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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 July 2016 (07.07.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/109104 A1

(51) International Patent Classification:
H04L 12/26 (2006.01) *H04L 12/935* (2013.01)
H04L 12/825 (2013.01) *H04L 12/931* (2013.01)

(21) International Application Number:
PCT/US2015/063520

(22) International Filing Date:
2 December 2015 (02.12.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/584,816 29 December 2014 (29.12.2014) US
14/584,824 29 December 2014 (29.12.2014) US

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5op7, Redwood Shores, California 94065 (US).

(72) Inventors: **SRINIVASAN, Arvind**; 1075 Happy Valley Avenue, San Jose, California 95129 (US). **CASTIL, Carlos**; 500 Oracle Parkway, M/S 5op7, Redwood Shores, California 94065 (US).

(74) Agents: **MEYER, Sheldon, R.** et al.; Tucker Ellis LLP, One Market Plaza, Steuart Tower, Suite 700, San Francisco, California 94105 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: SYSTEM AND METHOD FOR SUPPORTING EFFICIENT VIRTUAL OUTPUT QUEUE (VOQ) RESOURCE UTILIZATION IN A NETWORKING DEVICE

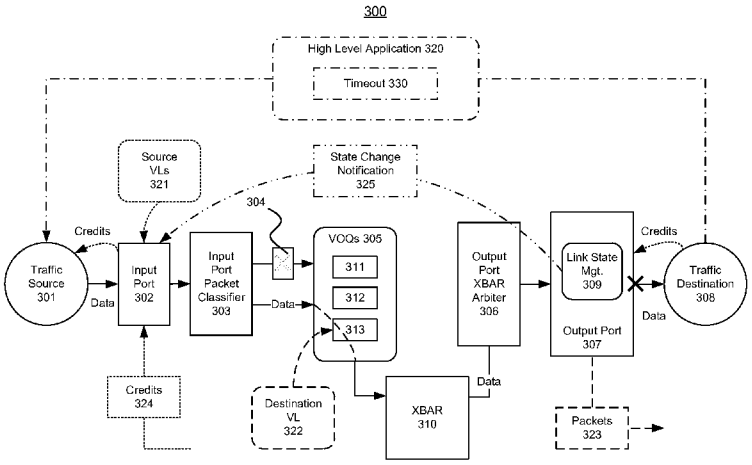

FIGURE 3

(57) Abstract: A system and method can support packet switching in a network environment. A networking device, such as a network switch, which includes a crossbar fabric, can be associated with a plurality of input ports and a plurality of output ports. Furthermore, the networking device can detect a link state change at an output port that is associated with the networking device. Then, the networking device can notify one or more input ports, via the output port, of the link state change at the output port.

WO 2016/109104 A1

SYSTEM AND METHOD FOR SUPPORTING EFFICIENT VIRTUAL OUTPUT QUEUE (VOQ) RESOURCE UTILIZATION IN A NETWORKING DEVICE

Copyright Notice:

Field of Invention:

[0002] The present invention is generally related to computer systems, and is particularly related to a high performance system in a cloud environment.

Background:

[0003] As larger cloud computing architectures are introduced, the performance and administrative bottlenecks associated with the traditional network and storage have become a significant problem. A high performance system can provide excellent processing speeds, significantly faster deployments, instant visuals for in-depth analysis, and manageable big data capability. This is the general area that embodiments of the invention are intended to address.

Summary:

[0004] Described herein are systems and methods that can support packet switching in a network environment. A networking device, such as a network switch, which includes a crossbar fabric, can be associated with a plurality of input ports and a plurality of output ports. Furthermore, the networking device can detect a link state change at an output port that is associated with the networking device. Then, the networking device can notify one or more input ports, via the output port, of the link state change at the output port. Additionally, the output port can provide one or more credits to an output scheduler, and the output scheduler allows one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

Brief Description of the Figures:

[0005] **Figure 1** shows an illustration of supporting a high performance system in a network environment, in accordance with an embodiment of the invention.

[0006] **Figure 2** shows an illustration of supporting a network switch in a high performance system, in accordance with an embodiment of the invention.

[0007] **Figure 3** shows an illustration of handling a link state change in a network environment, in accordance with an embodiment of the invention.

[0008] **Figure 4** shows an illustration of managing data flows in a high performance system, in accordance with an embodiment of the invention.

[0009] **Figure 5** illustrates an exemplary flow chart for handling a link state change in a network switch, in accordance with an embodiment of the invention.

[0010] **Figure 6** shows an illustration of managing credit for handling a link state change in a network environment, in accordance with an embodiment of the invention.

[0011] **Figure 7** shows an illustration of supporting credit management in a network switch, in accordance with an embodiment of the invention.

[0012] **Figure 8** illustrates an exemplary flow chart for supporting credit management in a network switch, in accordance with an embodiment of the invention.

Detailed Description:

[0013] The invention is illustrated, by way of example and not by way of limitation, in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" or "some" embodiment(s) in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0014] The description of the invention as following uses the InfiniBand (IB) network switch as an example for a high performance networking device. It will be apparent to those skilled in the art that other types of high performance networking devices can be used without limitation.

[0015] Described herein are systems and methods that can support packet switching in a network environment, such as a cloud environment.

High Performance System

[0016] **Figure 1** shows an illustration of supporting a high performance system in a network environment, in accordance with an embodiment of the invention. As shown in Figure 1, a high performance system 100 can include a plurality of host machines 101-103 (or servers) that are interconnected via a network switch fabric 110.

[0017] The network switch fabric 110 in the high performance system 100 can be responsible for directing the traffic movement between various virtual machines (VMs) 111-113 (and/or virtualized applications) that are running on the various host machines 101-103.

[0018] In accordance with an embodiment of the invention, the network switch fabric 110 can be based on the InfiniBand (IB) protocol, which can manage the peer-to-peer credit exchanges and provides lossless end-to-end connectivity. Thus, various networking devices in the network switch fabric 110 can maintain credit consistency under different conditions for supporting the data transfer in the high performance system 100.

[0019] Additionally, each physical IB link can be divided into multiple virtual link (VLs) in order to provide quality of service (QoS) for traffic between various VMs 111-113 (and/or applications).

For example, the network packet streams 120 between the host machines 101-103 can represent an aggregation of different services that the different VMs 111-113 and applications may desire. Furthermore, the individual packet streams 120, which are transmitted within the aggregated network pipes between the different source and destination pairs, can meet different service requirements (or even conflicting service requirements).

InfiniBand (IB) Network Switch

[0020] **Figure 2** shows an illustration of supporting a network switch in a high performance system, in accordance with an embodiment of the invention. As shown in Figure 2, a network device, such as an IB network switch 220 in a high performance system 200, can be responsible for directing data traffic from various traffic sources 201 and 211 to various traffic destinations 208 and 218.

[0021] For example, the IB network switch 220, which supports a large number of ports, such as the input ports 202 and 212 and the output ports 207 and 217, can be based on a crossbar (XBAR) fabric 210.

[0022] As shown in Figure 2, the input port 202 can receive various incoming data packets from the traffic source 201 using the source VLs 221, and the input port 212 can receive various data packets from the traffic source 211 using the source VLs 231. Also, the output port 207 can send outgoing data packets to the traffic destination 208 using the destination VLs 227, and the output port 217 can send outgoing data packets to the traffic destination 218 using the destination VLs 237.

[0023] Furthermore, the IB switch 220 can meet the different QoS demands, which supports the optimal usages of available network fabric resources. For example, the IB switch 220 may re-map an incoming VL for a packet (i.e. a source VL) to a different outgoing VL for the packet (i.e. a destination VL), based on the service levels (SL) of the traffic that is associated with an application.

[0024] In accordance with an embodiment of the invention, each of the input ports 202 or 212 can take advantage of an input port packet classifier 203 or 213, which can determine an output port for each incoming packet. For example, the input port packet classifiers 203 can determine an output port for each packet received at the input port 202 (and can use a port filter 204 to remove one or more packets), and the input port packet classifiers 213 can determine an output port for each packet received at the input port 212 (and can use a port filter 214 to remove one or more packets).

[0025] Additionally, the input port classifier 203 or 213 can determine multiple output destination ports for each multi-destination packet (such as for multicasting and broadcasting) that arrive at the input ports 202 or 212. The port filter 204 can remove one or more destination ports from the port list for the given packet. Furthermore, a multi-destination packet may be

dropped if all the destination ports are removed from the list. Otherwise, the packet can be queued for the available destination ports, which can be a subset of the originally classified port list (by the input port packet classifier).

[0026] On per input port basis, the input port 202 or 212 can store the received packets in an ingress buffer, e.g. the virtual output queues (VOQs) 205 or 215, before transmitting the received packets to a traffic destination 208 or 218 (e.g. via an output port 207 or 217). As shown in Figure 2, the packets received at the input port 202 can be stored in the VOQs 205 and the packets received at the input port 212 can be stored in the VOQs 215.

[0027] Additionally, each of the ingress buffers (e.g. the VOQs 205 or 215) may include a number of queues, each of which can be responsible for handling packets targeting a destination VL associated with an output port (e.g. the VLs 227 on the output port 207 and the VLs 237 on the output port 217). Thus, the total number of the queues on per input port basis can be the product of the number of the output ports and the number of the destination VLs supported on each output port. As a result, the system may require a large number of queues for each input port 202 or 212, if the number of ports and the number of VLs supported on each port are large.

[0028] In accordance with an embodiment of the invention, the VOQs 205 and 215 can be implemented using a shared memory structure, and the utilization of each queue in the VOQs 205 and 215 can be traffic dependent. For example, a VOQ resource can represent the number of the memory blocks, which are consumed when an incoming packet is queued (i.e. the receipt of a network packet) and eventually freed up when the packet is dequeued (i.e. the delivery of the packet to an output port). Thus, the utilization of the VOQ resource can be a function of the traffic patterns.

[0029] In accordance with an embodiment of the invention, the system can schedule the input ports 202 and 212 and direct the movement of the packets stored in the VOQs 205 and 215 toward the output ports 207 and 217. The drain rate of each queue in the ingress buffer may depend on the destination VLs and the output ports that the packets target.

[0030] As shown in Figure 2, each output port 207 or 217 can take advantage of an output scheduler (such as an output port XBAR arbiter 206 or 216). The output port XBAR arbiter 206 or 216 can make decisions that are related to the packet movement based on various criteria, such as the fullness of various VOQs and the available credits on the destination VLs.

[0031] In accordance with an embodiment of the invention, the IB network switch 220 can maintain credit consistency under different conditions. As shown in Figure 2, on the receive side of the IB network switch 220, the credits can be maintained consistent based on the incoming source VLs 221and 231 of the incoming packets; and on the transmit side of the IB network switch 220, the credits can be maintained consistent based on the destination VLs 227 and 237 of the outgoing packets.

[0032] Furthermore, on per input port basis, the queuing of each incoming packet can be

performed based on the source VL of the packet. Thus, the system can perform various credit accounting operations based on the source VLs 221 or 231. For example, for the purpose of credit accounting, a VOQ set can be assigned to each source VL in the IB network switch 220.

Link State Change in a Network Switch

[0033] **Figure 3** shows an illustration of handling a link state change in a network environment, in accordance with an embodiment of the invention. As shown in Figure 3, in a high performance system 300, a data flow in an IB network switch can involve an input port 302 and an output port 307, via a crossbar (XBAR) fabric 310.

[0034] The input port 302 can advertise one or more credits to and receives one or more data packets from a remote sender, such as the traffic source 301. The output port 307 can send one or more data packets to and receives one or more credits back from a remote receiver, such as the traffic destination 308.

[0035] Furthermore, the input port 302 can take advantage of an input packet classifier 303, which can determine one or more destinations for each incoming packet (and can use a port filter 304 to remove one or more packets). Additionally, the input port 302 can store the received packets in an ingress buffer, such as the virtual output queues (VOQs) 305, before forwarding the packets to the different output ports.

[0036] As shown in Figure 3, the VOQs 305 can include a plurality of queues 311-313, each of which can store packets targeting a different destination VL on the output ports. For example, the queue 313 can be responsible for storing packets targeting destination VL 322 on the output port 307.

[0037] In accordance with an embodiment of the invention, the traffic source 301 may not know whether the traffic destination 308 is reachable at the time when the traffic source 301 sends the packets. Thus, when the output port 307 goes down, the traffic source 301 may continually send more packets, which can result in the unnecessary high (or even wasteful) utilization of the VOQ resources for the packets that may eventually be dropped.

[0038] For example, when the output port 307 is down, the output port 307 can drain the packets 323 that arrive. As the packets 323 are drained, the credits 324, which are released, can be returned to the source VLs 321 on the input port 302. Since the traffic source 301 may not be aware that the output port 307 goes down, the traffic source 301 may continually send more packets to the input port 302 as long as enough credits are available, even though these packets may eventually be drained out at the output port 307.

[0039] Moreover, other output ports, which are part of the same VOQ structure, may not be able to utilize the VOQ resources, since the VOQ resources associated with the source VLs 321 may continually be consumed by the packets that are eventually dropped at the output port 307.

[0040] Furthermore, when the output port 307 goes down, it may take a long period of time for the high level applications 320 to be able to handle the link state changes, since the timeout

330 setting for the high level applications 320 tends to be relatively large. In the meantime, the traffic source 301 may keep on sending packets at a high speed (e.g. 100G per second). Thus, the incoming traffic can easily overwhelm the VOQ resources.

[0041] In accordance with an embodiment of the invention, the output port 307 can perform the link state management 309, and notify the input port 302 with regarding to the link state changes. For example, the output port 307, which detects the link state change, can broadcast the state change notification 325 across all VOQs (e.g. VOQs 305), e.g. via an output port arbiter 306. Link state management 309 includes, for example, access to a link state database table which describes each status of each link, and detection of change in each state. Eventually, the state change notification 325 may reach the input port 302 (and various other input ports).

[0042] As shown in Figure 3, the input port 302 can prevent the received packets from being presented at the output port 307, which is down. For example, the input packet classifier 303 can configure and/or use a mask (e.g. an output port filter mask based on the broadcast signal) as a final check before queuing the received packets into the VOQs 305.

[0043] Additionally, the input port 302 may drop the packets targeting the output port 307, before they are enqueued into the VOQ 305. These packets, which are dropped due to the going down of the output port 307, may not consume any VOQ space. Correspondently, the credits associated with these dropped packets can be returned to the traffic source 301 right away.

[0044] Thus, the system can prevent the VOQ resources from being wasted for storing the packets that may eventually be dropped.

[0045] **Figure 4** shows an illustration of managing data flows in a high performance system, in accordance with an embodiment of the invention. As shown in Figure 4, a network device, such as an IB network switch 420 in a high performance system 400, can be responsible for directing traffic from various remote senders, such as the traffic sources 401 and 411, to various remote receivers, such as the traffic destinations 408 and 418.

[0046] Furthermore, the IB network switch 420, which is based on a crossbar (XBAR) fabric 410, can support a large number of ports (with multiple VLs), such as the input ports 402 and 412 and the output ports 407 and 417.

[0047] As shown in Figure 4, each of the input ports 402 or 412 can advertise one or more credits to and receives one or more data packets from the traffic source 401 or 411. Each of the output port 407 or 417 can send one or more data packets to and receives one or more credits back from the traffic destination 408 or 418.

[0048] Additionally, each of the input ports 402 and 412 can take advantage of an input port packet classifier 403 or 413, which can determine an output port for each incoming packet. On per input port basis, the packets can be stored in an ingress buffer, e.g. the virtual output queues (VOQs) 405 or 415, before being transmitted to a traffic destination 408 or 418 (via the output port 407 or 417).

[0049] In accordance with an embodiment of the invention, the system can manage data flows and VOQ resources when one or more output ports 407 or 417 are going through link state changes (such as link up/down).

[0050] As shown in Figure 4, each output port 407 or 417 can perform the link state management 409 and 419. When an output port 407 or 417 detects any changes in the link state, the output port 407 or 417 can notify an output scheduler, such as an output port arbiter 406 or 416, which can broadcast the state change notifications, across all VOQs 405 and 415 (eventually to the different input ports 402 and 412).

[0051] Furthermore, the input port 402 or 412, which receives the state change notification, can prevent the received packets from being presented at the output port 407 or 417. For example, the input packet classifier 403 or 413 can configure an output port filter mask based on the broadcast signal, and use the mask for the port filter 404 or 414 as a final check before queuing the packets into the VOQs 405 or 415.

[0052] Additionally, the input port 402 or 412 can drop the packets targeting the output port 407 or 417 before these packets are queued into the VOQ 405 or 415. These packets, which are dropped due to the link state changes at the output port 407 or 417, may not consume any VOQ space. Correspondently, the credits associated with these packets can be returned right away.

[0053] Thus, the high performance system 400 can prevent the VOQ resources from being wasted for storing the packets that may eventually be dropped.

[0054] **Figure 5** illustrates an exemplary flow chart for handling a link state change in a network switch, in accordance with an embodiment of the invention. As shown in Figure 5, at step 501, the system can provide a networking device, which is associated with a plurality of input ports and a plurality of output ports. For example, the system activates the networking device, and the networking device becomes ready for operation accordingly. Furthermore, at step 502, the system can detect a link state change at an output port that is associated with the networking device. Then, at step 503, the output port can notify one or more input ports of the link state change at the output port.

Credit management in a Network Switch

[0055] **Figure 6** shows an illustration of managing credit for handling a link state change in a network environment, in accordance with an embodiment of the invention. As shown in Figure 6, in a high performance system 600, a data flow in an IB network switch can involve an input port 602 and an output port 607, via a crossbar (XBAR) fabric 610.

[0056] The input port 602 can advertise one or more credits to and receives one or more data packets from a remote sender, such as the traffic source 601. The output port 607 can send one or more data packets to and receives one or more credits back from a remote receiver, such as the traffic destination 608.

[0057] Additionally, the input port 602 can take advantage of an input port packet classifier 603, which can determine one or more destinations for each incoming packet (and can use a port filter 604 to remove one or more packets). On per input port basis, the packets can be stored in an ingress buffer, such as the virtual output queues (VOQs) 605, before being transmitted to the destination.

[0058] As shown in Figure 6, the ingress buffer, such as the virtual output queues (VOQs) 605, can include a plurality of queues 611-613. For example, the queue 613 can store the packets that are targeting the destination VL 622 on the output port 607.

[0059] In accordance with an embodiment of the invention, an output scheduler, such as an output port arbiter 606, can schedule the delivery of various packets from the different VOQs (including the queues other than the plurality of queues 611-613) toward the output port 607.

[0060] Furthermore, the output port arbiter 606 can select an input port from the different input ports on a network switch and can select a destination VL for delivering one or more packets targeting the output port 607, based on various criteria (such as available credits 626).

[0061] In accordance with an embodiment of the invention, the system can provide a framework that can provide an abstraction to the scheduling layer within the various output port crossbar arbiters. The system can achieve the link state abstraction by presenting the available credits 626 to the output scheduler, so that the output scheduler can be agnostic to any physical link state changes.

[0062] As shown in Figure 6, in order to maintain the credit consistency, the output port arbiter 606 can consider the available credits 626 in reaching its scheduling decisions. Additionally, the entire link related state management 609 can be performed within the physical output port 607. Also, the output port 607 can perform credit state management 629 independently.

[0063] In accordance with an embodiment of the invention, the system can provide an interface 639 on the output port 607 for indicating the maximum credit values to the output port arbiter 606. For example, the interface 639 can reside between the port logic and the output port arbiter 606.

[0064] When the output port arbiter 606 receives the initial credits 628, the output port arbiter 606 can lock the values for the initial credits 628 as the maximum credits that can be consumed (until the next time when a new set of initial values are presented).

[0065] Thus, the system can prevent various potential race conditions that are due to the asynchronous nature of the link state change and packet scheduling (e.g. the conditions may be caused by the inflight packets and the overflow of the credits when they are returned).

[0066] For example, when the link is up (or active) with the traffic moving, all updates on the initial credits 628 can be presented to the output port arbiter 606 based on the values coming from the remote destination 608. For example, these values can simply pass through the

interface 639. Then, the output port arbiter 606 can derive the values of the available credits 626 based on the information provided by the remote destination 608.

[0067] As shown in Figure 6, when the link between the output port 607 and the remote traffic destination 608 is active (i.e. when the output port 607 is up), the output port arbiter 606 can schedule the input port 602 to deliver one or more packets, which are stored in the queue 613, to the selected destination VL 622 on the output port 607.

[0068] Then, the remote traffic destination 608 can release the credits back to the output port 607, as the outgoing packets (or data) are drained. Additionally, the output port arbiter 606 can use the released credit to schedule the queue 613 to deliver more packets to the selected destination VL 622 on the output port 607, through the XBAR fabric 610.

[0069] In accordance with an embodiment of the invention, using the IB protocol, the movement of the packets can be based on the availability of credits, a lack of which can block the packet movement in the VOQs in the IB network switch. Furthermore, the block behavior of the VOQs may result in unnecessary high (or even wasteful) utilization of the VOQs resources, depending on the traffic flow from a source (or input port) to a destination (or output port).

[0070] For example, if the link between the output port 607 and the remote traffic destination 608 becomes inactive (i.e. when the output port 607 is down), the release of the credits from the remote traffic destination 608 may stop as well (i.e. the current value of the available credits can be in any state). It is possible that there are no credits (or very few credits) available, in which case the packets that are enqueued in the VOQs 605 may not be able to move out of the VOQs 605, due to the lack of available credits.

[0071] As shown in Figure 6, when the link on output port 607 goes down, the interface 639 can be used to maintain the abstraction. The link state management 609 (state machine) on the output port 607 can advertise a new set of initial credits (e.g. link down credits 627), in the same (or similar) manner as the initial credits 628 that are advertised when the link is up.

[0072] In accordance with an embodiment of the invention, the system can ensure that the values, which are advertised for the link down credits 627, can be sufficiently large. For example, the values can be estimated based on the turnaround time at the output port 607. Then, the output port arbiter 606 can lock on to the link down credits 627 as the new maximum number.

[0073] With the new credits available, the VOQs 605 can start sending packets (or data) towards the output port 607. As the data moving towards the physical output port 607, the packets 623 can be dropped and the credits 624 can be returned to the output port arbiter 606. This ensures that the output port arbiter 606 can consistently have available credits, in order to prevent the blocking behavior (even when the output port is down).

[0074] Furthermore, when the link comes back up again, the credit flow follows the same process as advertising in the new initial credits 628, which allows the continuing traffic movement.

[0075] In accordance with an embodiment of the invention, the system can manage the flow of credits in order to avoid various deadlock scenarios under different conditions. For example, a deadlock can occur when the VOQs 605 are filled with packets for an output port, which may eventually cause a backup on the source VLs 621. Also, a deadlock may occur when multicast packets are involved. For example, when the ports that are ahead in the replication order list go down, the ports may start to block ports that are still active, since a multicast packet may not be able to gain forward progress as they get replicated one by one.

[0076] Thus, the system can avoid the blocking behavior (or even deadlocks) by draining the packets in the VOQs 605. Also, the system can provide non-blocking behavior between output ports that are active while other ports are going through transitions.

[0077] **Figure 7** shows an illustration of supporting credit management in a network switch, in accordance with an embodiment of the invention. As shown in Figure 7, a network device, such as an IB network switch 720, can be responsible for directing traffic from various remote senders, such as the traffic sources 701 and 711, to various remote receivers, such as the traffic destinations 708 and 718, in a high performance system 700.

[0078] Furthermore, the IB network switch 720, which is based on a crossbar (XBAR) fabric 710, can support a large number of ports (with multiple VLs), such as the input ports 702 and 712 and the output ports 707 and 717.

[0079] Each of the input ports 702 or 712 can advertise one or more credits to and receives one or more data packets from the traffic source 701 or 711. Each of the output port 707 or 717 can send one or more data packets to and receives one or more credits back from the traffic destination 708 or 718.

[0080] Additionally, each of the input ports 702 and 712 can take advantage of an input port packet classifier 703 or 713, which can determine one or more output ports for each incoming packet (and can use a port filter 704 or 714 to remove one or more packets). On per input port basis, the packets can be stored in an ingress buffer, such as the virtual output queues (VOQs) 705 or 715, before being transmitted to a traffic destination 708 or 718 (via the output port 707 or 717).

[0081] In accordance with an embodiment of the invention, a different output scheduler, such as the output port arbiters 706 and 716, can schedule the delivery of various packets from the different VOQs 705 and 715 toward the output port 707 and 717. Also, the system can manage the flow of credits in order to avoid various deadlock scenarios under different conditions.

[0082] As shown in Figure 7, the output port 707 or 717 can perform credit state management 729 or 739. Additionally, the system can provide an interface 730 or 740 on the output port 707 or 717 for indicating the maximum credit values to the output XBAR arbiter 706 or 716. When the initial credits 728 or 738 are presented to the arbiter 706 or 716, the arbiter 706 or 716 can lock the values of the initial credits 728 or 738 as the maximum credits that can be

consumed (until the next time when a new set of initial values are presented).

[0083] When the link is up (or active) with the traffic moving, all updates on the initial credits 728 or 738 can be presented to the output port arbiter 706 or 716 based on the values coming from the remote destination 708 or 718.

[0084] On the other hand, when the link goes down, the current value of the maximum credits allowed can be in any state. It is possible that there are no credits (or very few credits) available.

[0085] As shown in Figure 7, when the link on the output port 707 or 717 goes down, the interface 730 or 740 can be used to maintain the abstraction. The link state management 709 or 719 (state machine) on the output port 707 or 717 can advertise a new set of initial credits (e.g. the link down credits 727 or 737), in the same (or similar) manner as the initial credits 728 or 738 that are advertised when the link is up.

[0086] Then, the VOQs 705 and 715 can start sending packets (or data) towards the output ports 707 or 717. As the data moving towards the physical output port 707 or 717, the packets can be dropped and the credits can be returned to the output port arbiter 706 or 716. This ensures that the arbiter 706 or 716 can constantly have available credits, even when the output port is down, which prevents the blocking behavior.

[0087] Thus, by draining the packets, which are in the VOQ 705 and 715, the system can avoid the blocking behavior in the VOQs 705 and 715 and among other output ports (or even deadlocks).

[0088] Furthermore, when the link comes back up again, the credit flow can follow the same process as advertising the new initial credits 728 or 738, which allows the continuing traffic movement.

[0089] **Figure 8** illustrates an exemplary flow chart for supporting credit management in a network switch, in accordance with an embodiment of the invention. As shown in Figure 8, at step 801, the system can detect a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports. Furthermore, at step 802, the output port can provide one or more credits to an output scheduler. Then, at step 803, the output scheduler allows one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

[0090] In an embodiment of the invention, a method for supporting packet switching in a network environment, comprising: detecting a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports; providing, via the output port, one or more credits to an output scheduler; and allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

[0091] In another embodiment of the invention, the method wherein the networking device is

a network switch, which includes a crossbar fabric.

In another embodiment of the invention, the method wherein storing, via one or more input ports, said one or more packets into said one or more said virtual output queues.

[0092] In another embodiment of the invention, the method wherein allowing said one or more credits comprise one or more initial credits that are received from a remote traffic destination, when the output port is up.

[0093] In another embodiment of the invention, the method wherein said one or more credits comprise one or more link down credits, which is configured by the output port, when the output port is down.

[0094] In another embodiment of the invention, the method wherein setting, via the output scheduler, the one or more credits to be maximum credits available.

[0095] In another embodiment of the invention, the method wherein deriving, via the output scheduler, available credits based on the maximum credits available.

[0096] In another embodiment of the invention, the method further comprising the output scheduler selecting a destination virtual lane based on the available credits.

[0097] In another embodiment of the invention, the method further comprising forwarding said one or more packets from the virtual output queue toward the destination virtual lane on the output port.

[0098] In another embodiment of the invention, the method further comprising draining said one or more packets at the output port, and returning one or more credits to one or more source virtual lanes on an input port.

[0099] In an embodiment of the invention, a system for supporting packet switching in a network environment in any of the above methods wherein the networking device is operable to: detect a link state change at an output port on the networking device; provide, via the output port, one or more credits to an output scheduler; and allow, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

[00100] In another embodiment of the invention, the system for supporting packet switching in a network environment, comprising: a networking device with a plurality of input ports and a plurality of output ports, wherein the networking device is operable to: detect a link state change at an output port on the networking device; provide, via the output port, one or more credits to an output scheduler; and allow, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

[00101] In another embodiment of the invention, the system wherein the networking device is a network switch with a crossbar fabric.

In another embodiment of the invention, the system wherein one or more input ports are operable to store said one or more packets into said one or more said virtual output queues.

[00102] In another embodiment of the invention, the system wherein said one or more credits are one or more initial credits that are received from a remote traffic destination, when the output port is up.

[00103] In another embodiment of the invention, the system wherein said one or more credits are one or more link down credits, when the output port is down.

[00104] In another embodiment of the invention, the system wherein the output scheduler is operable to set the one or more credits to be maximum credits available.

[00105] In another embodiment of the invention, the system wherein the output scheduler is operable to derive available credits based on the maximum credits available.

[00106] In another embodiment of the invention, the system wherein the output scheduler is operable to select a destination virtual lane based on the available credits.

[00107] In another embodiment of the invention, the system wherein the networking device is operable to: forward said one or more packets from the virtual output queue toward the destination virtual lane on the output port, drain said one or more packets at the output port, and return one or more credits to one or more source virtual lanes on an input port.

[00108] In an embodiment of the invention, a non-transitory machine readable storage medium having instructions stored thereon that when executed cause a system to perform the steps comprising: detecting a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports; providing, via the output port, one or more credits to an output scheduler; and allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

[00109] In an embodiment of the invention, a computer program comprising instructions in machine-readable format that when executed cause a system to perform the above methods.

[00110] Many features of the present invention can be performed in, using, or with the assistance of hardware, software, firmware, or combinations thereof. Consequently, features of the present invention may be implemented using a processing system (e.g., including one or more processors).

[00111] Features of the present invention can be implemented in, using, or with the assistance of a computer program product which is a storage medium (media) or computer readable medium (media) having instructions stored thereon/in which can be used to program a processing system to perform any of the features presented herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[00112] Stored on any one of the machine readable medium (media), features of the present

invention can be incorporated in software and/or firmware for controlling the hardware of a processing system, and for enabling a processing system to interact with other mechanism utilizing the results of the present invention. Such software or firmware may include, but is not limited to, application code, device drivers, operating systems and execution environments/containers.

[00113] Features of the invention may also be implemented in hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art.

[00114] Additionally, the present invention may be conveniently implemented using one or more conventional general purpose or specialized digital computer, computing device, machine, or microprocessor, including one or more processors, memory and/or computer readable storage media programmed according to the teachings of the present disclosure. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

[00115] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention.

[00116] The present invention has been described above with the aid of functional building blocks illustrating the performance of specified functions and relationships thereof. The boundaries of these functional building blocks have often been arbitrarily defined herein for the convenience of the description. Alternate boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Any such alternate boundaries are thus within the scope and spirit of the invention.

[00117] The foregoing description of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. The breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments. Many modifications and variations will be apparent to the practitioner skilled in the art. The modifications and variations include any relevant combination of the disclosed features. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

Claims:

What is claimed is:

1. A method for supporting packet switching in a network environment, comprising:

providing a networking device, which is associated with a plurality of input ports and a plurality of output ports;

detecting a link state change at an output port that is associated with the networking device; and

notifying one or more of the input ports, via the output port, of the link state change at the output port.

2. The method according to Claim 1, wherein the networking device is a network switch which includes a crossbar fabric.

3. The method according to Claim 1 or 2, further comprising:

using a plurality of virtual output queues to store one or more packets that are received at the plurality of input ports, wherein each said input port is associated with one or more of said virtual output queues.

4. The method according to Claim 3, further comprising:

using an output scheduler to schedule an input port to forward one or more packets stored in a virtual output queue to an output port.

5. The method according to Claim 4, further comprising:

draining said one or more packets stored in the virtual output queue, and

returning one or more credits to one or more source virtual lanes associated with the input port.

6. The method according to Claim 4 or 5, further comprising:

sending, via the output port, a state change notification to an output scheduler, and

broadcasting, via the output scheduler, the state change notification to a plurality of input ports.

7. The method according to Claim 6, further comprising:

using the state change notification to configure an output port filter mask.

8. The method according to Claim 7, further comprising:

using, via an input port packet classifier, the output port filter mask to check one or more packets before enqueuing said one or more packets into the virtual output queue.

9. The method according to any preceding Claim, further comprising:

preventing, via one of said input ports, one or more packets targeting the output port from enqueueing into a virtual output queue.

10. The method according to Claim 9, further comprising:

dropping said one or more packets targeting the output port; and

returning one or more credits to one or more source virtual lanes associated with the input port.

11. A system for supporting packet switching in a network environment, comprising:

a networking device, which is associated with a plurality of input ports and a plurality of output ports, wherein the networking device is operable to:

detect a link state change at an output port that is associated with the networking device; and

notify one or more of the input ports, via the output port, of the link state change at the output port.

12. The system according to Claim 11, wherein:

the networking device is a network switch, which includes a crossbar fabric.

13. The system according to Claim 11 or 12, wherein:

the networking device is operable to use a plurality of virtual output queues to store one or more packets that are received at the plurality of input ports, wherein each said input port is associated with one or more of said virtual output queues.

14. The system according to Claim 13, wherein:

the networking device is operable to use an output scheduler to schedule an input port to forward one or more packets stored in a virtual output queue to the output port.

15. The system according to Claim 14, wherein the networking device is operable to:

drain said one or more packets stored in the virtual output queue, and

return one or more credits to one or more source virtual lanes associated with the input port.

16. The system according to Claim 14 or 15, wherein the networking device is operable to:

send, via the output port, a state change notification to an output scheduler, and

broadcast, via the output scheduler, the state change notification to a plurality of input ports.

17. The system according to Claim 16, wherein:

said input ports are operable to use the state change notification to configure an output port filter mask.

18. The system according to Claim 17, wherein:

an input port packet classifier is operable to use the output port filter mask to check one or more packets before enqueuing said one or more packets into the virtual output queue.

19. The system according to any of Claims 11 to 18, wherein the networking device is operable to:

prevent one or more packets targeting the output port from enqueueing into a virtual output queue,

and/or to drop said one or more packets targeting the output port, and return one or more credits to one or more source virtual lanes associated with the input port.

20. A non-transitory machine readable storage medium having instructions stored thereon that when executed cause a system to perform the steps comprising:

providing a networking device, which is associated with a plurality of input ports and a plurality of output ports;

detecting a link state change at an output port associated with the networking device; and

notifying one or more input ports, via the output port, of the link state change at the output port.

21. A method for supporting packet switching in a network environment according to any of claims 1 to 10, further comprising:

providing, via the output port, one or more credits to an output scheduler; and

allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

22. A method for supporting packet switching in a network environment, comprising:

detecting a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports;

providing, via the output port, one or more credits to an output scheduler; and

allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

23. The method according to Claim 22, wherein the networking device is a network switch, which includes a crossbar fabric.

24. The method according to any of Claims 21 to 23, further comprising:

storing, via one or more input ports, said one or more packets into said one or more said virtual output queues.

25. The method according to any of Claims 21 to 24, wherein allowing said one or more credits comprise one or more initial credits that are received from a remote traffic destination, when the output port is up.

26. The method according to any of Claims 21 to 25, wherein said one or more credits comprise one or more link down credits, which is configured by the output port, when the output port is down.

27. The method according to any of Claims 21 to 26, further comprising:

setting, via the output scheduler, the one or more credits to be maximum credits available.

28. The method according to Claim 27, further comprising:

deriving, via the output scheduler, available credits based on the maximum credits available.

29. The method according to Claim 27 or 28, further comprising:

the output scheduler selecting a destination virtual lane based on the available credits.

30. The method according to Claim 29, further comprising:

forwarding said one or more packets from the virtual output queue toward the destination virtual lane on the output port.

31. The method according to Claim 30, further comprising:

draining said one or more packets at the output port, and

returning one or more credits to one or more source virtual lanes on an input port.

32. A system for supporting packet switching in a network environment according to any of claims 11 to 19, wherein the networking device is operable to:

detect a link state change at an output port on the networking device;

provide, via the output port, one or more credits to an output scheduler; and

allow, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

33. A system for supporting packet switching in a network environment, comprising:

a networking device with a plurality of input ports and a plurality of output ports, wherein the networking device is operable to:

detect a link state change at an output port on the networking device;

provide, via the output port, one or more credits to an output scheduler; and

allow, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

34. The system according to Claim 33, wherein:

the networking device is a network switch with a crossbar fabric.

35. The system according to any of Claims 32 to 34, wherein:

one or more input ports are operable to store said one or more packets into said one or more said virtual output queues.

36. The system according to any of Claims 32 to 35, wherein:

said one or more credits are one or more initial credits that are received from a remote traffic destination, when the output port is up.

37. The system according to any of Claims 32 to 36, wherein:

said one or more credits are one or more link down credits, when the output port is down.

38. The system according to any of Claims 32 to 37, wherein:

the output scheduler is operable to set the one or more credits to be maximum credits available.

39. The system according to Claim 38, wherein:

the output scheduler is operable to derive available credits based on the maximum credits available.

40. The system according to Claim 38 or 39, wherein:

the output scheduler is operable to select a destination virtual lane based on the available credits.

41. The system according to Claim 40, wherein the networking device is operable to:

forward said one or more packets from the virtual output queue toward the destination virtual lane on the output port,

drain said one or more packets at the output port, and

return one or more credits to one or more source virtual lanes on an input port.

42. A non-transitory machine readable storage medium having instructions stored thereon that when executed cause a system to perform the steps comprising:

detecting a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports;

providing, via the output port, one or more credits to an output scheduler; and

allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits.

43. A computer program comprising instructions in machine-readable format that when executed cause a system to perform the method of any of claims 1-10 or 21-31.

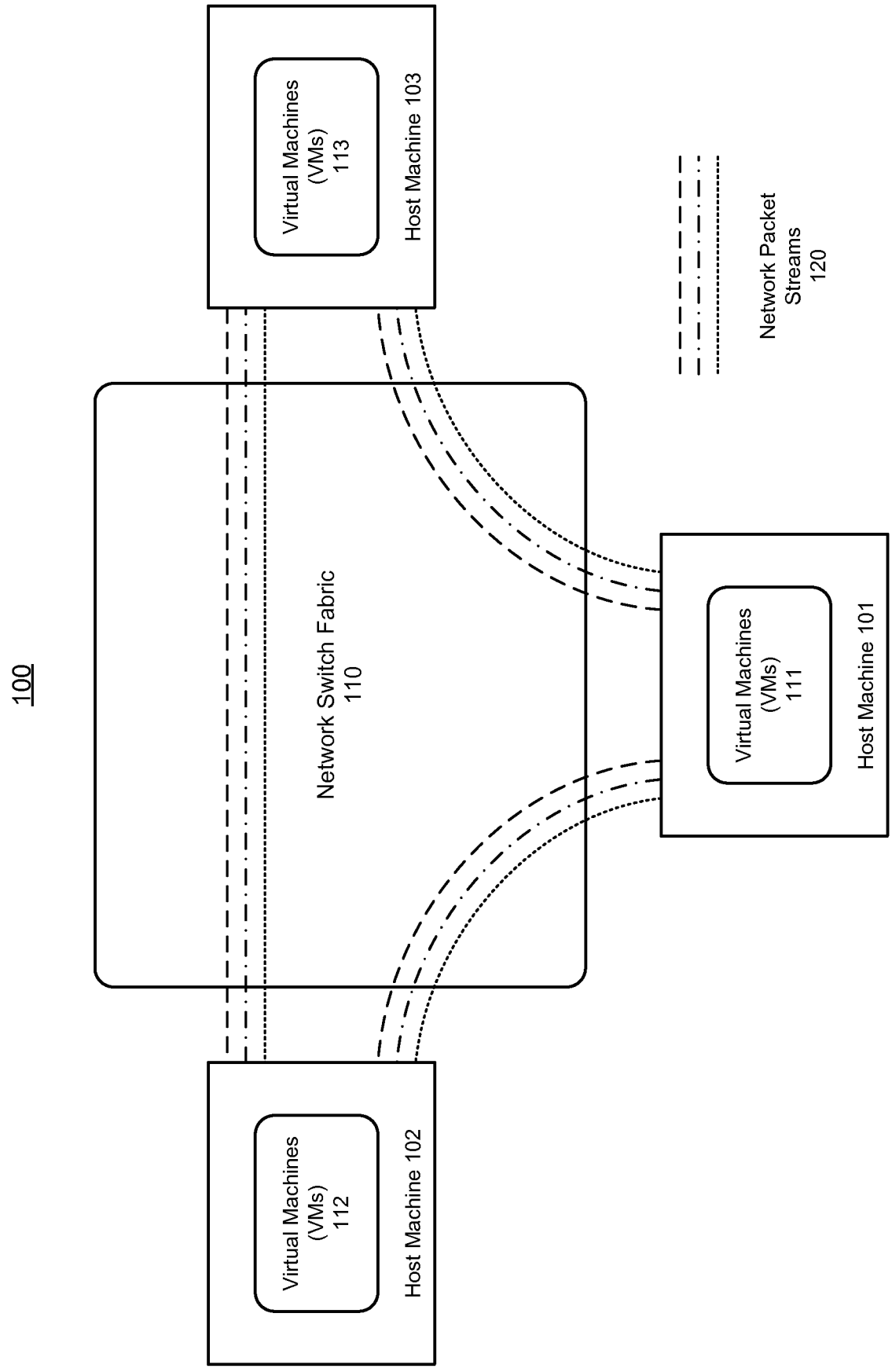
100
Network Switch Fabric
110
Host Machine 103
Virtual Machines
(VMs)
113
Host Machine 102
Virtual Machines
(VMs)
112
Host Machine 101
Virtual Machines
(VMs)
111
Network Packet
Streams
120

FIGURE 1

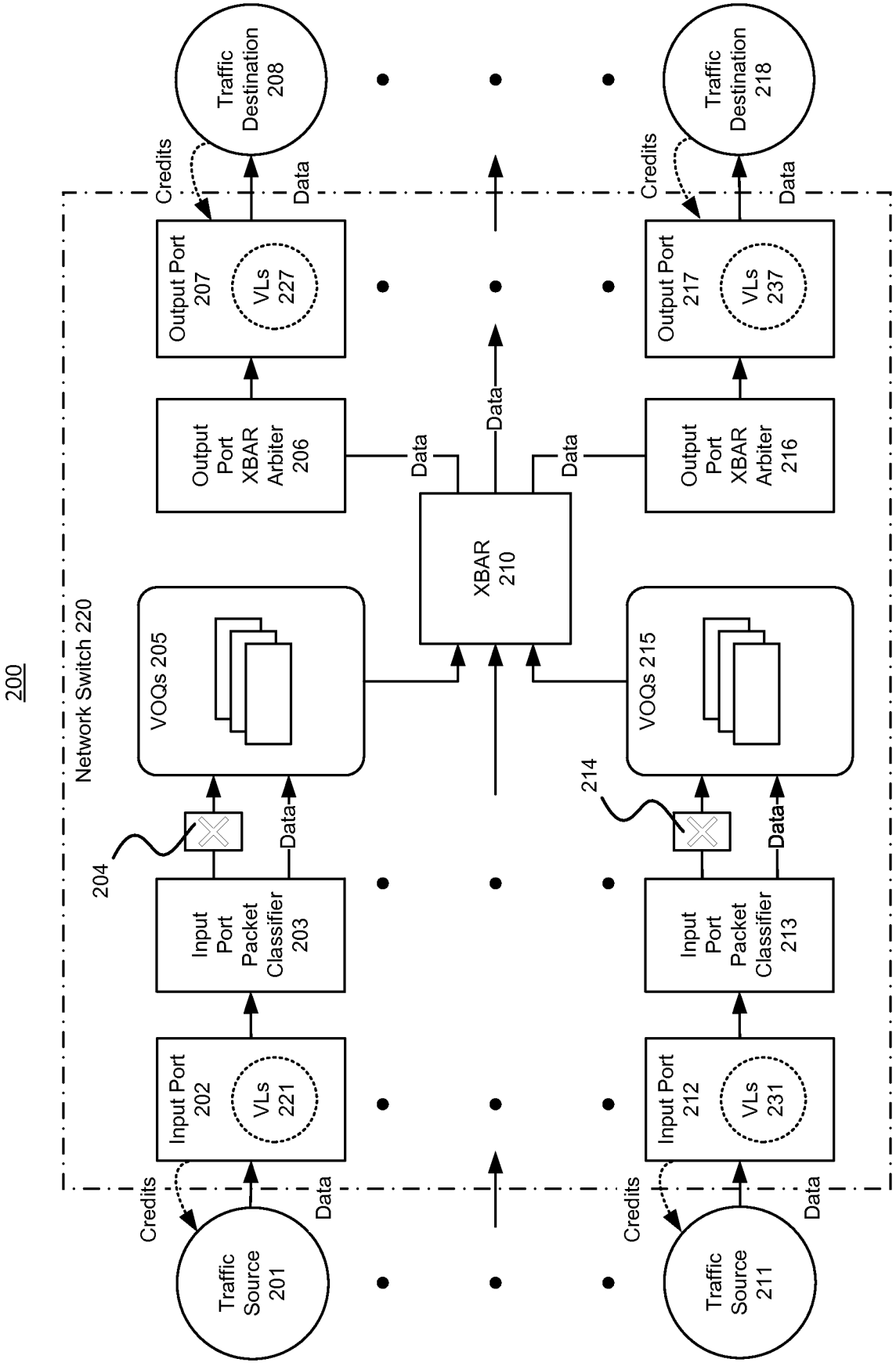
200
Network Switch 220
Traffic Source 201
Input Port 202
VLs 221
Input Port Packet Classifier 203
204
VOQs 205
Output Port XBAR Arbiter 206
Output Port 207
VLs 227
Traffic Destination 208
XBAR 210
Traffic Source 211
Input Port 212
VLs 231
Input Port Packet Classifier 213
214
VOQs 215
Output Port XBAR Arbiter 216
Output Port 217
VLs 237
Traffic Destination 218
Credits
Data

FIGURE 2

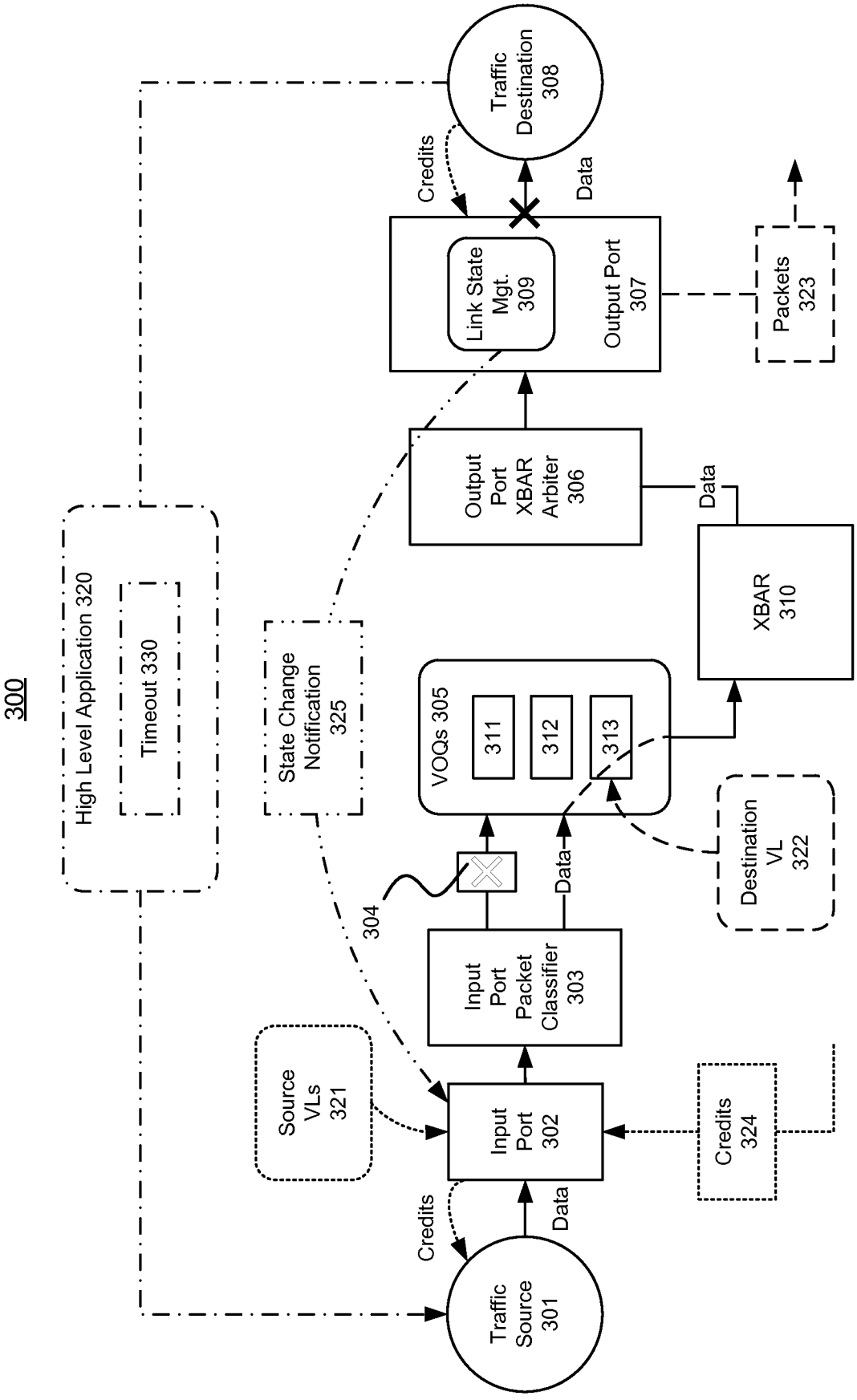
300
High Level Application 320
Timeout 330
Traffic Source 301
Traffic Destination 308
Credits
Data
Source VLs 321
Input Port 302
Input Port Packet Classifier 303
304
VOQs 305
311
312
313
State Change Notification 325
Destination VL 322
Credits 324
Output Port XBAR Arbiter 306
XBAR 310
Link State Mgt. 309
Output Port 307
Packets 323

FIGURE 3

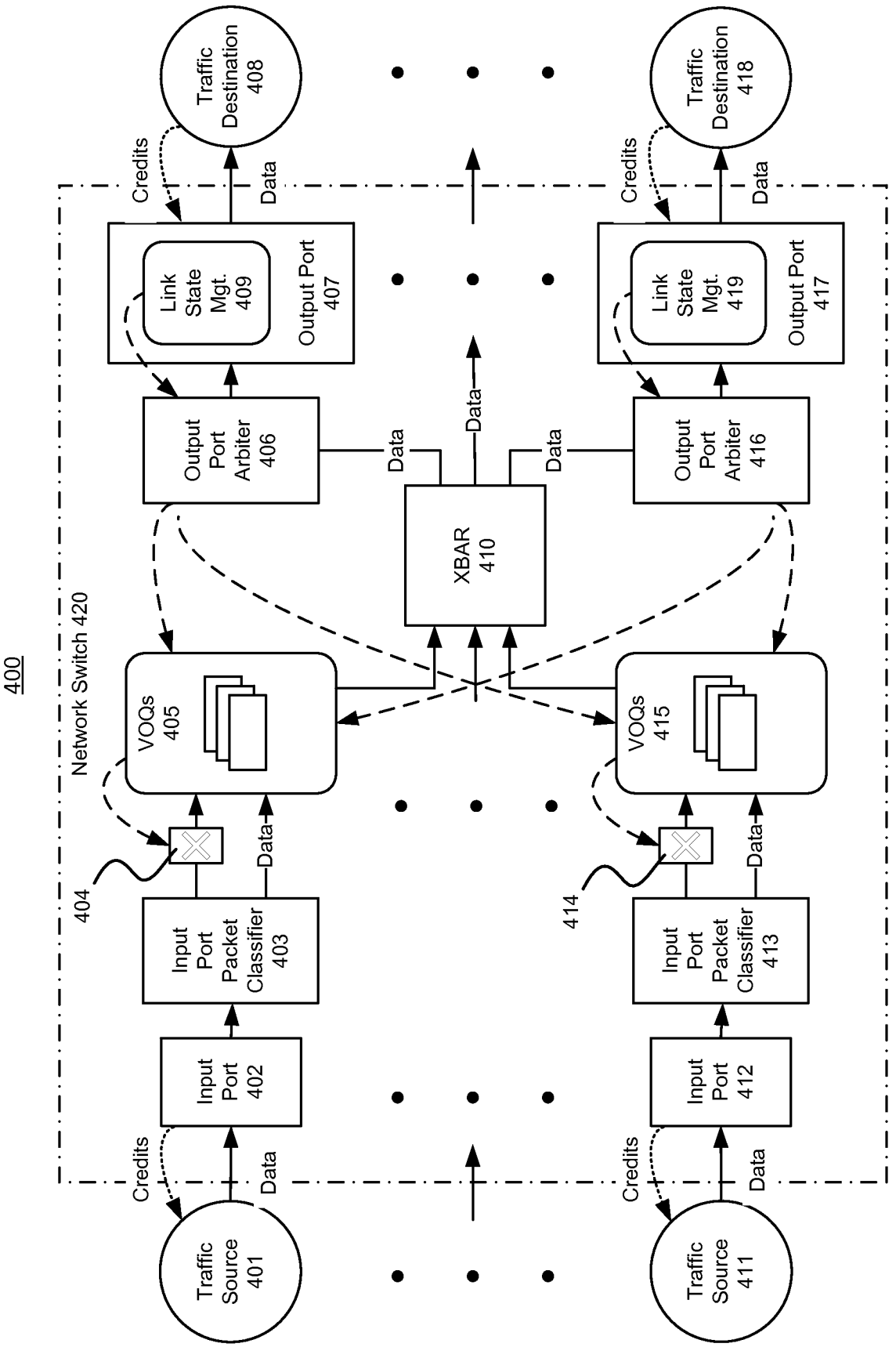
400
Network Switch 420
Traffic Source 401
Input Port 402
Input Port Packet Classifier 403
404
VOQs 405
Output Port Arbiter 406
Link State Mgt. 409
Output Port 407
Traffic Destination 408
Credits
Data
XBAR 410
Traffic Source 411
Input Port 412
Input Port Packet Classifier 413
414
VOQs 415
Output Port Arbiter 416
Link State Mgt. 419
Output Port 417
Traffic Destination 418

FIGURE 4

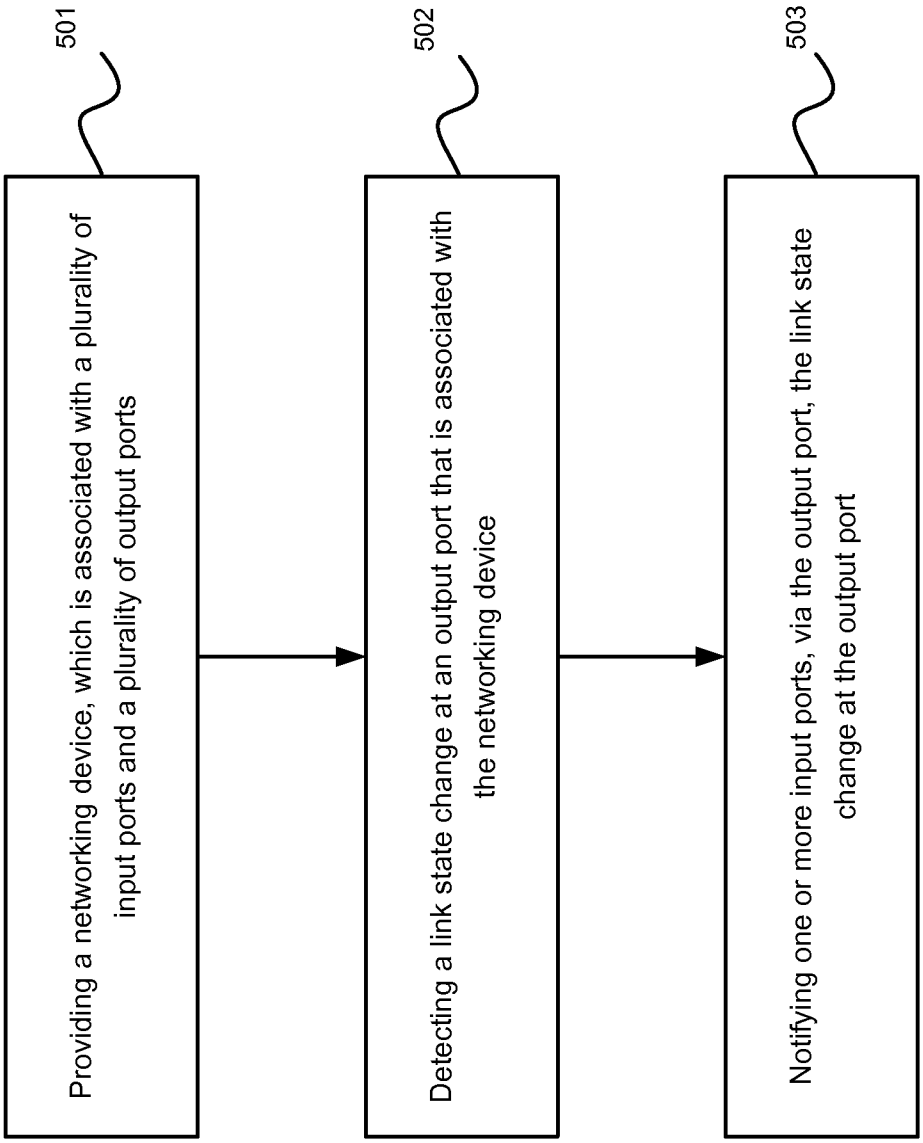
501
Providing a networking device, which is associated with a plurality of input ports and a plurality of output ports
502
Detecting a link state change at an output port that is associated with the networking device
503
Notifying one or more input ports, via the output port, the link state change at the output port

FIGURE 5

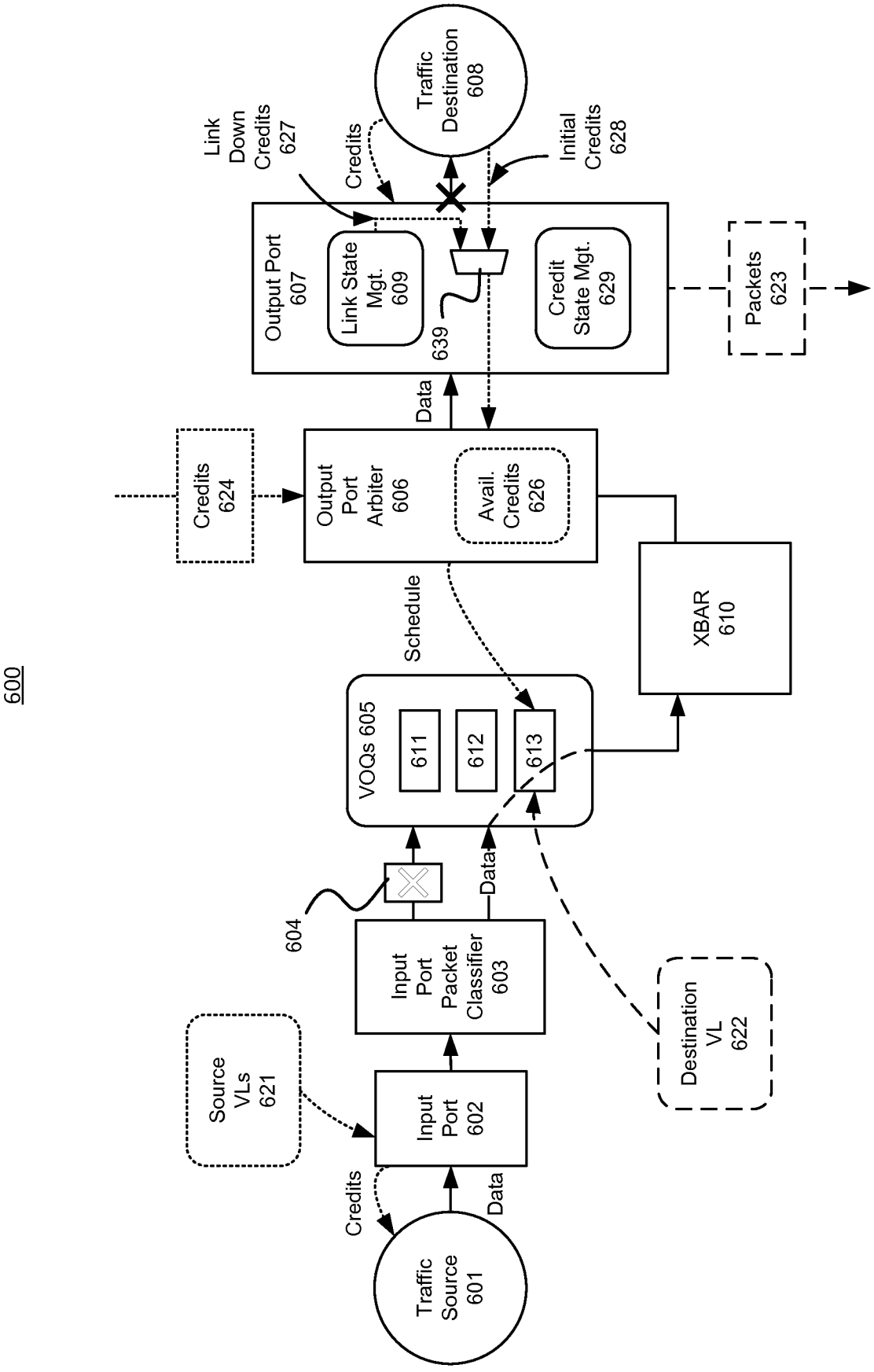
600
Traffic Source 601
Credits
Data
Input Port 602
Source VLs 621
Input Port Packet Classifier 603
604
Data
VOQs 605
611
612
613
Schedule
Output Port Arbiter 606
Credits 624
Avail. Credits 626
Data
Output Port 607
Link State Mgt. 609
639
Credit State Mgt. 629
Packets 623
Link Down Credits 627
Credits
Traffic Destination 608
Initial Credits 628
XBAR 610
Destination VL 622

FIGURE 6

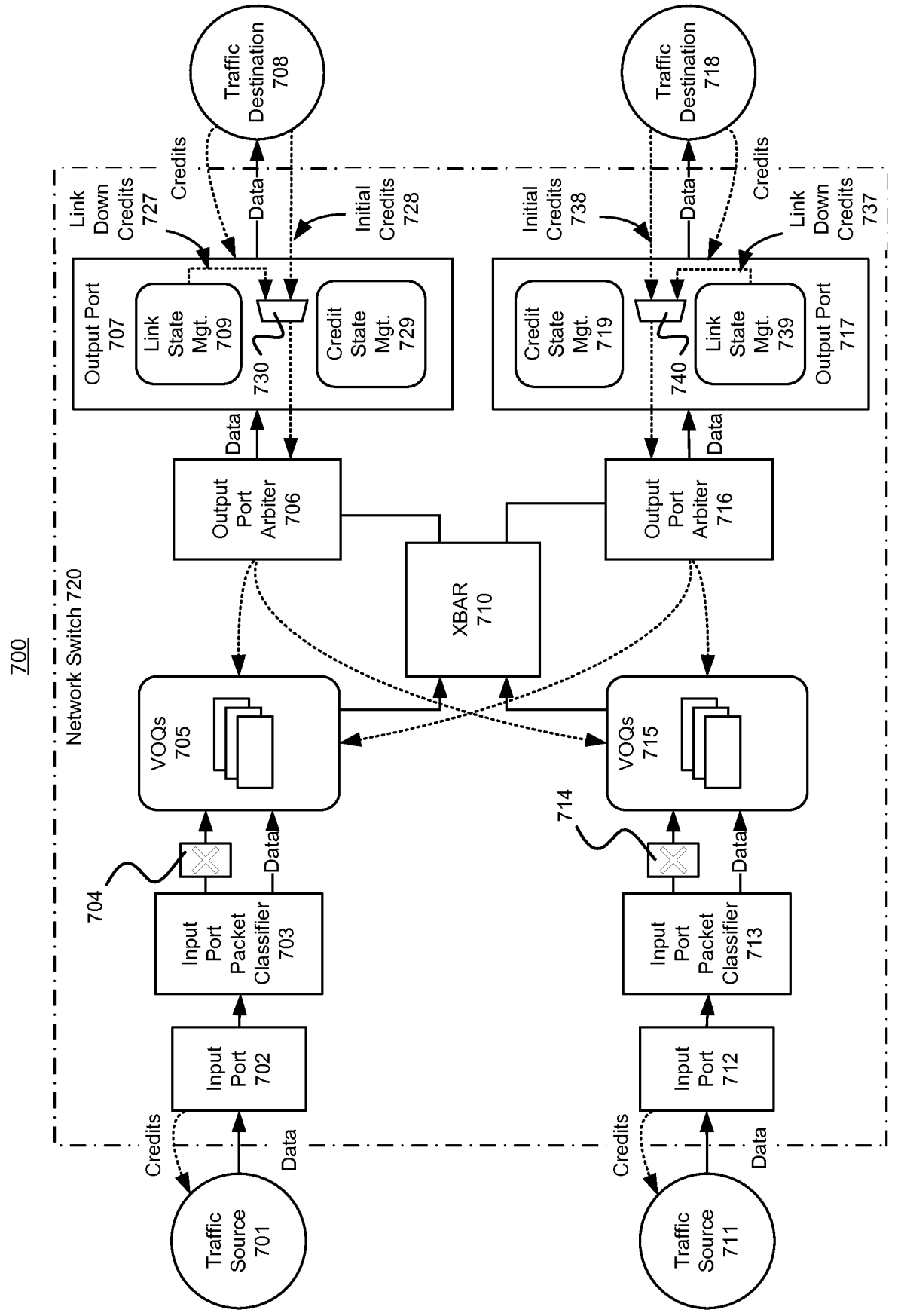
700
Network Switch 720
Traffic Source 701
Input Port 702
Input Port Packet Classifier 703
704
VOQs 705
Output Port Arbiter 706
Output Port 707
Link State Mgt. 709
730
Credit State Mgt. 729
Link Down Credits 727
Initial Credits 728
Credits
Data
Traffic Destination 708
XBAR 710
Traffic Source 711
Input Port 712
Input Port Packet Classifier 713
714
VOQs 715
Output Port Arbiter 716
Output Port 717
Credit State Mgt. 719
740
Link State Mgt. 739
Initial Credits 738
Link Down Credits 737
Traffic Destination 718

FIGURE 7

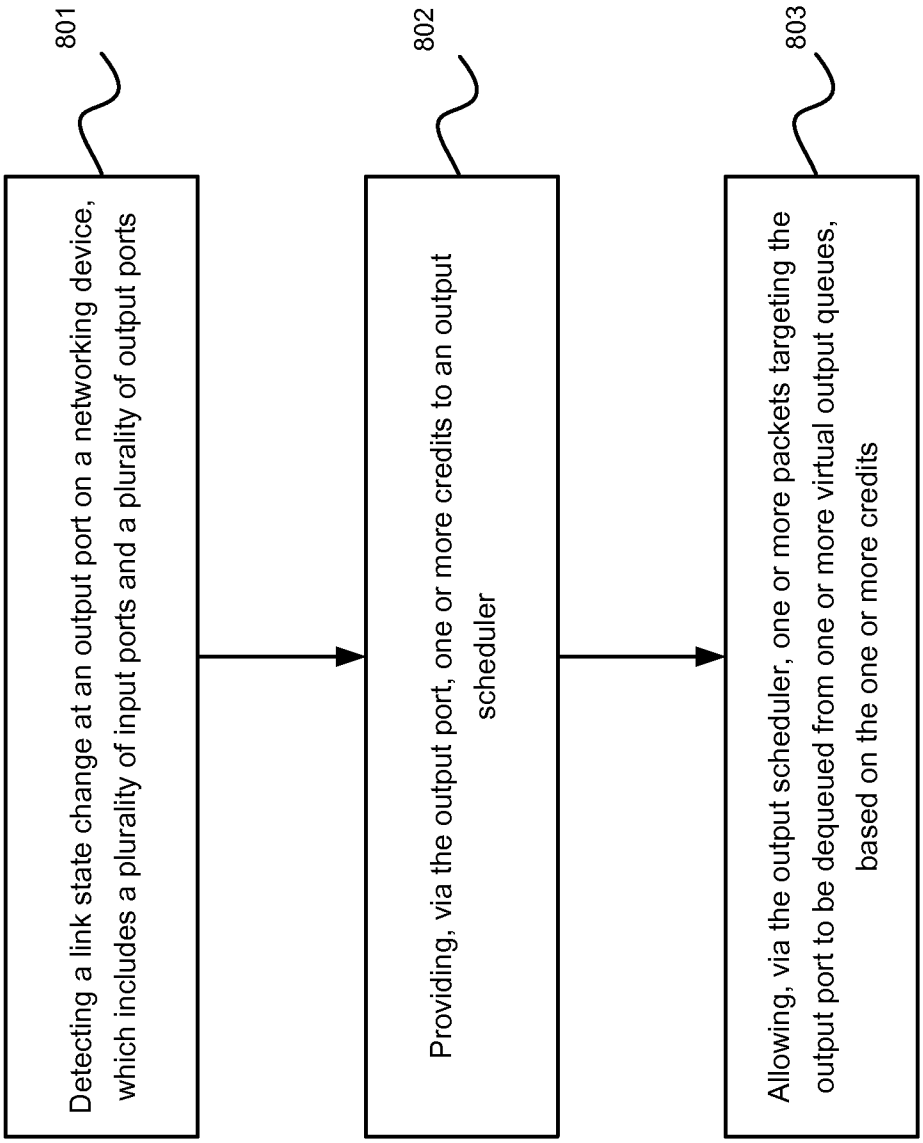
801
Detecting a link state change at an output port on a networking device, which includes a plurality of input ports and a plurality of output ports
802
Providing, via the output port, one or more credits to an output scheduler
803
Allowing, via the output scheduler, one or more packets targeting the output port to be dequeued from one or more virtual output queues, based on the one or more credits

FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/063520

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/26 H04L12/825 H04L12/935 H04L12/931
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/088969 A1 (CARLSEN SCOTT [US] ET AL) 28 April 2005 (2005-04-28) the whole document	1-43
A	US 876 775 A (CRITTENDEN IMMER O [US]) 14 January 1908 (1908-01-14) the whole document	1-43

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 March 2016	21/03/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	García Bolós, Ruth

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2015/063520

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005088969 A1	28-04-2005	AU 2002366842 A1	09-07-2003
		CA 2470758 A1	03-07-2003
		EP 1466449 A1	13-10-2004
		US 2003112818 A1	19-06-2003
		US 2005088969 A1	28-04-2005
		US 2005088970 A1	28-04-2005
		US 2010265821 A1	21-10-2010
		WO 03055157 A1	03-07-2003
US 876775 A	14-01-1908	NONE	