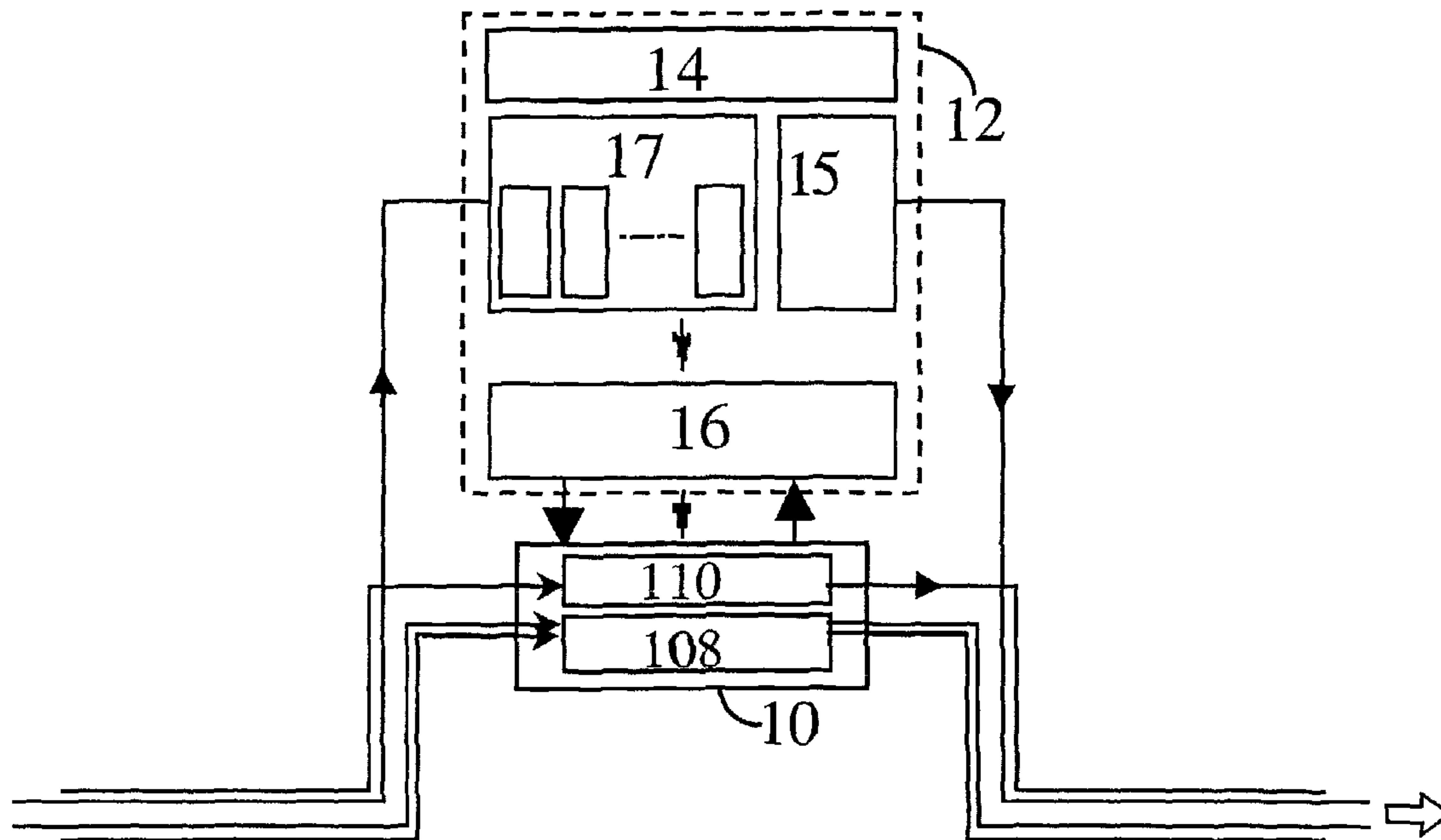




(86) Date de dépôt PCT/PCT Filing Date: 2002/11/15
 (87) Date publication PCT/PCT Publication Date: 2003/05/30
 (85) Entrée phase nationale/National Entry: 2004/05/14
 (86) N° demande PCT/PCT Application No.: AU 2002/001559
 (87) N° publication PCT/PCT Publication No.: 2003/045016
 (30) Priorité/Priority: 2001/11/16 (PR 8932/01) AU

(51) Cl.Int.⁷/Int.Cl.⁷ H04L 12/56, H04L 12/46, H04L 12/28,
H04L 29/08, H04L 29/06
 (71) Demandeurs/Applicants:
TELSTRA NEW WAVE PTY LTD, AU;
TELSTRA CORPORATION LIMITED, AU;
UNIVERSITY OF WOLLONGONG, AU
 (72) Inventeurs/Inventors:
BOUSTEAD, PAUL ANDREW, AU;
SAFAEI, FARZAD, AU;
RATCLIFF, BRUCE RAYMOND, AU
 (74) Agent: SMART & BIGGAR

(54) Titre : RESEAUX ACTIFS
 (54) Title: ACTIVE NETWORKS



(57) **Abrégé/Abstract:**

An active node of a communications network includes a forwarding component for forwarding a packet of an active application, a processor for processing the packet, and a forwarding control component for modifying path data associated with the packet, based on the processing. The forwarding component can forward said packet directly to the network or via the processor. The processing can include executing code contained within the packet, and/or executing code to process application data in the packet. The path data can include the MPLS label stack of the packet and/or MPLS tables of the forwarding component. To allow processing based on multiple labels of a packet, the forwarding component can include a loopback connection between its input and output ports.

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

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
30 May 2003 (30.05.2003)

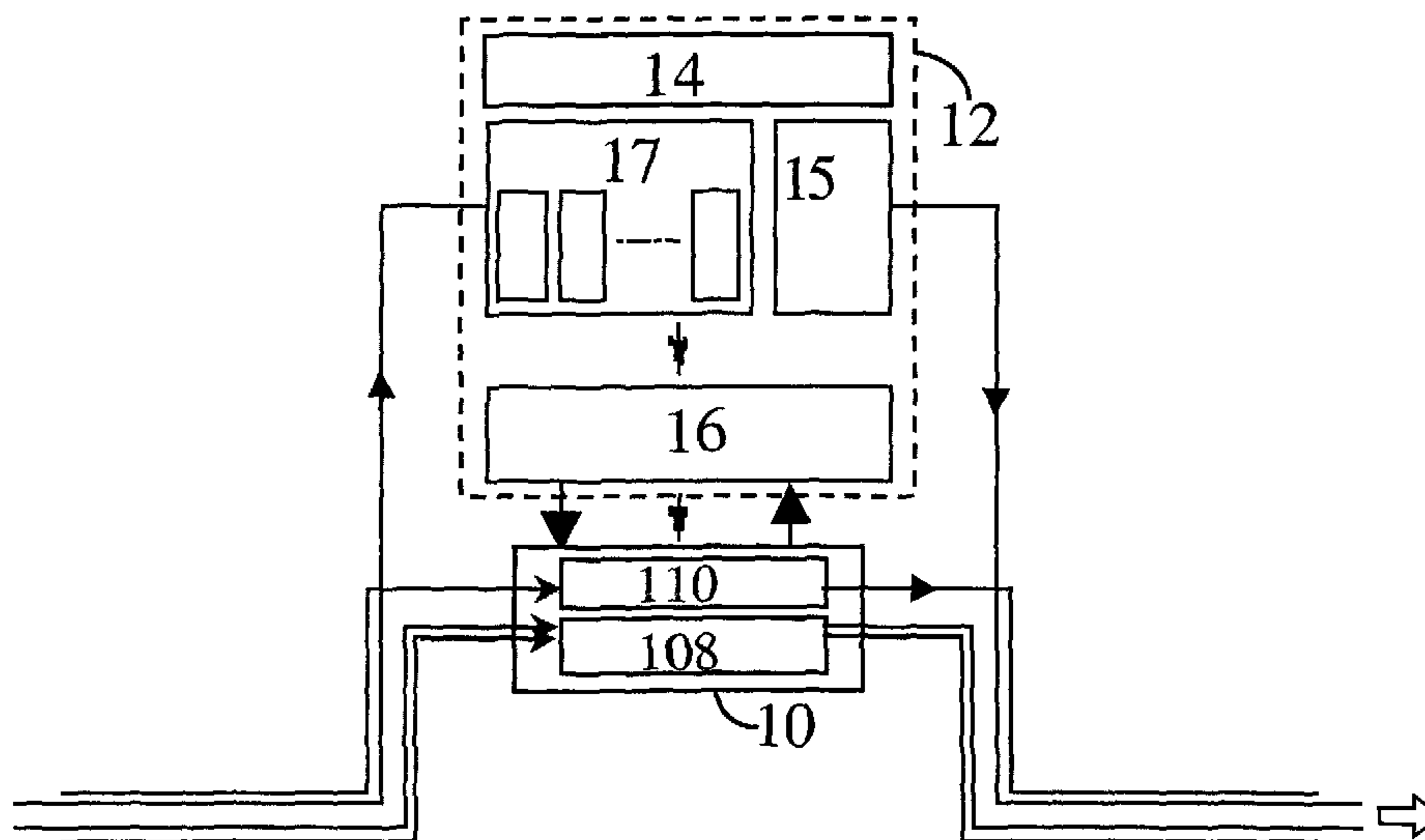
PCT

(10) International Publication Number
WO 03/045016 A1

- (51) International Patent Classification⁷: H04L 12/56, 12/28, 12/46, 29/06, 29/08
- (21) International Application Number: PCT/AU02/01559
- (22) International Filing Date:
15 November 2002 (15.11.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
PR 8932/01 16 November 2001 (16.11.2001) AU
- (71) Applicants (for all designated States except US): **TELSTRA NEW WAVE PTY LTD** [AU/AU]; 242 Exhibition Street, MELBOURNE, Victoria 3000 (AU). **TELSTRA CORPORATION LIMITED** [AU/AU]; 242 Exhibition Street, MELBOURNE, Victoria 3000 (AU). **UNIVERSITY OF WOLLONGONG** [AU/AU]; Wollongong, New South Wales 2522 (AU).
- (72) Inventors; and
(75) Inventors/Applicants (for US only): **BOUSTEAD, Paul, Andrew**, [AU/AU]; 4/19 Sperry Street, WOLLONGONG, New South Wales 2500 (AU). **SAFAEI, Farzad** [AU/AU]; 8 Ambleside Avenue, MT KEIRA, New South Wales 2500 (AU). **RATCLIFF, Bruce, Raymond**, [AU/AU]; 58 Westlands Road, EMERALD, Victoria 3782 (AU).
- (74) Agents: **DAVIES COLLISON CAVE**; WEBBER, David, Brian, 1 Little Collins Street, Melbourne, Victoria 3000 et al. (AU).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: ACTIVE NETWORKS



(57) **Abstract:** An active node of a communications network includes a forwarding component for forwarding a packet of an active application, a processor for processing the packet, and a forwarding control component for modifying path data associated with the packet, based on the processing. The forwarding component can forward said packet directly to the network or via the processor. The processing can include executing code contained within the packet, and/or executing code to process application data in the packet. The path data can include the MPLS label stack of the packet and/or MPLS tables of the forwarding component. To allow processing based on multiple labels of a packet, the forwarding component can include a loopback connection between its input and output ports.



WO 03/045016 A1

WO 03/045016 A1



Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,
ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK,
TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

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.

Published:

— *with international search report*

- 1 -

ACTIVE NETWORKS

FIELD OF THE INVENTION

The present invention relates to active communications networks, and in particular to an
5 active node and a process for processing packets in an active network.

BACKGROUND

Active or programmable networks are packet-switched networks in which packets sent
from a source node to a destination node can contain code that is executed by one or more
10 intermediate nodes. Active networks have been developed not so much to provide a
distributed processing environment as to provide a customisable network to meet the needs
of its users. For example, active networks can provide quality of service (QoS) and other
network management functions, virtual networking, and enhanced data services to their
users. In contrast, non-active networks are not generally controllable by their users.

15

The use of active network nodes in a telecommunications carrier grade network allows
executable code to be injected into the network to enable the provision of new services.
Practical deployment of active network components generally requires the data path to be
disturbed to a minimum. Packets in an active network can be considered to include active
20 packets that require processing within the network, and non-active packets that are simply
forwarded to their destination on a fast pathway. Active packets can contain code for
execution in the network, references to executable code, and/or data to be processed by an
active application within the network. Scalability is an issue for active networks, and active
applications with a majority of active packets are not generally scalable. However,
25 applications with a relatively small fraction of active packets and a majority of non-active
packets are more likely to scale. There is a need for an active node that allows scalable
extraction of a subset of active packets from a stream of active and non-active packets at
line rate to allow the implementation of new services and a process for processing packets

- 2 -

of an active application that alleviate one or more difficulties of the prior art, or at least provide a useful alternative to existing active nodes and packet processes.

SUMMARY OF THE INVENTION

5 In accordance with the present invention there is provided an active node for a communications network, including:

a forwarding component for forwarding a packet of an active application;

a processor for processing said packet; and

10 a forwarding control component for modifying path data associated with said packet based on said processing.

The present invention also provides a process for processing packets of an active application in a communications network, including:

15 processing code of an active packet of said application to determine a path for said packets;

storing path data representing said path in said packets; and

forwarding said packets on said path.

20 The present invention also provides a process for processing an active packet in a communications network, including:

receiving an active packet including path data ;

processing said active packet;

modifying said path data in accordance with said processing; and

forwarding said packet in accordance with the modified path data.

25

The present invention also provides a process for processing packets of an active application in a communications network, including:

30 selecting, for one or more packets of said application, one or more paths to respective processors based on requirements of said one or more packets and capabilities of said processors;

- 3 -

storing path data representing said selected one or more paths in said one or more packets; and

forwarding each of said one or more packets in accordance with said path data of the packet.

5

The present invention also provides a process for processing packets of an active application in a communications network, including:

determining requirements of said application from the first at least one packet of said application;

10 selecting one or more paths to respective processors based on said requirements and capabilities of said processors ;

storing path data representing the selected one or more paths in each packet; and forwarding said packets in accordance with said path data.

15 The present invention also provides a process for processing an active packet in a communications network, including:

receiving an active packet including first path data and second path data;

removing said first path data from said packet; and

forwarding said packet in accordance with said second path data of said packet.

20

The present invention also provides an active node for a communications network having:

a processor for processing active packets; and

a forwarding component for determining, on the basis of at least one label of packets received from the network, to forward said packets to said processor or output said

25 packets to said network.

The present invention also provides an active node for executing the steps of any one of the above methods.

30

- 4 -

The present invention also provides software modules having code for executing the steps of any one of the above methods.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a schematic diagram of a preferred embodiment of an active node of a communications network;

10 Figures 2A to 2C are schematic diagrams illustrating paths taken by packets of an application in a network including active nodes;

Figure 3 is a schematic diagram illustrating paths taken by packets of an application through three active nodes;

Figures 4A to 4E are schematic diagrams illustrating the processing of five packets with different MPLS stack contents in the first active node of Figure 3;

15 Figure 5 is a schematic diagram of a carrier grade active network topology defined by active nodes implemented by adding active processors and content switches to existing MPLS switches in a legacy carrier grade network;

Figure 6 is a schematic diagram of an alternative embodiment of an active node of a communications network; and

20 Figures 7 and 8 are flow diagrams of a packet process executed by the active node.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An active node of a communications network executes a packet process that processes an active application packet and forwards the packet and subsequent packets of the application to other active nodes or non-active nodes of the network on the basis of code contained in the active application packet. As shown in Figure 1, the active node includes a high-speed forwarding component or fast path component 10 and a processing component 12. The processing component 12 includes an active processing engine 14, control modules 15, application modules 17, and a forwarding control module 16.

30

- 5 -

The processing component 12 is connected to output and input ports of the fast path component 10 so that the processing component 12 can function as an adjunct processor to the fast path component 10. Specifically, a first output port of the fast path component 10 is connected to an input port of the processing component 12. An output port of the processing component 12 is fed back to an input port of the fast path component 10. Using the forwarding control module 16, the processing component 12 also communicates with the fast path component 10 using a separate connection to a management port of the fast path component 10.

10 The fast path component 10 receives packets from the network and forwards them to an appropriate destination. Some of these packets are passed to the processing component 12 for processing by the active processing engine 14. The active processing engine 14 includes one or more microprocessors for executing active application code in active packets of an active application. The forwarding control module 16 is used by an active application to change the forwarding destination of non-active packets belonging to the same application and forwarded on the fast path component 10. The fast path component 10 is described as rigid if application-specific integrated circuits (ASICs) are used to forward packets. If network processors including software modules are used to process packets, then the fast path component 10 has a rigid component 108 defined by the functions implemented in hardware, and a semi-flexible component 110 that is defined by the network processor software. The semi-flexible component 110 enhances the flexibility of active fast path forwarding, particularly for quality-of-service (QoS) support.

In the described embodiment, the fast path component 10 is a high-speed packet switch such as a Cisco® or Nortel® multi-protocol label switching (MPLS) switch. The processing component 12 is a standard computer system such as an Intel® x86-based personal computer running a Linux® operating system. The packet process executed by the active node is implemented as software modules, being the control, application, and forwarding control modules 15 to 17 of the active node. However, it will be apparent to the skilled addressee that at least some of the steps of the packet process may be alternatively implemented by dedicated hardware components such as application-specific integrated

- 6 -

circuits (ASICs). It will also be apparent that the fast path component 10 and the processing component 12, described herein as separate but linked entities, can alternatively be combined to form a single, integrated active node device.

5 The active node uses a protocol known as multi-protocol label switching (MPLS) for packet routing, as described in RFC 3031. MPLS allows multiple labels to be included in each packet in the form of a label stack. The label stack is placed in a shim header that is inserted between the layer 2 and layer 3 packet headers. Permissible label stack operations include push, pop and forward. Labels can be pushed onto the top of the stack and labels
10 can be popped off the top of the stack. The label at the top of the stack is used for forwarding the packet on a particular label-switched path (LSP), determined by using the label as an index into an MPLS forwarding table. MPLS label stacks have been used for traffic engineering, and for creating overlay networks such as virtual private networks (VPNs). Label stacks enable multiple levels of tunnelling, and this property makes them
15 useful for active packet forwarding.

The active node uses MPLS tunnels for active packet forwarding by allowing active applications to modify MPLS forwarding tables stored in the fast path component 10 and by allowing MPLS label push and pop operations to be performed on packets within the
20 fast path component 10 and within the processing component 12. Moreover, the processing component 12 can be partitioned into multiple virtual processing components using virtual server software such as Ensim Private Server, described at <http://www.ensim.com> or VMware, described at <http://www.vmware.com>, providing secure execution environments for users of the network. Accordingly, an LSP can also specify a particular virtual
25 processing component or execution environment within an active node of the network. Finally, an LSP can also identify a particular application to be used to process an active packet. The use of labels improves the efficiency of active networking by specifying one or more fast paths to destinations where active processing is to be performed. The expression 'fast path' refers to a path through one or more fast path components 10 of one
30 or more active nodes without passing through any processing components 12 of those nodes.

- 7 -

Prior art active networks have used various filtering or tagging methods that allow each active node receiving a packet to determine whether it needs to process the packet or simply forward it on the network. However, as shown in Figure 2A, this requires every
5 packet 200 to be received and processed by every active node 206 to 210. This is an inefficient use of resources, wasting processing cycles and decreasing throughput. In contrast, the use of a label stack allows packets 200 to be directed to particular active nodes for processing. For example, all the packets 200 of an application can be directed to the processing component 12 of a particular active node 208, as shown in Figure 2B.
10 Alternatively, the packets 200 can be divided into two flows or label-switched paths 202, 204, whereby only the packets that require processing are directed on a first path 202 to be processed by a particular active node 208, with the remainder sent on a fast path 204 bypassing the processing components 12 of all active nodes 206 to 210, as shown in Figure 2C. The determination of whether a packet requires processing or not can be
15 performed by a content switch at the edge of the MPLS network, as described below.

A communications network includes three active nodes 302 to 306, as shown in Figure 3. Six label switched paths (LSPs) A to E and Z are defined in MPLS forwarding tables of each node 302 to 306. Packets on LSP A are switched through all active nodes 302 to 306
20 on a fast path through the fast path component 10 of each active node without passing through any processing components 12 of the nodes 302 to 306. LSPs C0 to C3 connect the active processing engine 14 of each active node with the active processing engine 14 of the next active node in the path. LSP B connects the active processor 14 of the first node 302 to the active processor 14 of the third node 306. LSP Z represents an alternate path
25 from the first node 302 to the third node 306.

The network supports applications that implement a Programmable Virtual Network (PVN). A PVN allows a telecommunications carrier to provide a virtual network topology to its customers, allowing them to implement their own routing, billing, network
30 management protocols and other customized protocols such as multicast. In order to implement a PVN, it is necessary to identify which packets belong to the PVN so that

- 8 -

custom routing and forwarding mechanisms can be used. It is also necessary to determine which packets are active and are processed by active applications of the application modules 17 for mechanisms such as custom routing protocols.

5 A particular PVN application uses active packets to monitor link failures in a PVN. LSP A follows the path of an established PVN link, and LSPs C0 to C3 are used for active packets whose content is processed by each active node 302 to 306. As in standard MPLS networks, labels are inserted into the headers of packets by content switches (not shown, but typically a level 4–7 network switch such as a Nortel Alteon® web switching module)
10 at the edge of the MPLS network, based on the content of each packet. For example, a content switch receives five packets of the PVN link application. Packet 1 is a non-active data packet in the PVN and only requires fast path forwarding. Accordingly, label A is pushed on the top of the packet's label stack and the packet is forwarded to the first node 302. Packets 2 and 3 are active packets that are to be processed by the first node 302
15 and so have label C0 at the top of their stacks, followed by label A.

The active node executes a packet process, as shown in Figure 7, that either forwards received packets directly on the fast path component 10, or via the processing component 12. Figures 4A to 4D illustrate the processing of the five packets of the PVN link
20 application by the first node 302 of the three active nodes 302 to 306 shown in Figure 3.

The packet process begins at step 702 when the first node 302 receives an active packet. At step 704, the fast path component 10 examines the label on the top of the packet's label stack, and at step 706, the fast path component 10 performs a lookup operation on its MPLS label tables to determine the appropriate action(s). Packet 1 of Figure 4A is labelled
25 "A". Accordingly, the test at step 708 indicates that no processing of the packet is required, and the packet is therefore forwarded on the fast path component 10 of the first node 302 at step 716.

Packet 2 in Figure 4B has the label "C0" on the top of its stack and is therefore passed to
30 the processing component 12 and processed by the active processing engine 14 of the first node 302 at step 710. The packet can contain active application code, a reference to an

- 9 -

active application of the application modules 17, and/or application data to be processed. The result of the processing may determine that this packet will also require processing in the active processor 14 of the second node 304. If so, then the packet's label stack is modified at step 712 by popping "C0" off the label stack and pushing "C1" onto the top of the label stack. No modifications to the label tables are required at step 714, so the packet is then returned to the fast path component 10. The packet now has the label "C1" at the top of its label stack. The fast path component 10 of the first node 302 forwards the packet to the second node 304 at step 716.

10 Packet 3 in Figure 4C, also being labelled "C0", is also forwarded to the active processing engine 14. However, this time the active processing engine 14 determines that further active processing is not required at any subsequent active nodes 304, 306, and therefore pops "C0" off the top of the stack and pushes the label "A" onto the top of the label stack of the packet at step 712. The packet is then forwarded through the fast path component 10 of the first node 302 with the label "A" at step 716, and the packet therefore follows the fast path through the second and third nodes 304, 306.

In Figure 4D, packet 4 contains "C0" at the top of its label stack and is therefore processed by the active processing engine 14 of the first node 302 at step 710. The processing of packet 4 determines that this packet should be sent on an alternate route (this may be due to the detection of congestion or link failure by an application executing within the active processing engine 14 of the first node). Within the active processing engine 14, "C0" is popped off the stack, and "C3" and "Z" are pushed onto the label stack at step 712. This results in the packet being forwarded to the third node 306 on the alternate route "Z", bypassing the second node 304. At the third node 306, the label "C3" causes the packet to be forwarded to the active processing engine 14 of the next active node. Subsequent packets can be forwarded on the alternate LSP Z without further processing by modifying, at step 714, the configuration (*i.e.*, the MPLS table) of the fast path component 10 of the first node 302 to push "Z" onto the label stack of all packets arriving at the switch with label "A", for example. The subsequent diversion of a packet onto the alternate route "Z" by the fast path component 10 is shown in Figure 4E.

- 10 -

As described above, the use of label stacks enables flexible forwarding by allowing active applications to change the forwarding decision for packets being processed by the active processing engine 14, and for packets forwarded on the fast path component 10 without
5 active processing. In addition, LSPs can be configured to ensure that packets are processed by active processing engines 14 only at nodes where processing is required. For default best effort packet forwarding, the MPLS Label Distribution Protocol, as described in <http://www.ietf.org/internet-drafts/draft-ietf-mpls-ldp-09.txt>, can be used. The ability of active applications to modify the MPLS label tables of the fast path component 10 enables
10 custom active routing protocols to be introduced into the network. In order to create LSPs between active processing engines 14, an additional label distribution method is used, such as the Resource ReSerVation Protocol, or RSVP, as described in RFC 2205 and related RFC documents. However, network performance can be improved by using a resource reservation protocol customised to the properties of the active application.

15

In an alternative embodiment, the first active packet or packets of an application stream is used to determine which active processing engines 14 are to be used on the route and to create the LSPs between them on demand. The first packet follows the predetermined LSPs C0, C1, C2, and C3 between the active processing engines 14 of the active nodes.
20 This active packet determines the capabilities of each node 302 to 306, and on the basis of these capabilities determines what operations are to be performed by each node. On this basis, the active packet configures LSPs for the remainder of the packets.

For example, the active packet can use operating system calls of the processing
25 components 12 to determine whether sufficient resources (*e.g.*, processor cycles, physical memory) are available at a particular active node to perform the required processing. For example, the processing can include a processor-intensive process such as encryption or decryption of packet data. In such a case, the active packet is sent through the MPLS network with labels that indicate the packet is to be sent to every active node in the
30 network. This can be achieved by labelling the active packet with a stack of labels such as C0, C1, C2, C3, ..., or preferably a single label that the fast path 10 of each active node

- 11 -

recognises as referencing its processing component 12. After visiting all of the active nodes in the network, or after finding one or more suitable active nodes sufficient to perform the required processing, the active packet can then configure the active nodes in the network to include suitable LSPs (if they do not already exist) and/or to label active packets of the application in order to direct such packets to the active nodes identified by the first active packet(s) as having sufficient resources to perform the required processing. A content switch at the edge of the MPLS network is also configured to label the active packets of the application accordingly. These configuration changes are performed using a suitable resource reservation protocol, as described above. The result is the creation of LSPs that bypass active processing in some nodes and result in active processing in others.

For example, if processing is only required in the second node 304, then an additional LSP is created (in this case LSP D of Figure 3) to bypass active processing in the first node 302. A packet enters the network with label D on top of its stack, followed by label A. The packet is forwarded on the fast path component 10 through the first node 302 and processed by the active processing engine 14 of the second node 304. The active processing engine 14 of the second node 304 pops label D from the label stack of the packet. The packet is then forwarded through the fast path component 10 of the third node 306, as specified by label A.

Existing MPLS switches are used to switch packets based on a single label on the top of the label stack of the packet. In a further alternative embodiment, the flexibility of forwarding in the fast path component 10 is enhanced by allowing the switching of packets based upon multiple labels in a packet simultaneously. This can be achieved by modifying internal hardware components of an existing MPLS switch or by adding a multiple label switch hardware component to the ports of an existing MPLS switch. For example, a loopback connection 70 can be added to an existing MPLS switch that allows different labels to select different output ports, as shown in Figure 6. The loopback 70 enables more than one MPLS switching operation on a packet by recycling the packet through the switching fabric of the fast path component 10.

- 12 -

In the example of Figure 6, the traffic aggregate is directed through the network on LSP A. An additional label is placed under A by an active node or content switch at the edge of the MPLS network, as described above. Ordinarily, the underlying labels are ignored. However, the packet can be switched on the basis of both labels using the following
5 procedure. The active processing engine 14 determines that packets labelled Cy are to be redirected on LSP Z. The forwarding control component 16 of the processing component 12 then manipulates the MPLS tables of the fast path component 10 to pop the top label of packets labelled A and to redirect those packets to the loopback output port 1. The loopback 70 returns the packets to the fast path component 10 at input port 1. The MPLS
10 tables of the fast path component 10 are also modified so that packets arriving at input port 1 are switched as follows: if the packet label is Cx, then push label A onto the packet's label stack and send the packet to output port 2. Conversely, if the packet label is Cy, then push label Z onto the packet's label stack and send the packet to output port 3. The effect of this process is to switch the packets based on multiple labels, rather than only the top
15 label used by standard MPLS switching.

Alternatively, the top label can represent path information and the lower label can represent application information (such as customer number, or packet dropping priorities, for example). One example of this use of multiple labels is an implementation of
20 application-specific packet dropping or rerouting algorithms. In this example, an application stream follows a path defined by an MPLS label in the fast path 10. If an application executing in the processing component 12 detects congestion, then the fast path component 10 can be configured (by switching based on the values of the top two labels) to drop packets belonging to certain low-priority components of the stream (as indicated by
25 the application-specific second label).

The active nodes described above use standard MPLS switches as the fast path component 10. However, in order to implement a high degree of flexibility, a large number of LSPs may be required to ensure that packets are only processed by active processing
30 engines 14 at selected active nodes. In yet a further embodiment, the number of LSPs required is reduced by using reserved bits in the MPLS header to indicate which packets

- 13 -

require processing by the active processor. These reserved bits are currently not used by standard MPLS switches. This provides a more scalable active network solution, and only requires modifications to be made to the MPLS switches associated with active nodes; intermediate non-active MPLS switches do not need to be changed.

5

The active nodes described above can be used to provide active network services in a legacy carrier grade non-active network, as shown in Figure 5. The network includes six non-programmable MPLS switches 516 to 520, a number of active processors 12, a Processor Farm (PF) 530, Content Switches (CS) 525, 526, Active Edge Routers (AER) 10 536, and a Non-Active Router 508. The active processors 12 are connected to four MPLS switches 516, 518, 519, 520 by packet i/o ports and also by a switch management port to set-up LSPs and change MPLS table entries. The Content Switches 525, 526 are capable of switching packets at line rate based on information within the packet header, and are used to extract packets from an aggregate flow, and to insert the appropriate MPLS labels to 15 enable processing in required nodes within the carrier grade network 504. The Active Edge Routers 536 are programmable nodes with processing and MPLS capabilities. The Non-Active Router 508 is an edge switch that is not aware of active applications.

A particular PVN application provides a multimedia multicast service that supports many 20 features such as transcoding and the addition of local or targeted advertising, (*i.e.*, processor-intensive active applications of the application modules 17). In Figure 5, a PVN logical topology 500 is mapped over the physical infrastructure of the network. This PVN 500 connects a network 502 in Singapore through the Australian carrier grade network 504 to a network 506 in Sydney and a network (not shown) in Perth. The 25 Singapore network 502 is designed and implemented (using specialised hardware) to support proprietary protocols. The PVN topology 500 enables a company based in Singapore to effectively extend their network to customers in Australia without implementing their own hardware. To support their multimedia service, the company needs to peer their non-active edge router 508 in Singapore to a virtual router 510 in the 30 Australian carrier grade network 504 that supports their proprietary multicast, routing and billing protocols. They also require virtual routers 512 with similar capabilities near their

- 14 -

customer bases in Sydney and Perth. In addition, they require a virtual router 514 between Sydney and Perth to enable efficient multicast and caching.

In order to support the multimedia PVN over the Australian carrier grade network 504, active processing components 12 are connected to several existing MPLS switches 516, 518, 519, 520. An active processing component 12 is connected to the MPLS switch 516 peering the Singapore network, defining a first active node 532. In addition, active processing components 12 are connected to an MPLS switch 519 located in Sydney, defining a Sydney active node 528, and another located in Perth. The intermediate virtual router 514 between Sydney and Perth (required for efficient multicast) requires an active processing component 12 connected to an MPLS switch 520 within the core of the carrier grade network 504, defining an intermediate active node 534. The position of this MPLS switch 520 is determined by the delay constraints of the multimedia service. Because the Singapore network 502 is a proprietary network, a content switch 526 is placed at the edge router of the Australian carrier grade network 504. This content switch 526 uses filters to extract packets that belong to the multimedia PVN and to push the appropriate MPLS labels onto the label stacks of these packets. Content switches are not required in the remainder of the Australian carrier grade network 504 because the active edge nodes 536 in Sydney and the active edge nodes in Perth are programmable and place packets belonging to the PVN onto the appropriate LSPs. In this example, the active edge nodes 536 in Sydney are fully integrated programmable routers that support the PVN.

The carrier grade network 504 may also require one or more processor farms 530 for performing processor-intensive work that cannot be adequately supported by the active processing components 12 of the active nodes 528, 532, 534. For example, the router 508 in the Singapore network 502 wishes to multicast packets to nodes in the Sydney network 506 and the Perth network. The Singapore network 502 subscribes to a PVN multicast service. The content switch 526 in the first active node 532 extracts all packets belonging to a particular media streaming application (determined from the network address and/or port range of the packets) and places these packets on the multimedia PVN destined for the second node 304. The packets reach the intermediate active node 534 and are forwarded by

- 15 -

a multicast protocol to the Sydney and Perth networks 506. The virtual router 514 of the intermediate node 534 determines that the multicast branch destined for the Perth network requires a high level of processing, such as the addition of localised advertising. There is insufficient processing power in the intermediate node 534 to perform this processing
5 locally. The label for a pre-configured LSP linking a nearby processor farm (PF) 530 is pushed onto the stack for all packets belonging to this media stream. The processor intensive work of adding localised advertising is performed by the processor farm 530 and the packets are then returned to the intermediate node 534 for forwarding. The LSP may also indicate a particular virtual processor or execution environment within the processor
10 farm 530, as described above.

If the same multicast originated in the Sydney network 506 which contains active edge routers 536, then the content switch 525 is not required because the active edge routers 536 direct the appropriate multimedia packets onto the multicast PVN. However, the addition
15 of content switches in active-capable edge networks provides additional flexibility. For example, it enables the implementation of a reliable multicast protocol that requires processing of negative acknowledgements (NACKs) by branch nodes in the active multicast tree. A content switch in the edge network can be used to detect and add MPLS labels to NACK packets generated by hosts (which may be unaware of the reliable
20 multicast protocol) to enable active processing in the required nodes.

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as herein described with reference to the accompanying drawings.

- 16 -

CLAIMS:

1. An active node for a communications network, including:
a forwarding component for forwarding a packet of an active application;
a processor for processing said packet; and
5 a forwarding control component for modifying path data associated with said
packet based on said processing.
2. An active node as claimed in claim 1, wherein said forwarding control component
modifies path tables of said forwarding component.
10
3. An active node as claimed in claim 1, wherein said forwarding component is
adapted to allow processing of multiple path data entries of said packet.
4. An active node as claimed in claim 1, wherein said forwarding component includes
15 a loopback connection between input and output ports of said forwarding component to
allow processing of multiple path data entries of said packet.
5. An active node as claimed in claim 3, wherein said processing of said multiple path
data entries determines a route for said packet.
20
6. An active node as claimed in claim 1, wherein said processor includes one or more
microprocessors for executing active code of said packet.
7. An active node as claimed in claim 1, wherein said processor includes one or more
25 microprocessors for executing code to process data of said packet.
8. An active node as claimed in claim 1, wherein said forwarding component is
adapted to forward said packet directly to said network or to forward said packet to said
processor, based on path data of said packet and path data of said active node.
30

- 17 -

9. An active node as claimed in claim 1, wherein said path data includes at least one of a path stack of said packet and a path table of said active node.

10. A process for processing packets of an active application in a communications
5 network, including:

processing code of an active packet of said application to determine a path for said packets;

storing path data representing said path in said packets; and

forwarding said packets on said path.

10

11. A process as claimed in claim 10, wherein said step of forwarding includes forwarding said packets on said path based on a path table, and the process further includes modifying said path table to include second path data representing said path.

15 12. A process as claimed in claim 10, wherein said step of processing includes selecting a path to an active node based on capabilities of said active node and requirements of said application.

20 13. A process as claimed in claim 12, wherein said step of processing includes selecting a path to a processor of said active node based on capabilities of said processor and requirements of said application.

14. A process as claimed in claim 12 or 13, wherein said step of selecting is based on requirements of said packet of said application.

25

15. A process as claimed in claim 10, wherein said step of storing includes pushing said path data onto a path stack of said packets.

16. A process as claimed in claim 15, wherein said path data includes path labels.

30

- 18 -

17. A process as claimed in claim 16, wherein said path stack includes a multi-protocol label switching stack.
18. A process as claimed in claim 10, wherein said path data is stored in reserved bits
5 of MPLS headers of said packets
19. A process for processing an active packet in a communications network, including:
receiving an active packet including path data ;
processing said active packet;
10 modifying said path data in accordance with said processing; and
forwarding said packet in accordance with the modified path data.
20. A process as claimed in claim 19, wherein said path data includes a stack of one or
more path data entries.
15
21. A process as claimed in claim 20, wherein said step of modifying includes
removing an entry from said stack.
22. A process as claimed in claim 20, wherein said step of modifying includes adding
20 an entry to said stack.
23. A process for processing packets of an active application in a communications
network, including:
selecting, for one or more packets of said application, one or more paths to
25 respective processors based on requirements of said one or more packets and
capabilities of said processors;
storing path data representing said selected one or more paths in said one or more
packets; and
forwarding each of said one or more packets in accordance with said path data of
30 the packet.

- 19 -

24. A process for processing packets of an active application in a communications network, including:

5 determining requirements of said application from the first at least one packet of said application;

selecting one or more paths to respective processors based on said requirements and capabilities of said processors ;

storing path data representing the selected one or more paths in each packet; and forwarding said packets in accordance with said path data.

10

25. A process for processing an active packet in a communications network, including:

receiving an active packet including first path data and second path data;

removing said first path data from said packet; and

forwarding said packet in accordance with said second path data of said packet.

15

26. An active node having components for executing the steps of any one of claims 10 to 25.

27. A machine readable storage medium, having stored thereon program code for 20 executing the steps of any one of claims 10 to 25.

28. An active node for a communications network having:

a processor for processing active packets; and

25 a forwarding component for determining, on the basis of at least one label of a packet received from the network, to forward said packet to said processor or output said packet to said network.

29. An active node as claimed in claim 1, wherein said forwarding component is adapted to forward or drop said packet on the basis of at least one label of said packet.

30

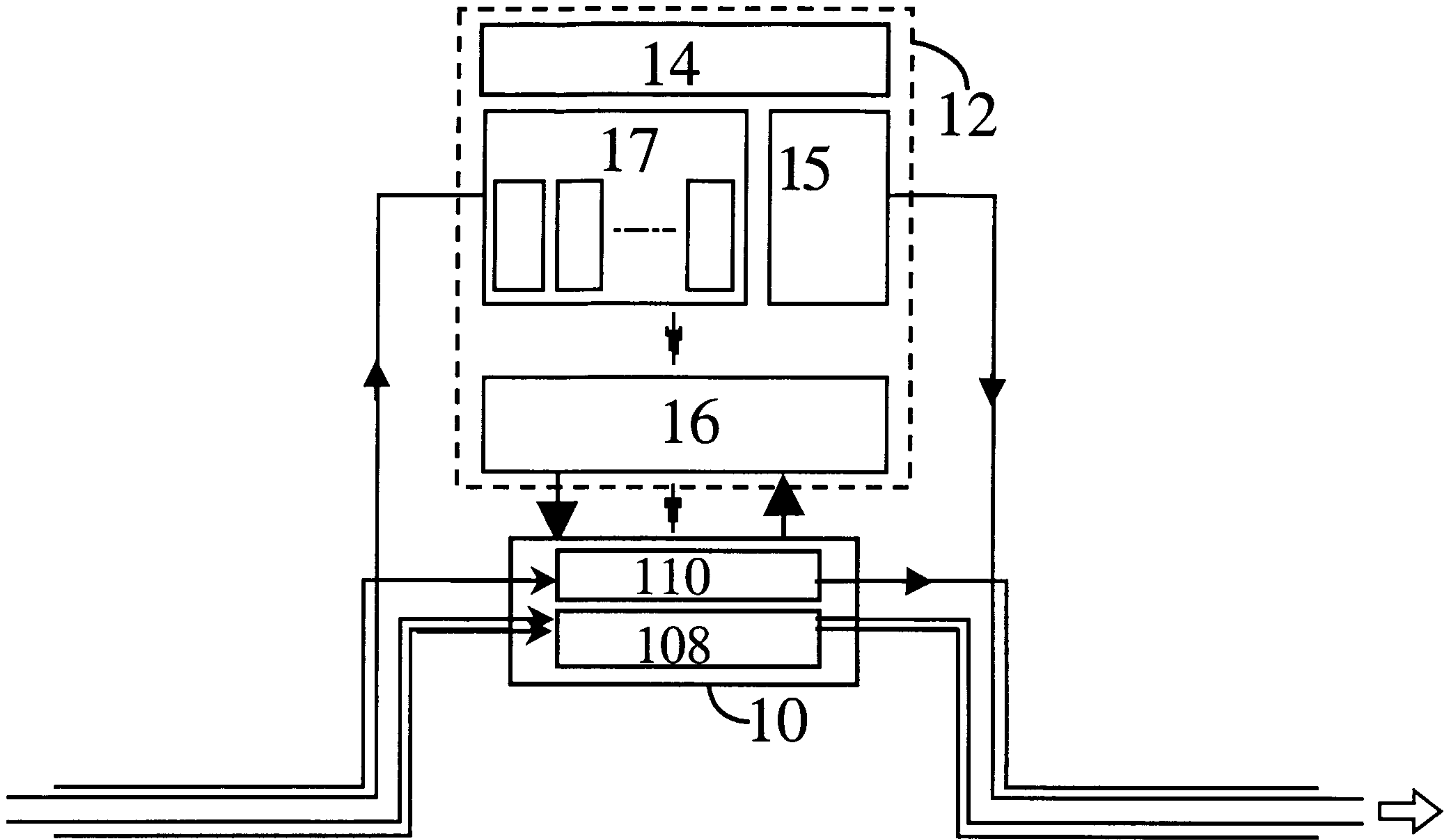


Figure 1

2/10

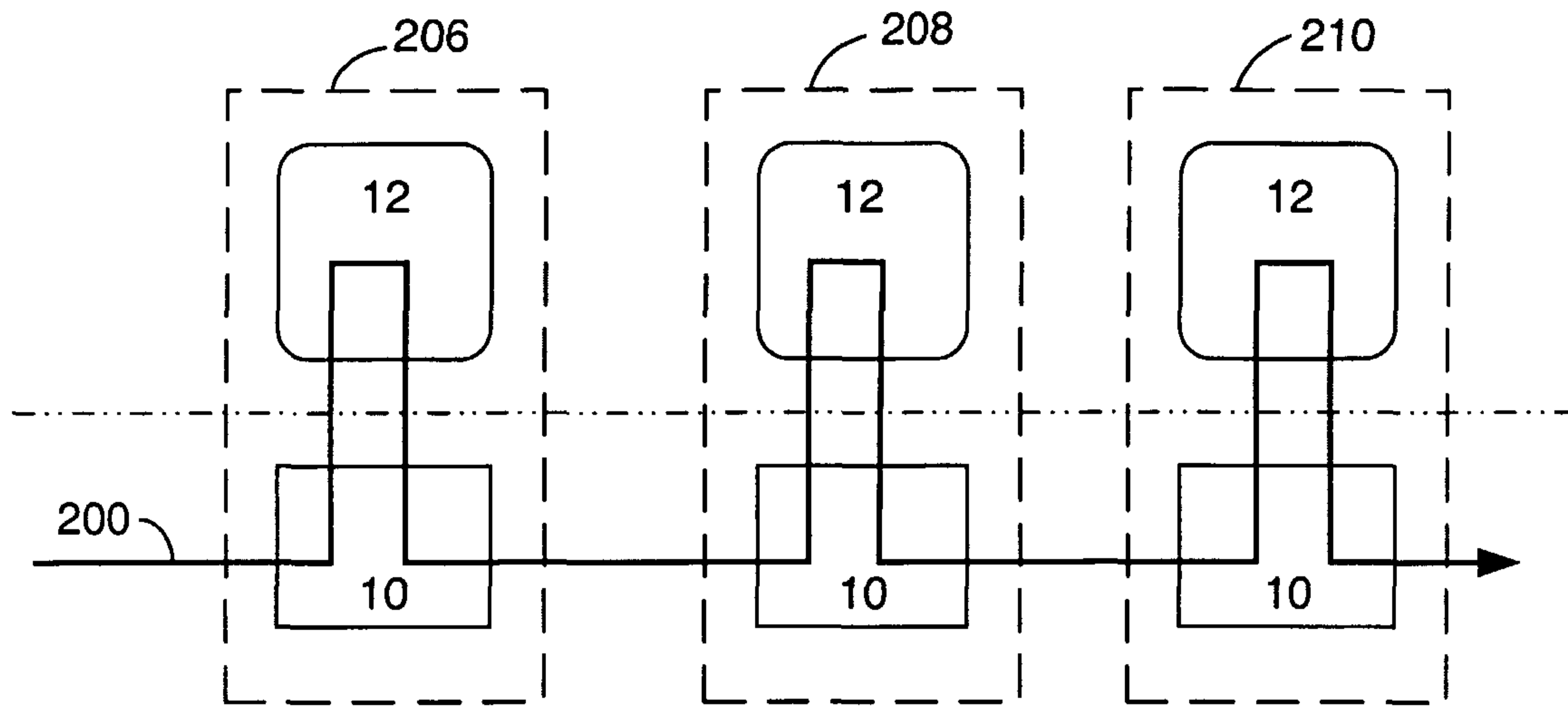


Figure 2A

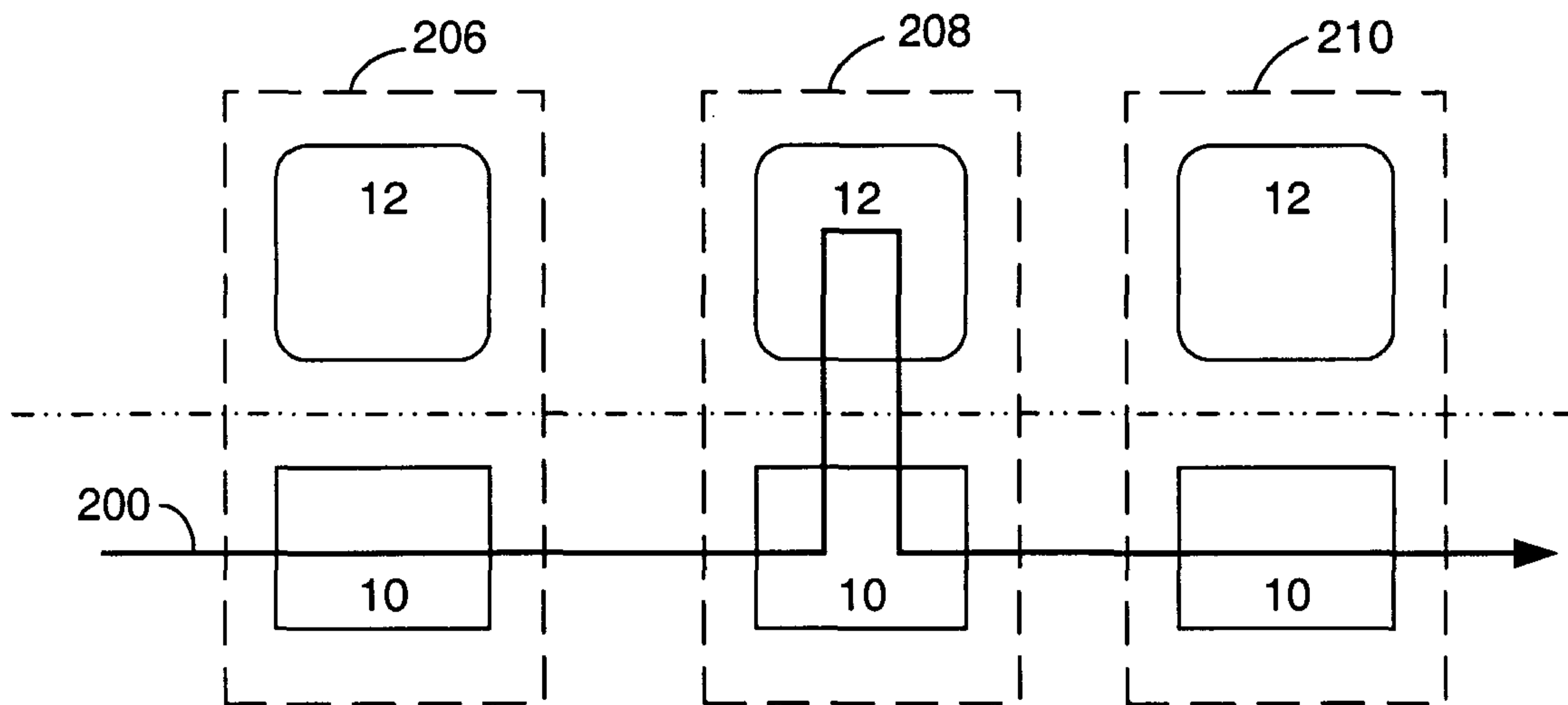


Figure 2B

3/10

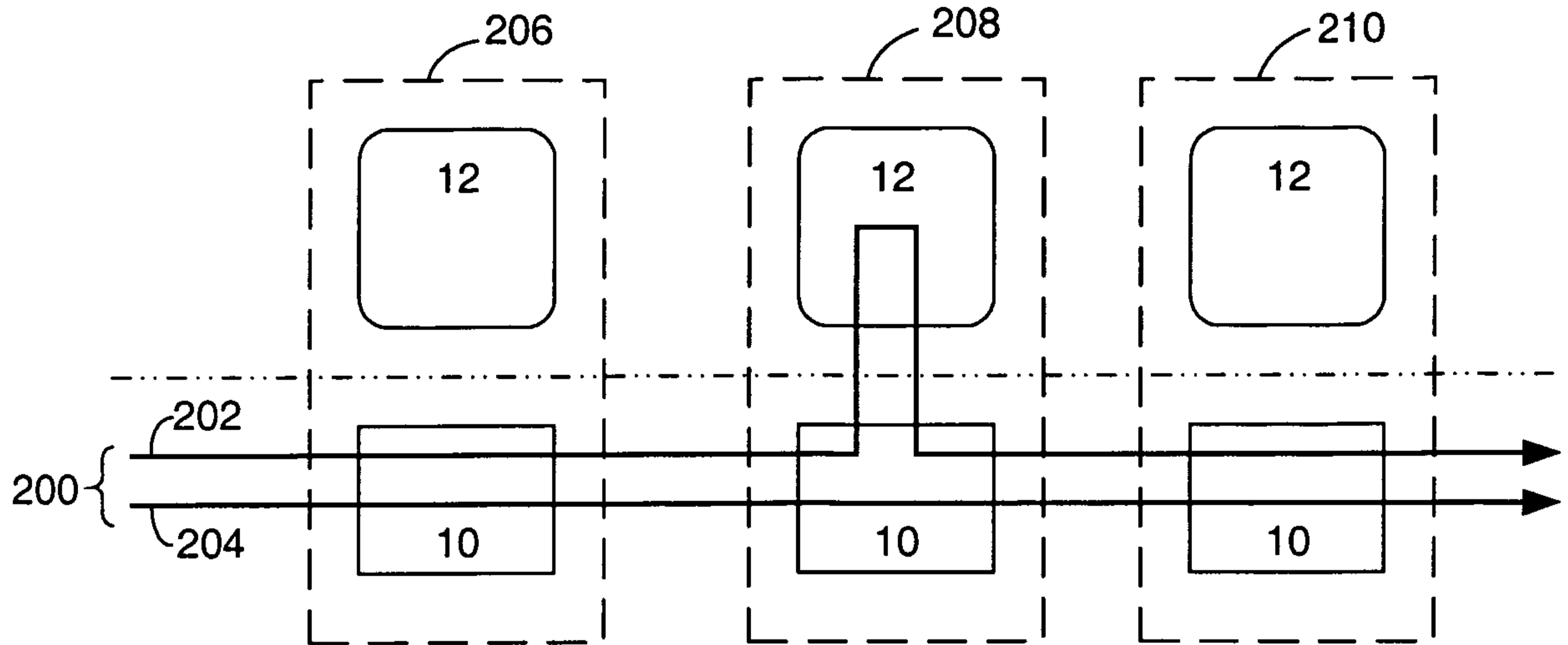


Figure 2C

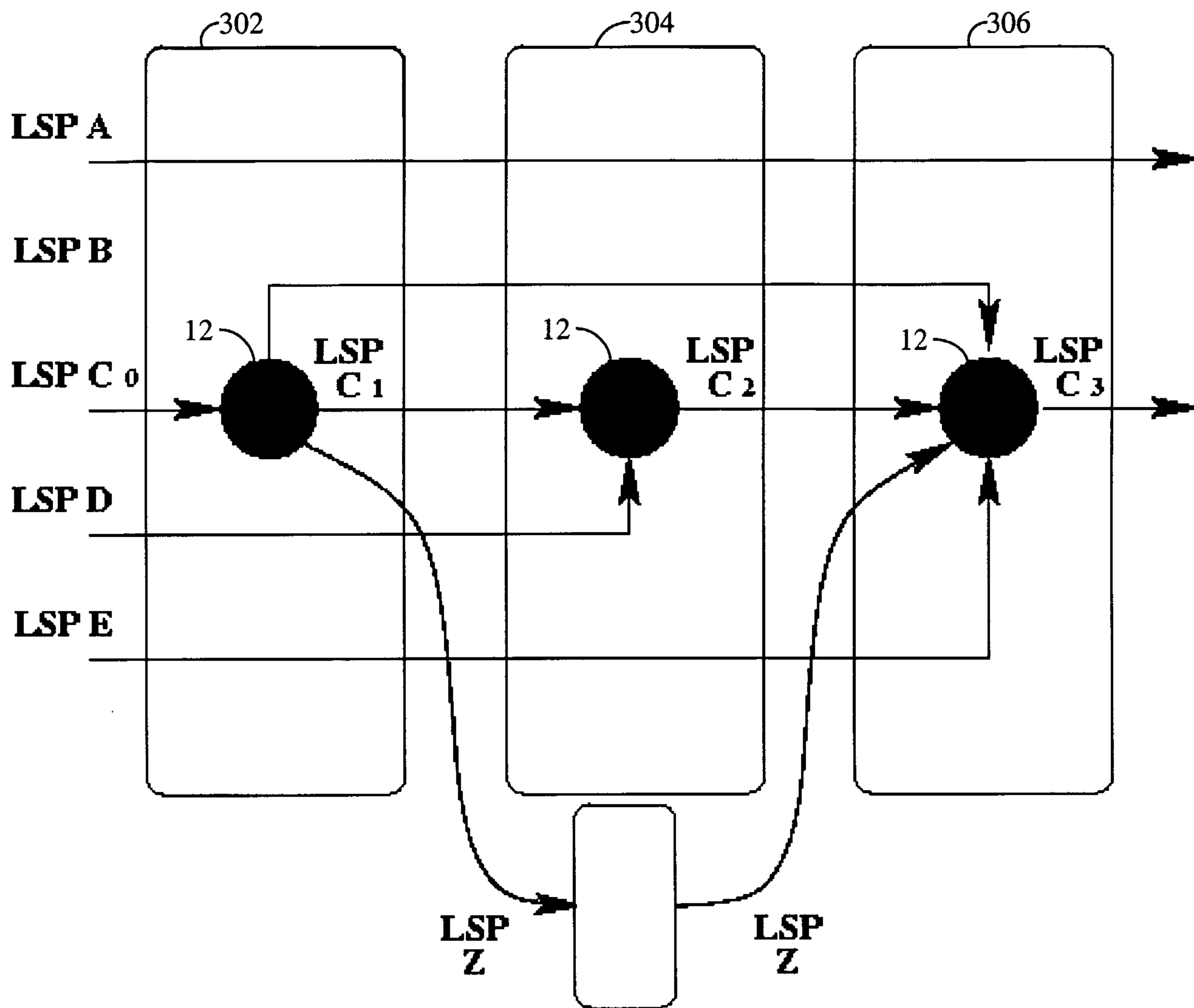


Figure 3

5/10

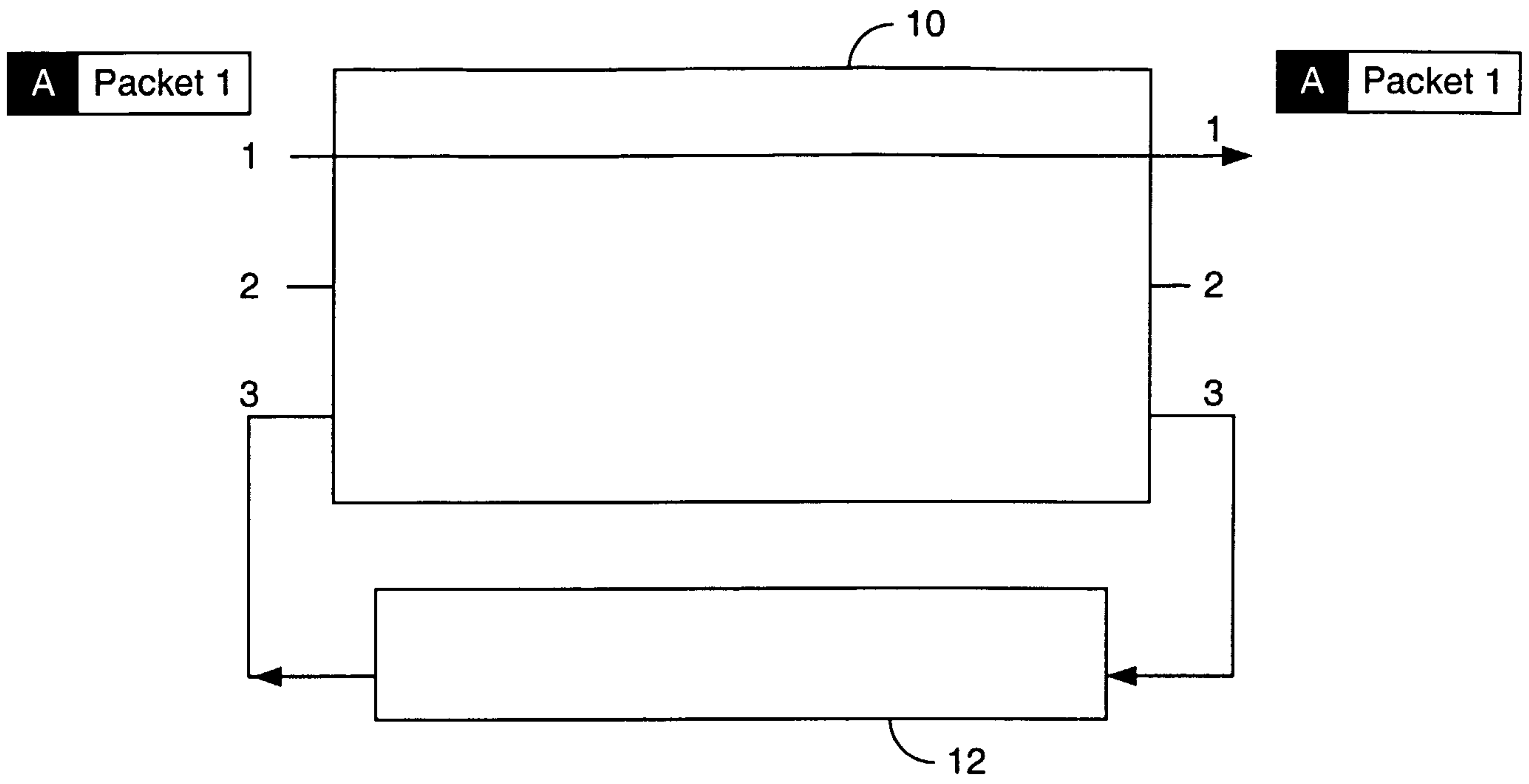


Figure 4A

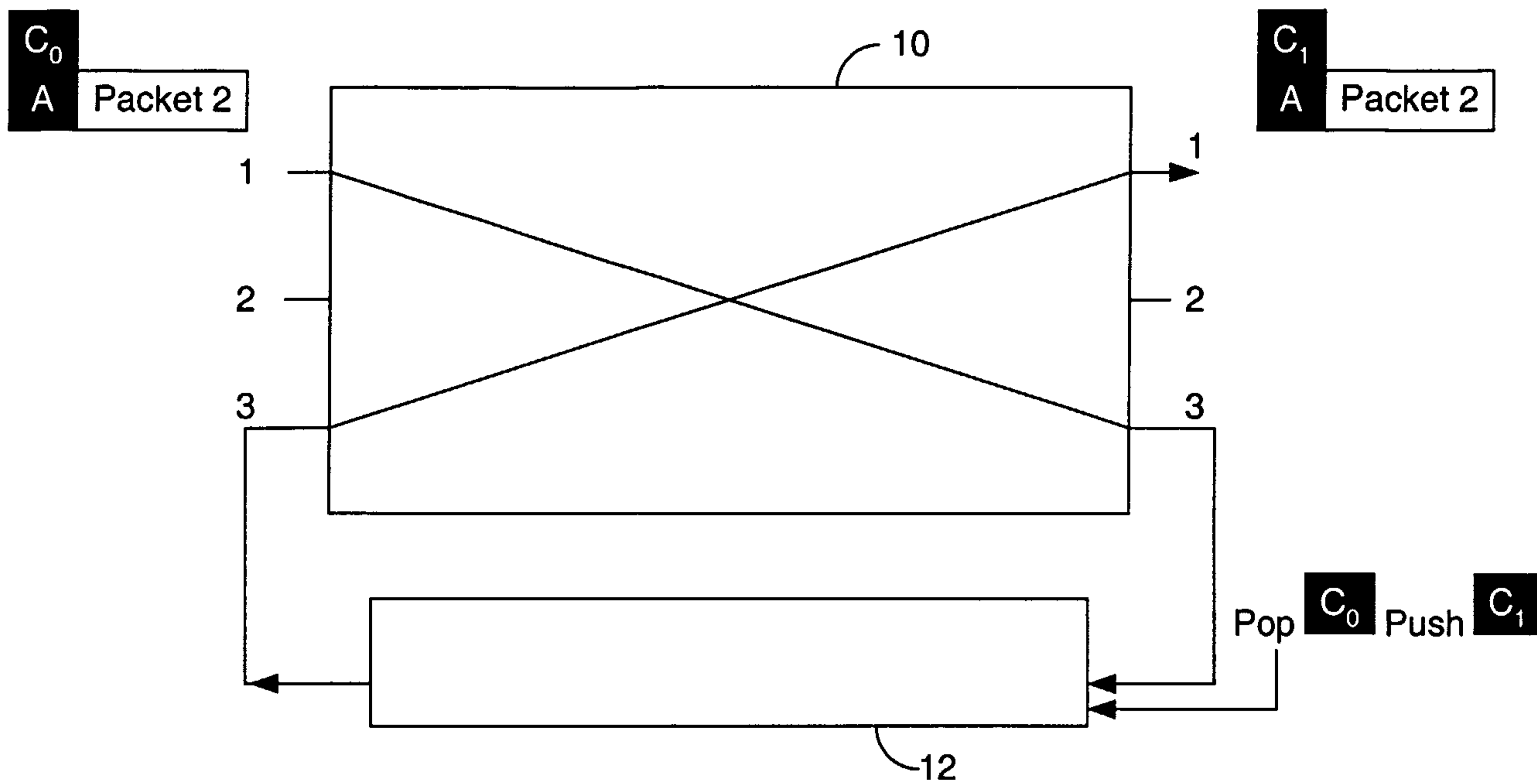


Figure 4B

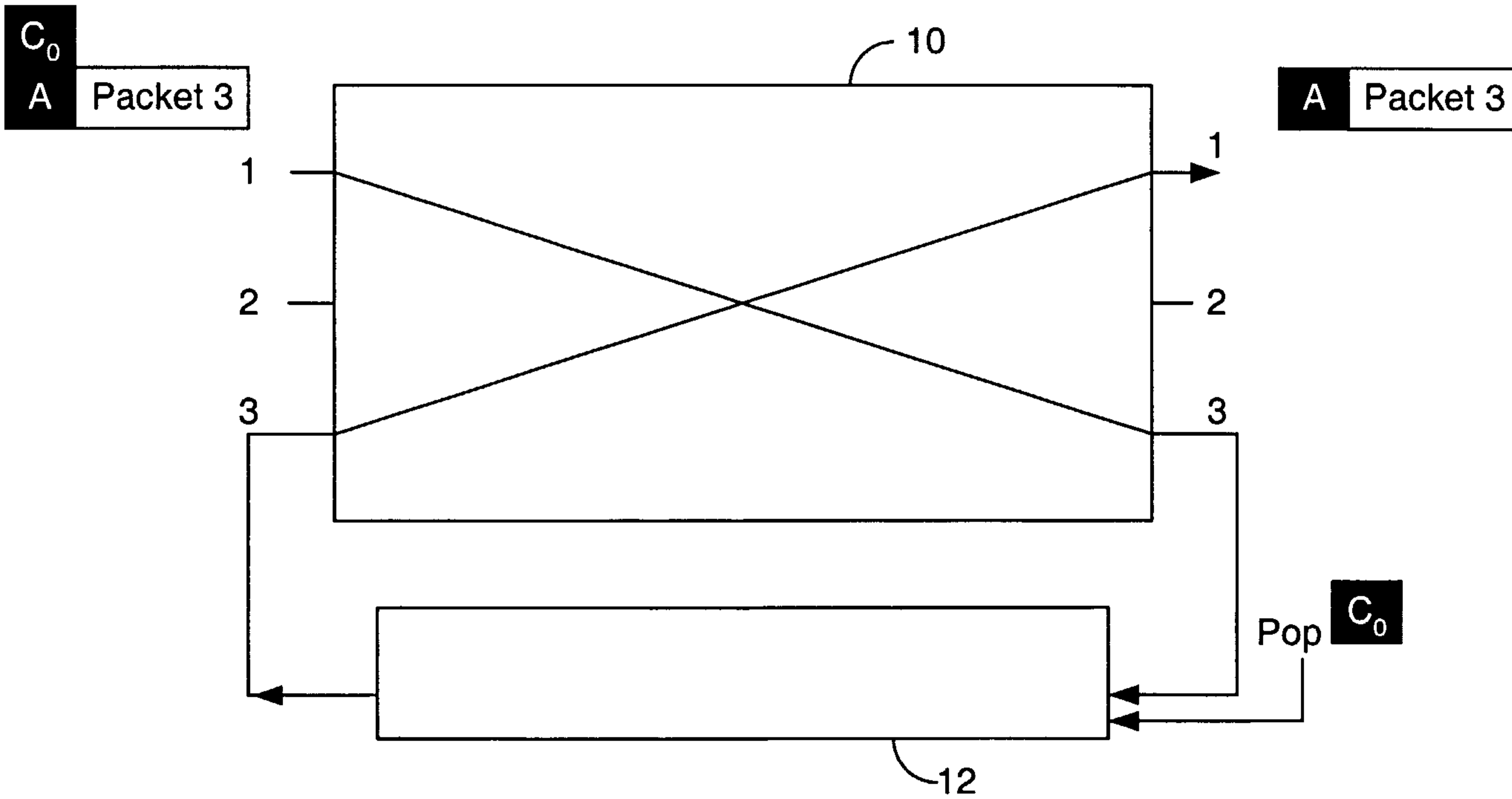


Figure 4C

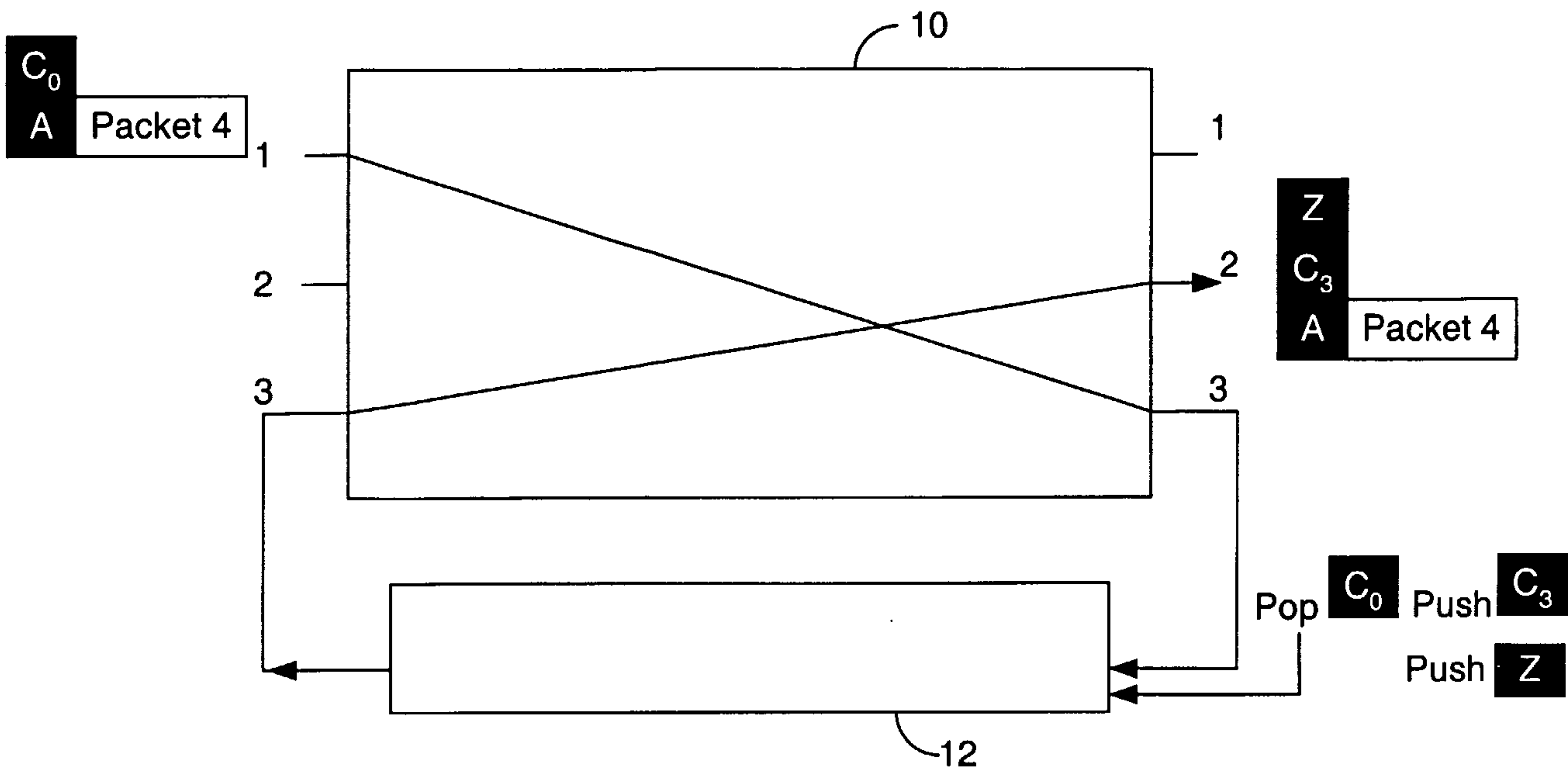


Figure 4D

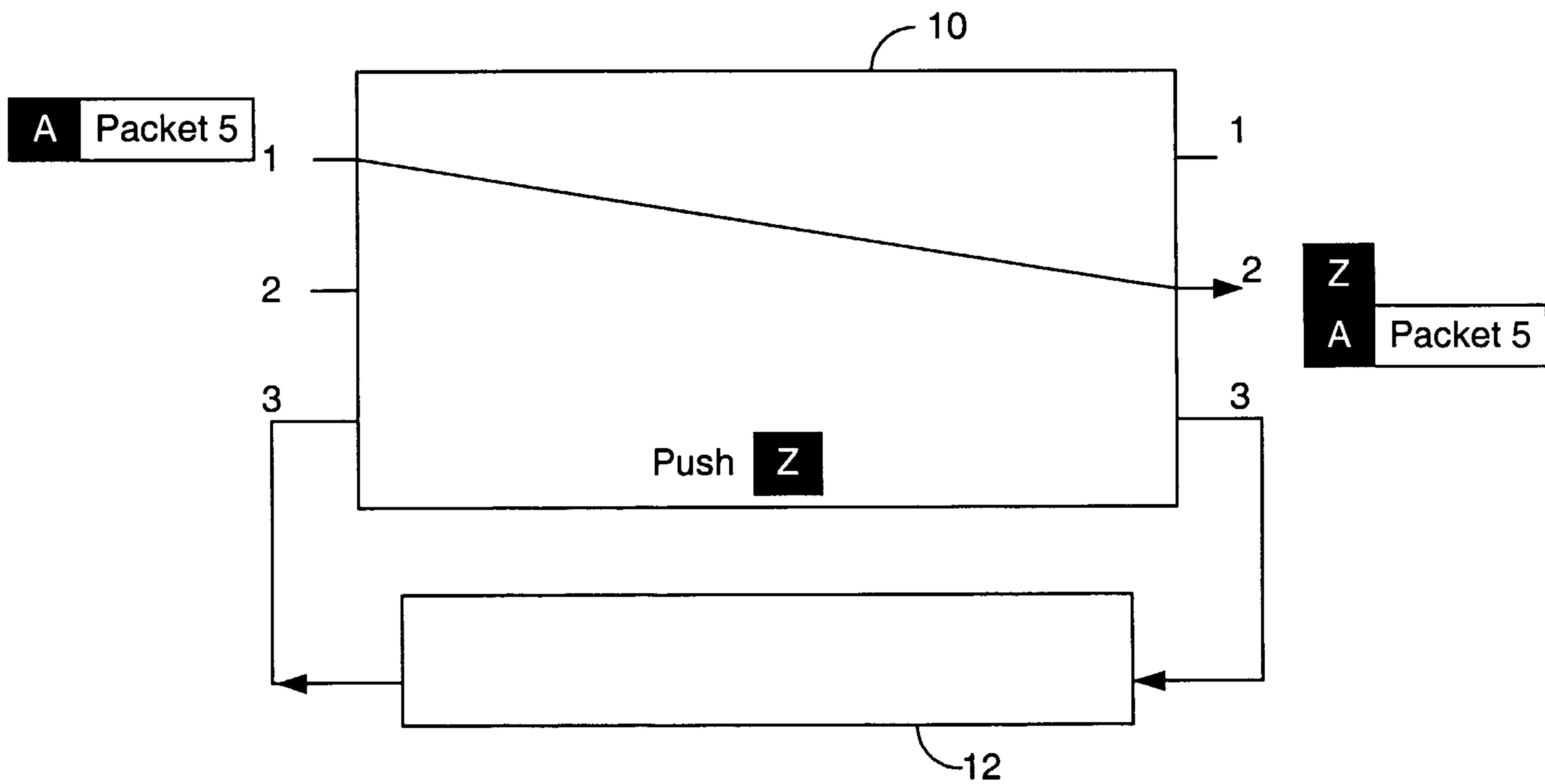


Figure 4E

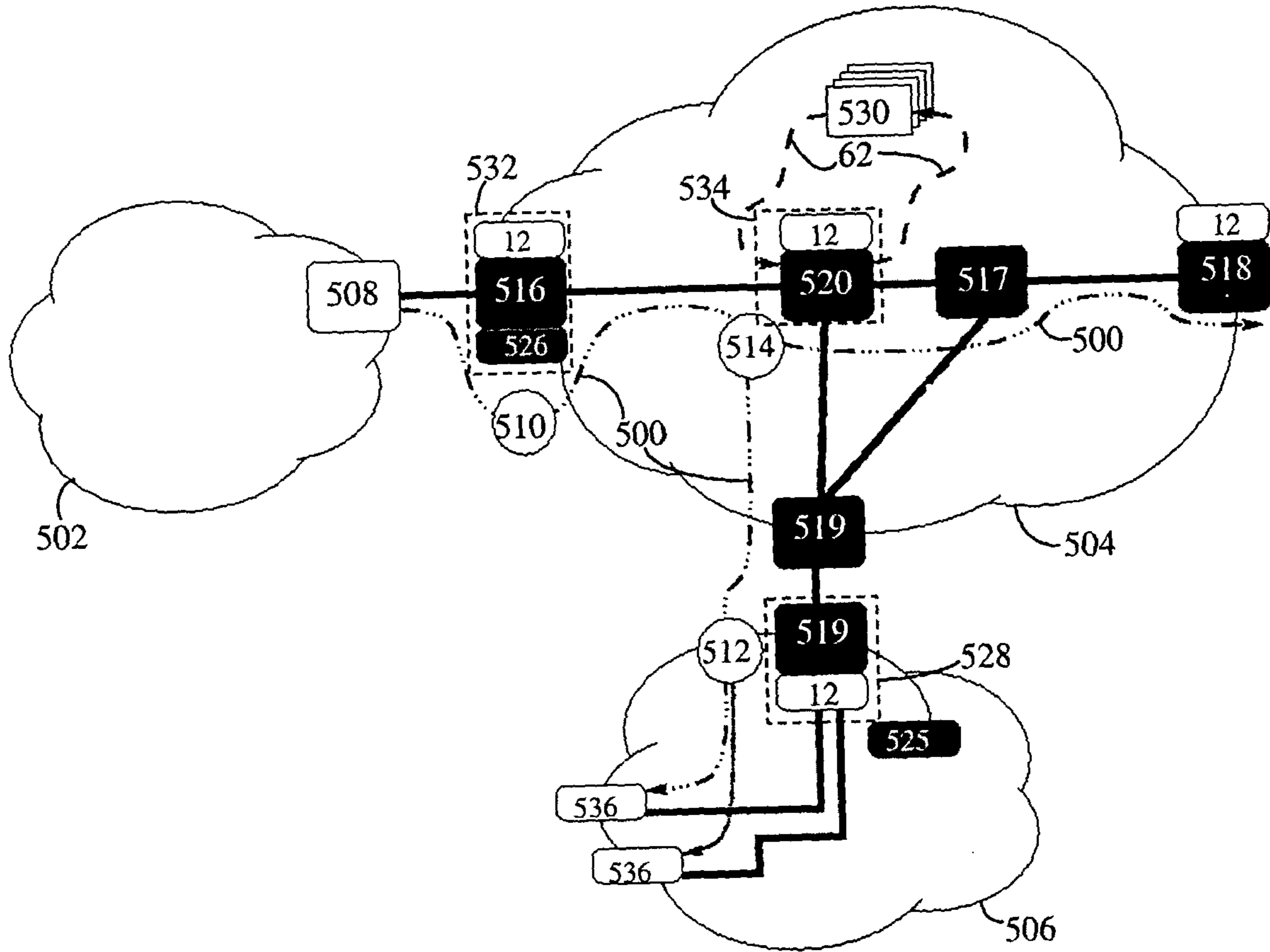


Figure 5

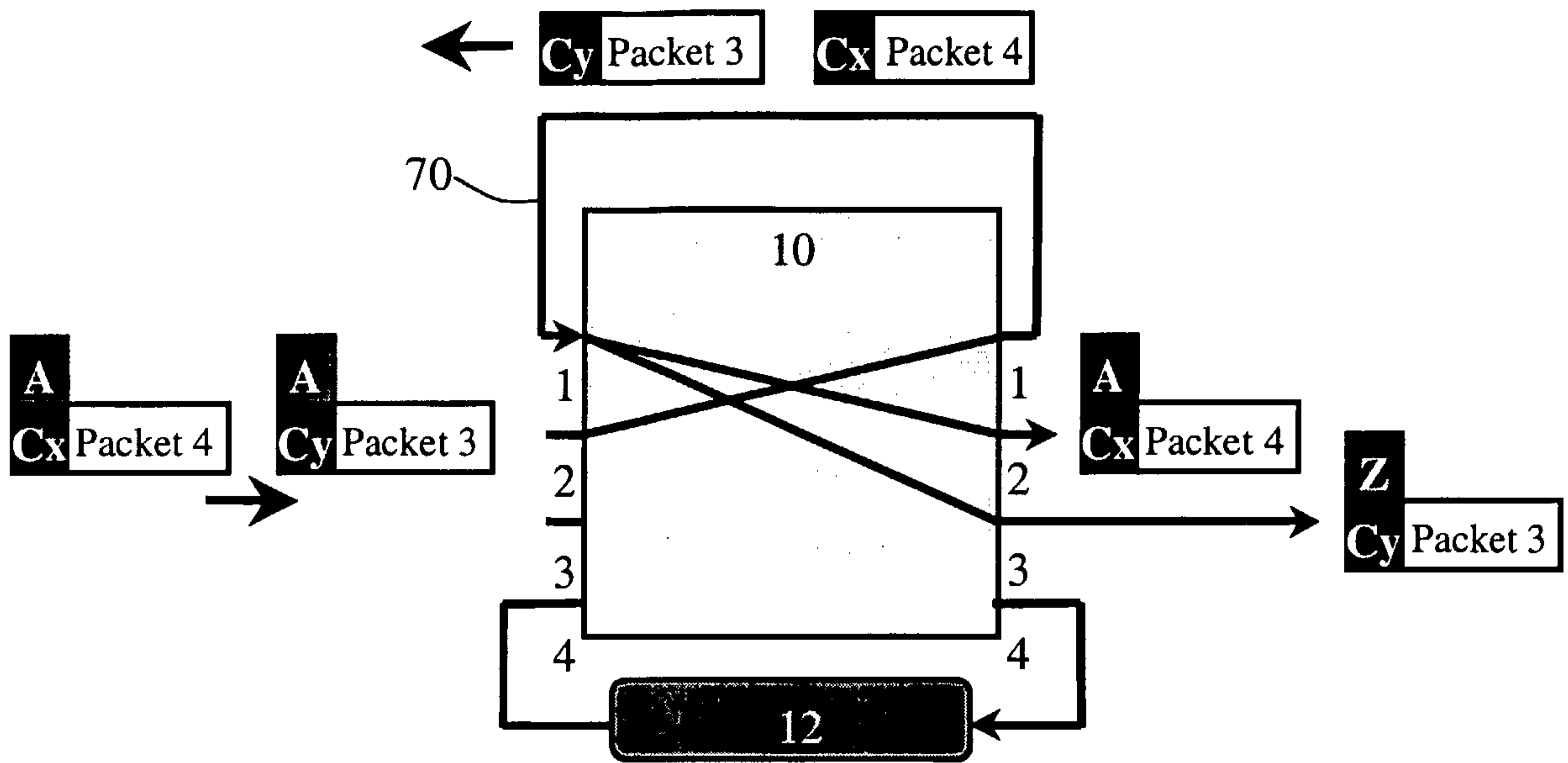


Figure 6

10/10

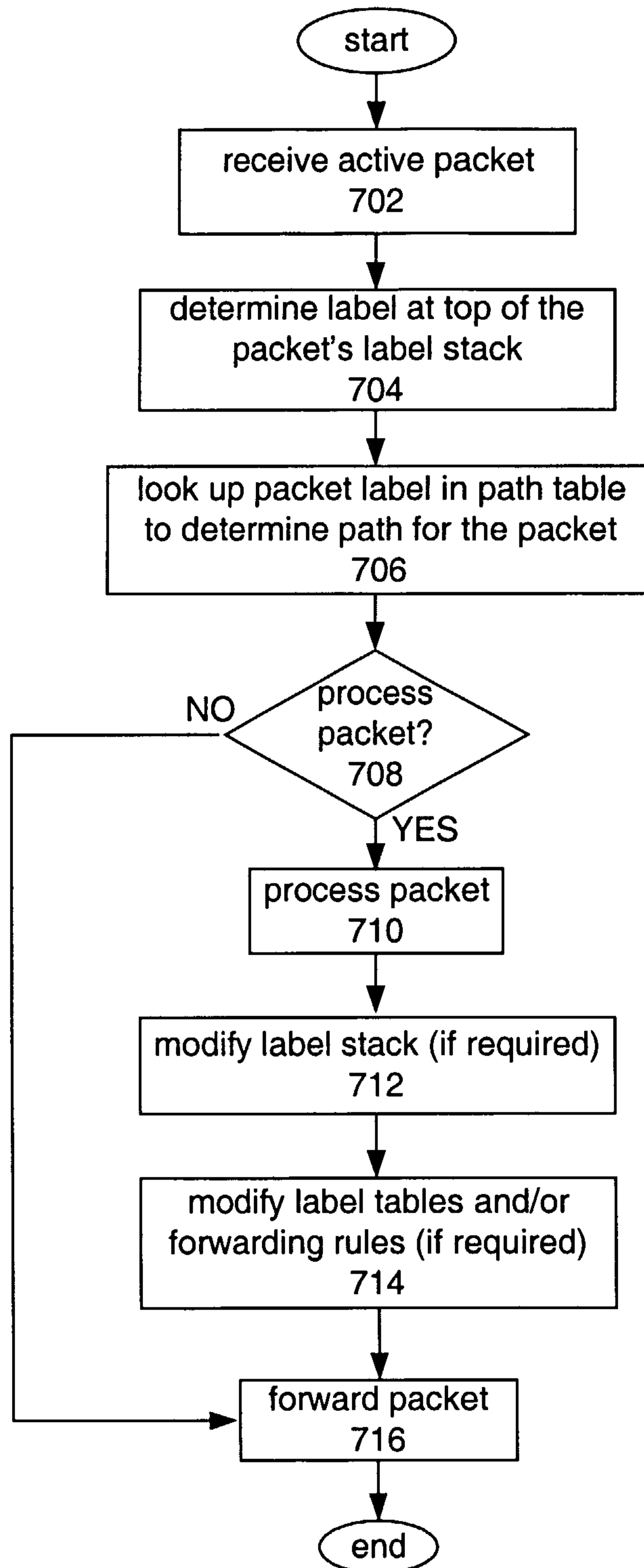


Figure 7

