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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 July 2007 (26.07.2007)

PCT

(10) International Publication Number
WO 2007/084707 A2

(51) International Patent Classification:
H04J 1/16 (2006.01) *H04L 12/56* (2006.01)

(21) International Application Number:
PCT/US2007/001516

(22) International Filing Date: 19 January 2007 (19.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/335,973 20 January 2006 (20.01.2006) US

(71) Applicant *(for all designated States except US)*: **ANAGRAN, INC.** [US/US]; 2055 Woodside Road, Suite 200, Redwood City, CA 94061 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ROBERTS, Lawrence G.** [US/US]; 170 Sunrise Drive, Woodside, CA 94062 (US). **HENDERSON, Alex E.** [US/US]; 130 Golden Hills Drive, Portola Valley, CA 94028 (US).

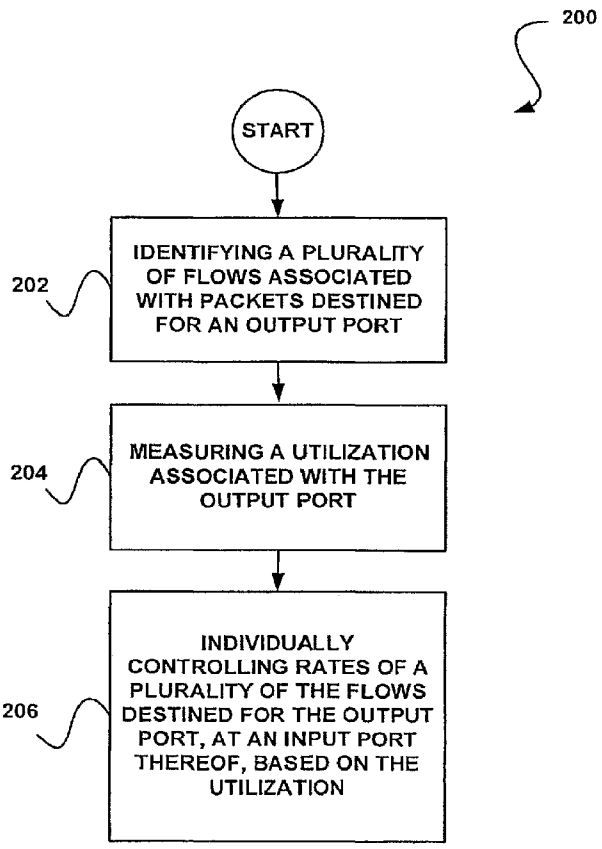
(74) Agent: **KOTAB, Dominic M.**; Zilka-kotab, PC, P.O. Box 721120, San Jose, CA 95172-1120 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR CONTROLLING OUTPUT PORT UTILIZATION



(57) Abstract: A system, method and computer program product are provided. In use, a plurality of flows associated with packets destined for an output port is identified. A utilization associated with the output port is further measured. Thus, rates of a plurality of the flows destined for the output port may be individually controlled at an input port thereof, based on the utilization to ensure that the utilization remains less than 99.9% and avoid buffering more than 400 packets with a correspondingly low delay.

WO 2007/084707 A2

FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR CONTROLLING OUTPUT PORT UTILIZATION

BACKGROUND AND FIELD OF THE INVENTION

The present invention relates to routers, and more particularly, to routing flows of packets.

SUMMARY

A system, method and computer program product are provided. In use, a plurality of flows associated with packets destined for an output port are identified. A utilization associated with the output port is further measured. Thus, rates of a plurality of the flows destined for the output port may be individually controlled at an input port thereof, based on the utilization to ensure that the utilization remains less than 99.9% and avoid buffering more than 400 packets with a correspondingly low delay.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a network architecture, in accordance with one embodiment.

Figure 2 shows a method for controlling the delay and utilization at an output port, in accordance with one embodiment.

Figure 3 shows a router system for controlling the delay and utilization at an output port, in accordance with one embodiment.

DETAILED DESCRIPTION

Figure **1** illustrates a network architecture **100**, in accordance with one embodiment. As shown, a plurality of networks **102** is provided. In the context of the present network architecture **100**, the networks **102** may each take any form including, but not limited to a local area network (LAN), wireless network, wide area network (WAN) such as the Internet, etc.

Coupled to the networks **102** are server computers **104** which are capable of communicating over the networks **102**. Also coupled to the networks **102** and the server computers **104** is a plurality of client computers **106**. Such client computers **106** may each include a desktop computer, lap-top computer, hand-held computer, mobile phone, hand-held computer, personal video recorder (PVR), a digital media [e.g. compact disc (CD), digital video disc (DVD), MP3, etc.] player, printer, and/or any other type of logic.

In order to facilitate communication among the networks **102**, at least one router **108** is coupled between the networks **102**. In the context of the present description, such router **108** may include any hardware and/or software capable of facilitating the communication of packets from one point in the network architecture **100** to another. More information regarding various features for enhancing such functionality will be set forth hereinafter in greater detail.

Figure **2** shows a method **200** for controlling the delay and utilization at an output port, in accordance with one embodiment. As an option, the present method **200** may be implemented in the context of the architecture and environment of Figure **1**. Of course, however, the method **200** may be carried out in any desired environment.

As shown, a plurality of flows associated with packets destined for an output port are identified. Note operation **202**. In the context of the present description, such packet may refer to any unit of information capable of being communicated in a computer

network (e.g. see, for example, the networks **102** of Figure **1**, etc.). For example, in one illustrative embodiment, the packet may include an Internet Protocol (IP) packet.

Further in the context of the present description, the term flow refers to a collection of packets that relate to a common data transfer. In various optional embodiments, however, the flow may include a bit-stream of some arbitrary length and constitute a single data transfer. In such embodiments, each flow may be broken into packets for the purpose of facilitating delay reduction and error recovery.

Still yet, the output port may refer to any logic output associated with a router (e.g. see, for example, the router **108** of Figure **1**, etc.) and/or component thereof.

As shown in operation **204**, a utilization associated with the output port is further measured. In the context of the present description, such utilization may refer to any unit of measurements that reflects an amount of usage of the capacity of the output port that is being utilized. Of course, such measurement may be performed in any desired manner that identifies such utilization.

In various embodiments, the utilization may be measured periodically. For example, the utilization may be measured periodically every millisecond. Of course, it should be noted that the measurement may occur in accordance with any desired timing.

Thus, in operation **206**, rates of a plurality of the flows destined for the output port may be individually controlled at an input port thereof, based on the utilization to ensure that the utilization remains less than (and/or possibly equal to) 99.9% and avoid buffering more than 400 packets. In the context of the present description, such rates may each refer to any unit of data, packets, etc. utilizing the output port that is a function of time. Similar to the aforementioned output port, the input port may refer to any logical input associated with a router (e.g. see, for example, the router **108** of Figure **1**, etc.) and/or component thereof.

In one embodiment, the rates may be individually by controlled by rejecting or discarding the new and/or existing flows and/or components thereof. Of course, however, the rates may be controlled in any way that impacts utilization of the output port.

In various embodiments, some or all of the flows destined for the output port may be individually controlled. Further, such control may be independent in nature, such that the rates of the plurality of flows are individually controlled in a different manner. See Table 1 which illustrates an exemplary set of individually controlled flow rates. Further, it should be noted that the rates may be more stringently controlled, so as to ensure that the utilization remains less than 95%, 90%, 80%, or less.

Table 1

Output port 1	Flow rate 1
Output port 2	Flow rate 2
Output port 3	Flow rate 3
Output port 4	Flow rate 4

Of course, such table is illustrative in nature and should not be construed as limiting any manner.

Thus, in some embodiments, buffering at the output port may be reduced and/or eliminated, thereby providing a lower delay associated with packet transmission, where such lower delay "corresponds" to (e.g. results from, is a function of, and/or is associated with, etc.) such buffer reduction. For example, as mentioned previously, such buffering may amount to less than 400 packets. In still other embodiments, such buffering may require less than 1MB of storage capacity and, in some embodiments require 250KB or less. Even still, the buffering may be further reduced to less than 100 packets in another embodiment, less than 50 packets in another embodiment, less than 20 packets in another embodiment, less than 10 packets in another embodiment, etc.

This may be possibly beneficial particularly when transmitting streaming media (e.g. audio streaming media, video streaming media, etc.). More information regarding such optional feature will be set forth in greater detail during reference to Figure 3.

In one exemplary embodiment, the rates may be individually controlled utilizing an input flow manager. Further, the measuring may carried out utilizing an output flow manager. Still yet, the utilization may be reported by the output flow manager to a network processing unit (NPU). In such embodiment, such NPU may include one or more processors capable of routing packets. Thus, the input flow manager may be adapted to control the rates based on input from the NPU.

In the context of the present description, the term routing refers to any communication of packets from one point in a network architecture to another, that involves the identification of a destination address by at least being capable of identifying a "longest prefix" match. In various exemplary embodiments that are not to be construed as limiting with respect to the above definition of routing, the aforementioned "longest prefix" match may require only one memory cycle, but may, in other embodiments, require 3-5 memory cycles. Further, the match may, but need not necessarily, be a complete match. Instead, it may involve just enough bytes of the address to determine a desired output port. For example, European communications may be sent to one port so there is no need to keep track of all the Europe addresses, but rather just a first part correlating to Europe, etc.

In other exemplary embodiments that are, again, not to be construed as limiting with respect to the above definition of routing, a second router function may involve determining if traffic to or from certain addresses are to be blocked and/or discarded in relation to a denial of service (DOS) function. Optionally, more than mere addresses may be used to make such decision and an associative memory may be used to accomplish the same. Of course, various other functions may be included, such as a function for

prioritizing traffic so that certain types of packets receive a lower delay during the course of traffic shaping, etc.

More illustrative information will now be set forth regarding various optional architectures and features with which the foregoing technique may or may not be implemented, per the desires of the user. It should be strongly noted that the following information is set forth for illustrative purposes and should not be construed as limiting in any manner. Any of the following features may be optionally incorporated with or without the exclusion of other features described.

Figure **3** shows a router system **300** for controlling a utilization at an output port, in accordance with one embodiment. As an option, the present system **300** may be implemented in the context of the architecture and environment of Figures **1-2**. Of course, however, the system **300** may be carried out in any desired environment. Further, the foregoing definitions may equally apply in the present description.

As shown, the router system **300** includes an input trunk **301** and an output trunk **303**. The input trunk **301** is coupled to an input transceiver **302** for receiving packets via the input trunk **301** and feeding the same to an input framer **304** for performing packet framing. In one embodiment, such packet framing may refer to the method by which packets are sent over a serial line. For example, framing options for T1 serial lines may include D4 and ESF. Further, framing options for E1 serial lines may include CRC4, no-CRC4, multiframe-CRC4, and multiframe-no-CRC4.

Further included is an input flow manager **306** coupled between the input framer **304** and a switching fabric architecture **312**. In the present embodiment, the switching fabric architecture **312** may include hardware (e.g. switching integrated circuit, etc.) and/or software that switches incoming packets (e.g. moves incoming packets out via an appropriate output port, etc.) in a manner that will soon become apparent. For controlling such switching fabric architecture **312**, a central processing unit **311** may be in

communication therewith. Further, the input flow manager **306** may further be coupled to input flow memory **308**.

Still yet, an NPU **310** may be in communication with the input flow manager **306** and/or switching fabric architecture **312** for routing incoming packets. Further included is an output flow manager **316** coupled between the switching fabric architecture **312** and an output framer **318**. Similar to the input flow manager **306**, the output flow manager **316** includes output flow memory **316** for performing similar functions.

Finally, the output framer **318** is coupled to an output transceiver **320** which communicates via the output trunk **303**. While the various components are shown to be included in a single package associated with the router system 300, it should be noted that such components may be distributed in any desired manner.

As an option, the input flow manager **306** may function to efficiently control IP flow routing. Specifically, the input flow manager **306** may determine whether a flow associated with a received packet is new. To accomplish this, the input flow manager **306** extracts a header of the packet. In one embodiment, such header may include various fields including, but not limited to a destination address, source address, protocol, destination port, source port, and/or any other desired information.

Next, one or more of the fields are combined in the form of a hash. As an option, such hash may take the form of a 32-bit flow identifier. The input flow manager **306** then uses the hash (e.g. a lower 21 bits of the 32-bit flow identifier, etc.), and does a memory look up in a hash table stored in the input flow memory **308**. Specifically, in one exemplary embodiment, a binary tree is followed using a remaining 11 bits of the 32-bit flow identifier until a pointer to a flow record is located in the input flow memory **308** that makes an exact match with the destination address, source address, protocol, destination port, and source port, etc. Such record (if it exists) constitutes a flow record for the identified flow.

If it is determined that the flow associated with the packet is new, at least a portion of the packet may be routed utilizing the NPU **310**. If, on the other hand, it is determined that the flow associated with the packet is not new, at least a portion of the packet may be routed or switched utilizing the switching fabric architecture **312**, which may cost at least 10 times less than the first module.

Further during operation, the output flow manager **316** is equipped with output flow management functionality. Specifically, the output flow manager **316** may operate in conjunction with the remaining components shown in Figure **3** to carry out the functionality associated with the method **300** of Figure **3**.

In one exemplary implementation of the method **300** of Figure **3**, a queuing buffer size and packet delay associated with an output port may be controlled. In particular, the output frame manager **316** measures an output utilization on each of a plurality of output ports. Such output port-specific utilization may then be reported by the output frame manager **316** to the NPU **310** periodically (e.g. every millisecond, etc.).

When a new flow is identified in the manner described above, it is assigned to a particular output port. At this time, by virtue of the foregoing measurement, the utilization of such output port is known. With this knowledge, a rate may be determined for the flow that will not overload the output port beyond 99.9%.

It should be noted that there are two basic types of flows, namely those that a network controls the rate (e.g. TCP, typical file transfers, etc.), and those where a sender controls the rate (e.g. UDP, typical voice and video streaming media, etc.). For those flows that the network controls, a rate may be set that will not overload the output port.

On the other hand, for flows that the user controls the rate, lost packets may be harmful. Thus, it may be beneficial to reject new flows, if they would overload the output port. This may be done when a first packet of a flow is identified. If it is

discarded, no flow record need necessarily be made, and no further packets need necessarily be accepted for that flow until there is sufficient capacity.

Thus, the input frame manager **308** may be informed by the NPU **310** of the rate and/or whether to accept the new flow. The input frame manager **308** may then manage the rates by rejecting and/or discarding new flows to ensure that a sum of the rates associated with each output port is kept under a predetermined utilization (e.g. 95%, etc.). Given this technique, a queue at the output trunk **303** may be designed to be small (e.g. 100 packets or less, etc.) and an associated delay low (e.g. 100 microseconds or less, 100 packets of 1200 bytes at 10 Gbps, etc.).

Thus, a significant improvement in both buffer expense and packet delay may optionally be achieved with the present approach. These results are facilitated by separating the foregoing functionality associated with the input frame manager **308** with respect to the NPU **310**, so that flows to the NPU **310** may be governed in the foregoing manner.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. For example, any of the network elements may employ any of the desired functionality set forth hereinabove. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

CLAIM(S)

What is claimed is:

1. A method, comprising:

identifying a plurality of flows associated with packets destined for an output port;

measuring a utilization associated with the output port; and

individually controlling rates of a plurality of the flows destined for the output port, at an input port thereof, based on the utilization to ensure that the utilization remains less than 99.9% and avoid buffering more than 400 packets with a correspondingly low delay.

2. The method of Claim 1, wherein the rates of all of the flows destined for the output port are individually controlled.

3. The method of Claim 1, wherein the rates of the plurality of flows destined for the output port are each individually controlled in a different manner, at the input port thereof.

4. The method of Claim 1, wherein the rates of the plurality of flows destined for the output port are individually controlled, at the input port thereof, based on the utilization to ensure that the utilization remains less than 95%.

5. The method of Claim 1, wherein the rates of the plurality of flows destined for the output port are individually controlled, at the input port thereof, based on the utilization to avoid use of more than 1MB of buffer capacity.

6. The method of Claim 1, wherein the rates of the plurality of flows destined for the output port are individually controlled, at the input port thereof, based on the utilization to avoid use of more than 100KB of buffer capacity.

7. The method of Claim 1, wherein the packets together comprise streaming media.

8. The method of Claim 7, wherein the streaming media includes audio streaming media.

9. The method of Claim 7, wherein the streaming media includes video streaming media.

10. The method of Claim 1, wherein the controlling is carried out utilizing an input flow manager.

11. The method of Claim 10, wherein the measuring is carried out utilizing an output flow manager.

12. The method of Claim 11, wherein the utilization is reported by the output flow manager to a network processing unit (NPU).

13. The method of Claim 12, wherein the input flow manager controls the rates based on input from the NPU.

14. The method of Claim 1, wherein the utilization is measured periodically.

15. The method of Claim 14, wherein the utilization is measured periodically every millisecond.

16. The method of Claim 1, wherein the controlling includes rejecting a new flow.

17. The method of Claim 1, wherein the controlling includes discarding a new flow.

18. A computer program product embodied on a computer readable medium, comprising:

computer code for identifying a plurality of flows associated with packets destined for an output port;

computer code for measuring a utilization associated with the output port; and

computer code for individually controlling rates of a plurality of the flows destined for the output port, at an input port thereof, based on the utilization to ensure that the utilization remains less than or equal to 99.9% and avoid buffering more than 400 packets with a correspondingly low delay.

19. A system, comprising:

an input flow manager for identifying a plurality of flows associated with packets destined for an output port; and

an output flow manager for measuring a utilization associated with the output port;

wherein the input flow manager individually controls rates of a plurality of the flows destined for the output port, at an input port thereof, based on the utilization to ensure that the utilization remains less than 99.9% and avoid buffering more than 400 packets with a correspondingly low delay.

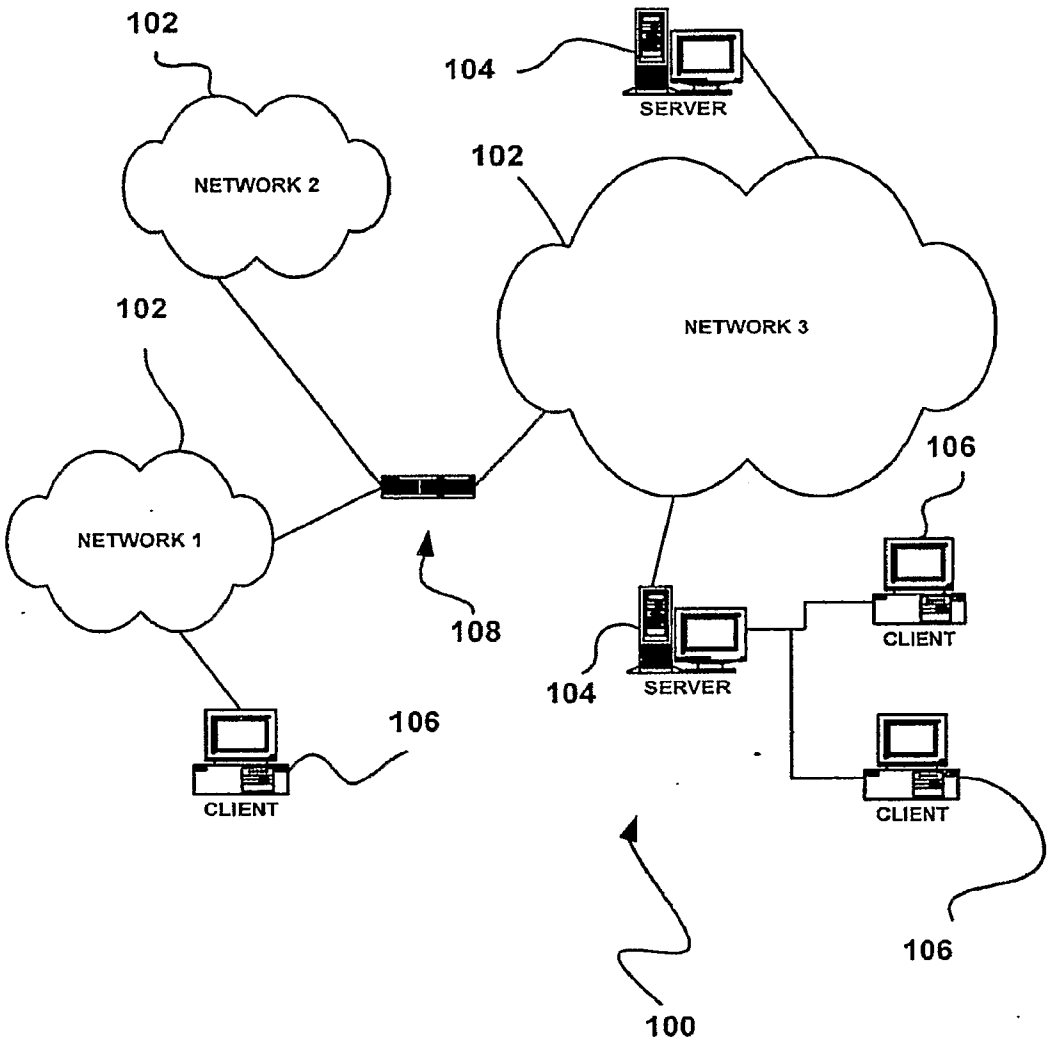

102
104
SERVER
NETWORK 2
102
NETWORK 3
102
106
NETWORK 1
108
CLIENT
104
SERVER
106
CLIENT
CLIENT
106
100

FIGURE 1

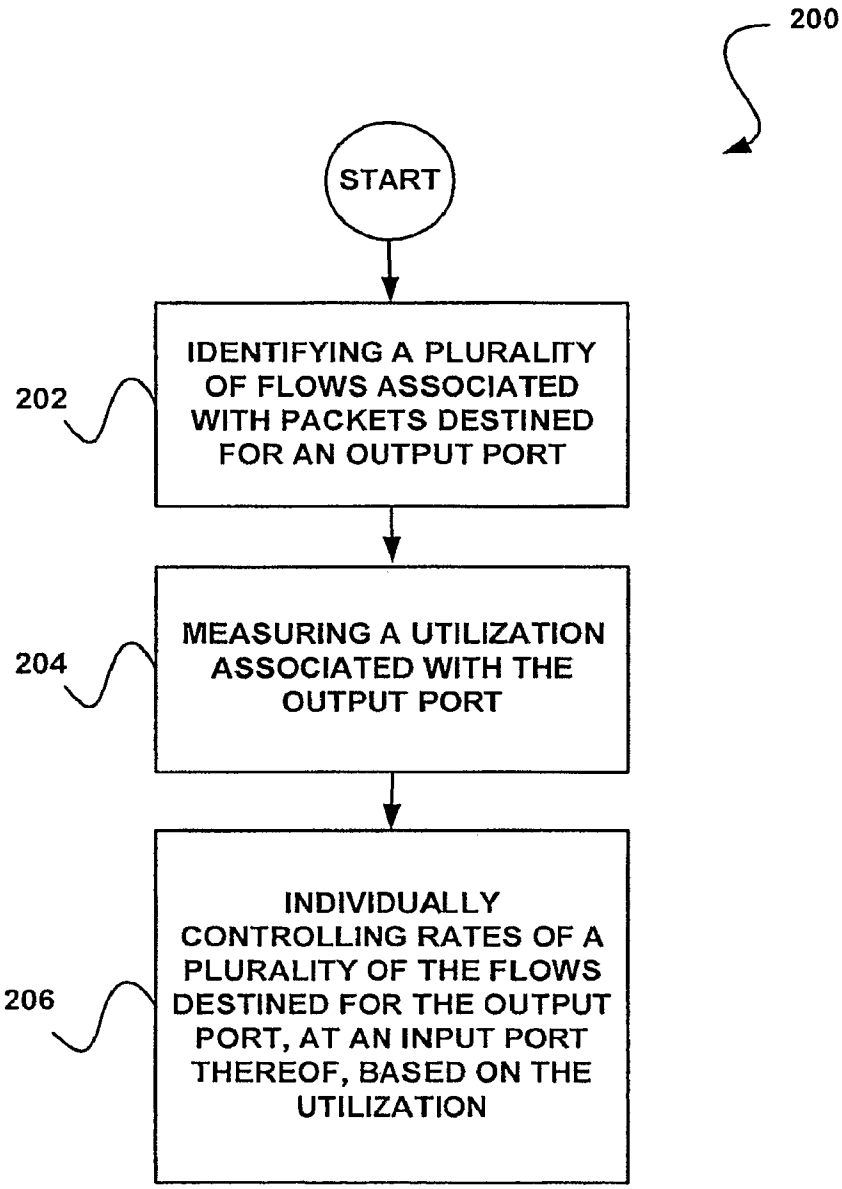

200
START
IDENTIFYING A PLURALITY OF FLOWS ASSOCIATED WITH PACKETS DESTINED FOR AN OUTPUT PORT
202
MEASURING A UTILIZATION ASSOCIATED WITH THE OUTPUT PORT
204
INDIVIDUALLY CONTROLLING RATES OF A PLURALITY OF THE FLOWS DESTINED FOR THE OUTPUT PORT, AT AN INPUT PORT THEREOF, BASED ON THE UTILIZATION
206

FIGURE 2

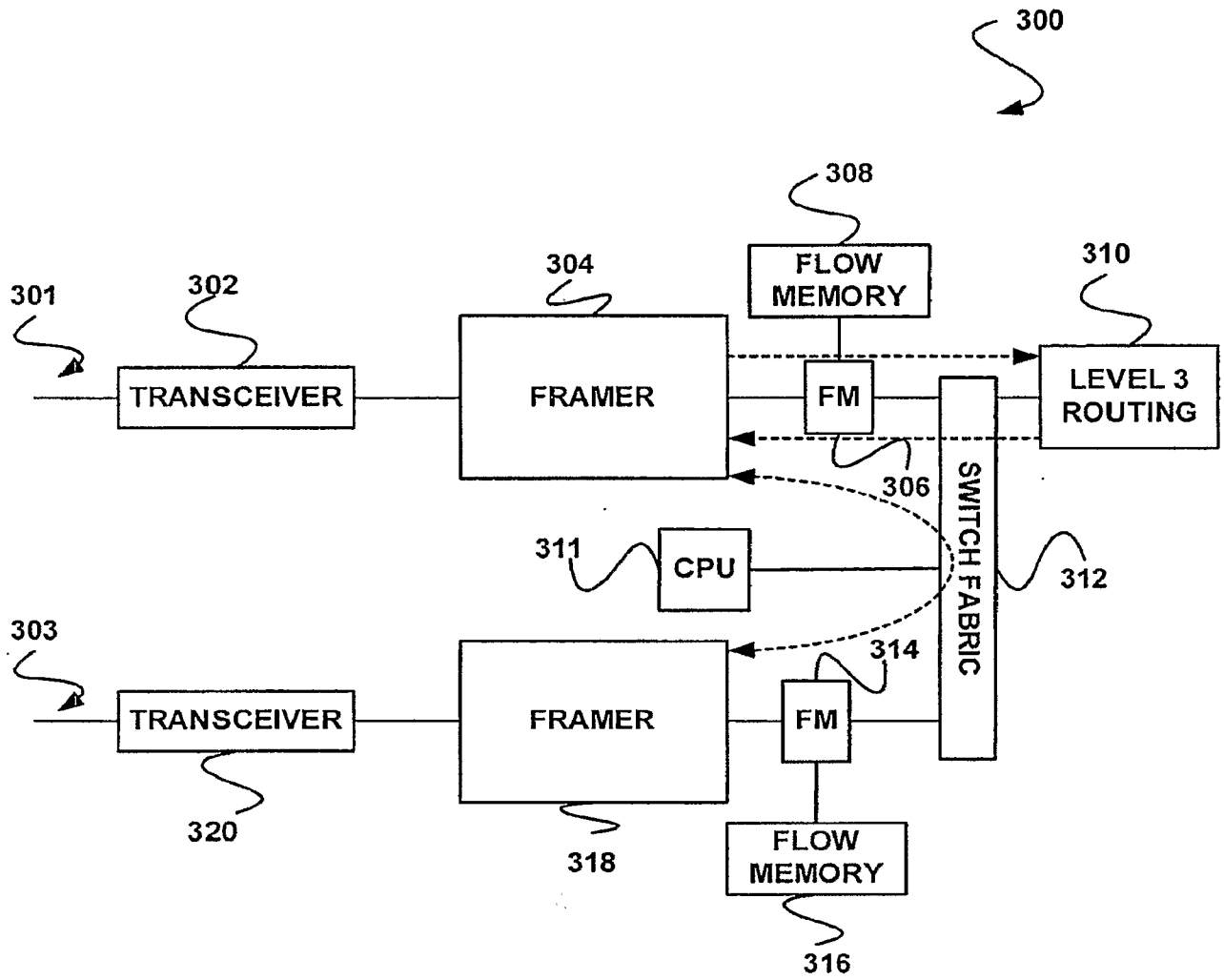

300
308
FLOW MEMORY
310
304
301
302
LEVEL 3 ROUTING
TRANSCEIVER
FRAMER
FM
306
SWITCH FABRIC
311
CPU
312
303
314
TRANSCEIVER
FRAMER
FM
320
318
FLOW MEMORY
316

FIGURE 3