, the process **400** ends at stage **430**.

If at stage **425**, a determination is made that there is not a normal-latency packet in the normal-latency queue, the process **400** ends at stage **430**. It should be understood that this is a simplification of the packet flow classification assignment and apportionment process for illustrative purposes. In normal operation, packets are being regularly received for multiple flows concurrently, and the process typically continues to receive new packets for classification and assignment.

It should be understood that the process **400** can include the additional stages of identifying one or more other queues, prioritizing the other queues, determining whether data packets are present in the other queues, and/or transmitting data packets from the other queues downstream.

FIG. **5** is a block diagram of a hardware configuration **500** operable to apportion a transmission medium's available resources between multiple subscribers and to allocate a subscriber's apportioned resources between a plurality of sub-flows. While a CMTS is shown, it should be understood that other kinds of devices can be operable to facilitate apportioning a transmission medium's available resources between multiple subscribers and allocating a subscriber's apportioned resources between a plurality of sub-flows. The hardware configuration **500** can include a processor **510**, memory **520**, a data storage unit **530**, and an input/output device **540**. Each of the components **510**, **520**, **530**, and **540** can, for example, be interconnected using a system bus **550**. The processor **510** can be capable of processing instructions for execution within the hardware configuration **500**. In one implementation, the processor **510** can be a single-threaded processor. In another implementation, the processor **510** can be a multi-threaded processor. The processor **510** can be capable of processing instructions stored in the memory **520** or on the data storage unit **530**.

The memory **520** can store information within the hardware configuration **500**. In one implementation, the memory **520** can be a computer-readable medium. In one implementation, the memory **520** can be a volatile memory unit. In another implementation, the memory **520** can be a non-volatile memory unit. In various implementations, the memory can be used to store computer program instructions and logic (e.g., subscriber allocation logic **230** of FIG. **2**, sub-flow allocation logic **240**, queue selection logic **270**, etc.).

In some implementations, the storage unit **530** can be capable of providing mass storage for the hardware configuration **500**. In one implementation, the storage unit **530** can be a computer-readable medium. In various other implementations, the storage unit **530** can, for example, include a hard disk device, an optical disk device, flash memory or some other large capacity storage device. In other implementations, the storage unit **530** can be a device external to the hardware configuration **500**.

The input/output device **540** can provide input/output operations for the hardware configuration **500**. In some implementations, the input/output device **540** can include one or more of a network interface device (e.g., an Ethernet card), a serial communication device (e.g., an RS-232 port), one or more universal serial bus (USB) interfaces (e.g., a USB 2.0 port) and/or a wireless interface device (e.g., an 802.11 card). For example, the input/output device can include an interface device operable to communicate with CPE devices through a content delivery network (e.g., HFC network **120**). In various implementations, the input/output device can include driver devices configured to request and to receive data from one or more networks **130**.

The subject matter of this disclosure, and components thereof, can be realized by instructions that upon execution cause one or more processing devices to carry out the processes and functions described above. Such instructions can, for example, comprise interpreted instructions, such as script instructions, e.g., JavaScript or ECMAScript instructions, or executable code, or other instructions stored in a computer readable medium.

Implementations of the subject matter and the functional operations described in this specification can be provided in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a tangible program carrier for execution by, or to control the operation of, data processing apparatus. The tangible program carrier can be a propagated signal or a computer readable medium. The propagated signal is an artificially generated signal (e.g., a machine generated electrical, optical, or electromagnetic signal) that is generated to encode information for transmission to suitable receiver apparatus for execution by a computer. The computer readable medium can be a machine readable storage device, a machine readable storage substrate, a memory device, a composition of matter effecting a machine readable propagated signal, or a combination of one or more of them.

The term "system processor" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The system processor can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

The processes and logic flows described in this specification are performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output

thereby tying the process to a particular machine (e.g., a machine programmed to perform the processes described herein). The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The elements of a computer typically include a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile communications device, a telephone, a cable modem, a set-top box, a mobile audio or video player, or a game console, to name just a few.

Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices (e.g., EPROM, EEPROM, and flash memory devices); magnetic disks (e.g., internal hard disks or removable disks); magneto optical disks; and CD ROM and DVD ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

Particular embodiments of the subject matter described in this specification have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results, unless expressly noted otherwise. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In some implementations, multitasking and parallel processing may be advantageous.

What is claimed is:

1. A computer-implemented method, comprising:

assigning one or more data packets to a respective one of a plurality of data flows, wherein each data flow is associated with a subscriber accessing a transmission medium;

determining a latency treatment associated with each of the one or more data packets;

modifying a single header bit of each respective one of the one or more data packets, wherein the modified header bit provides an indication of the latency treatment associated with the respective data packet;

assigning each respective one of the one or more data packets assigned to the respective data flow to a respective one sub-flow of one or more sub-flows, wherein the assignment of each respective one data packet of the one or more data packets to the respective one sub-flow is based upon the latency treatment indicated by the modified header bit associated with the respective data packet such that each of the one or more data packets assigned to the respective sub-flow are associated with the same one or more modified header bit;

associating each sub-flow with a priority level, wherein the priority level associated with each respective sub-flow is based upon the latency treatment indicated by the modified header bit associated with each of the one or more data packets that are assigned to the respective sub-flow;

apportioning resources associated with the transmission medium between subscribers accessing the transmission medium; and

when the respective one data flow requires consumption of more resources than are apportioned to the subscriber associated with the respective one data flow, dedicating resources apportioned to the subscriber to transmit, to the subscriber, one or more data packets assigned to a first sub-flow comprising one or more data packets associated with a first latency treatment and delaying a transmission, to the subscriber, of one or more data packets assigned to a second sub-flow comprising one or more data packets associated with a second latency treatment, wherein the first sub-flow is associated with a higher priority level than the second sub-flow when the first latency treatment comprises a lower latency level than the second latency treatment.

2. The computer-implemented method of claim **1**, wherein the one or more data packets are assigned to the respective one data flow based on a destination associated with the one or more data packets that are assigned to the respective one data flow.

3. The computer-implemented method of claim **1**, wherein the latency treatment associated with each respective one of the one or more data packets is based on a type of application generating the respective data packet.

4. The computer-implemented method of claim **1**, wherein the latency treatment associated with each respective one of the one or more data packets is based on a level of service subscribed to by a subscriber addressed by the respective data packet.

5. The computer-implemented method of claim **1**, wherein the first sub-flow comprises data packets that are to be treated as requiring low-latency, and the first sub-flow is associated with a high priority level; and the second sub-flow comprises data packets that are to be treated as allowing higher latency, relative to the low-latency associated with the data packets of

the first sub-flow, the data packets of the second sub-flow being associated with a lower priority level, relative to the high priority level associated with the first sub-flow.

6. The computer-implemented method of claim **1**, further comprising:

transmitting one or more data packets from a sub-flow associated with the highest priority level and delaying transmitting one or more data packets from one or more sub-flows associated with lower priority levels, wherein the one or more data packets from the sub-flow associated with the highest priority level are transmitted at a predetermined bitrate, the predetermined bitrate being a percentage of the total bitrate allocated to the subscriber associated with the respective one data flow.

7. A system, comprising:

one or more interfaces configured to receive one or more data packets and to transmit the one or more data packets within one or more data flows or sub-flows to one or more devices;

storage configured to store computer program instructions; and

a processor configured to execute said computer program instructions, the computer program instructions being configured to cause the processor to:

assign one or more data packets to a data flow to be subdivided into two or more sub-flows, wherein the data flow is associated with a subscriber accessing a transmission medium;

determine a latency treatment associated with each of the one or more data packets assigned to the data flow;

modify a single header bit of each respective one of the one or more data packets assigned to the data flow, wherein the modified header bit provides an indication of the latency treatment associated with the respective data packet;

assign each respective one of the one or more data packets assigned to the data flow to a respective one of two or more sub-flows, wherein the assignment of each respective data packet of the one or more data packets assigned to the data flow to the respective sub-flow is based upon the latency treatment indicated by the modified header bit associated with the respective data packet such that each of the one or more data packets assigned to the respective sub-flow are associated with the same modified header bit;

associate each sub-flow with a priority level, wherein the priority level associated with each respective sub-flow is based upon the latency treatment indicated by the modified header bit associated with each of the one or more data packets that are assigned to the respective sub-flow; and

when the data flow requires consumption of more resources than are apportioned to the subscriber associated with the data flow, dedicating resources apportioned to the subscriber to transmit one or more data packets assigned to a first sub-flow comprising one or more data packets associated with a first latency treatment and delaying a transmission of one or more data packets assigned to a second sub-flow comprising one or more data packets associated with a second latency treatment, wherein the first sub-flow is associated with a higher priority level than the second sub-flow when the first latency treatment comprises a lower latency level than the second latency treatment.

8. The system of claim **7**, wherein the one or more data packets are assigned to the data flow based on a destination associated with the one or more data packets that are assigned to the data flow.

9. The system of claim **7**, wherein the latency treatment associated with each respective one of the one or more data packets assigned to the data flow is based on a type of application generating the respective data packet or on a level of service subscribed to by a subscriber addressed by the respective data packet.

10. The system of claim **7**, wherein the first sub-flow comprises data packets that are to be treated as requiring low-latency, and the first sub-flow is associated with a high priority level; and the second sub-flow comprises data packets that allow higher latency, relative to the low-latency associated with the data packets of the first sub-flow, the second sub-flow being associated with a lower priority level, relative to the high priority level associated with the first sub-flow.

11. The system of claim **7**, wherein the computer program instructions are further configured to cause the processor to:

apportion resources associated with the transmission medium between subscribers accessing the transmission medium; and

transmit one or more data packets from a sub-flow associated with the highest priority level and delay a transmission of one or more data packets from one or more sub-flows associated with lower priority levels, wherein the one or more data packets from the sub-flow associated with the highest priority level are transmitted at a predetermined bitrate, the predetermined bitrate being a percentage of the total bitrate allocated to the subscriber associated with the data flow.

12. One or more non-transitory computer readable media comprising code configured to execute on one or more processors, the one or more non-transitory computer readable media being configured to cause the one or more processors to perform operations comprising:

assigning one or more data packets to one of a plurality of data flows;

determining a latency treatment associated with each of the one or more data packets;

modifying a single header bit of each respective one of the one or more data packets, wherein the modified header bit provides an indication of the latency treatment associated with the respective data packet;

assigning each respective data packet of the one or more data packets to a respective one of two or more sub-flows associated with the data flow to which the one or more data packets are assigned, wherein the assignment of each respective data packet of the one or more data packets to the respective one sub-flow is based upon the latency treatment indicated by the modified header bit associated with the respective data packet such that each of the one or more data packets assigned to the respective sub-flow are associated with the same modified header bit;

associating each of the two or more sub-flows with a priority level, wherein the priority level associated with each respective sub-flow is based upon the latency treatment indicated by the modified header bit associated with each of the one or more data packets that are assigned to the respective sub-flow;

transmitting each of the one or more data packets in an order based on the priority level associated with the respective sub-flow to which each respective data packet of the one or more data packets is assigned;

limiting the transmitting of the one or more data packets based on an allowed bitrate associated with the data flow to which the one or more data packets are assigned, wherein, the allowed bitrate is determined based on a configured bitrate associated with the data flow and on a current level of congestion associated with a transmission medium used by the plurality of data flows; and

dropping data packets from a sub-flow associated with a low priority level.

13. The one or more non-transitory computer-readable media of claim **12**, wherein the one or more data packets are assigned to the data flow based on a destination associated with the one or more data packets that are assigned to the data flow.

14. The one or more non-transitory computer-readable media of claim **12**, wherein the latency treatment given to each respective one data packet of the one or more data packets is based on a type of application generating the respective one data packet or on a level of service subscribed to by a subscriber addressed by the respective one data packet.

15. The one or more non-transitory computer-readable media of claim **12**, wherein a first sub-flow comprises data packets that receive low-latency treatment and is associated with a high priority level and a second sub-flow comprises data packets that allow higher latency, relative to the low-latency associated with the data packets of the first sub-flow, the second sub-flow being associated with a lower priority level, relative to the high priority level associated with the first sub-flow.

16. The one or more non-transitory computer-readable media of claim **12**, further comprising:

apportioning resources associated with the transmission medium between subscribers accessing the transmission medium; and

transmitting one or more data packets from a sub-flow associated with the highest priority level and delaying transmitting one or more data packets from one or more sub-flows associated with lower priority levels, wherein the one or more data packets from the sub-flow associated with the highest priority level are transmitted at a predetermined bitrate, the predetermined bitrate being a percentage of the total bitrate allocated to a subscriber associated with the data flow.

* * * * *