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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
26 April 2007 (26.04.2007)

PCT

(10) **International Publication Number**
WO 2007/047153 A2

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

(21) **International Application Number:** PCT/US2006/039099

(22) **International Filing Date:** 5 October 2006 (05.10.2006)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
11/253,119 17 October 2005 (17.10.2005) US

(71) **Applicant** *(for all designated States except US)*: **CISCO TECHNOLOGY, INC.** [US/US]; 170 West Tasman Drive, San Jose, CA 95134-1706 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **WARD, David, D.** [US/US]; 301 221st Avenue, Somerset, WI 54025 (US). **SCUDDER, John** [US/US]; 900 Spring St., Ann Arbor, MI 48103 (US).

(74) **Agents: BROKAW, Christopher** et al.; Hickman Palermo Truong & Becker LLP, Suite 550, 2055 Gateway Place, San Jose, CA 95110-1089 (US).

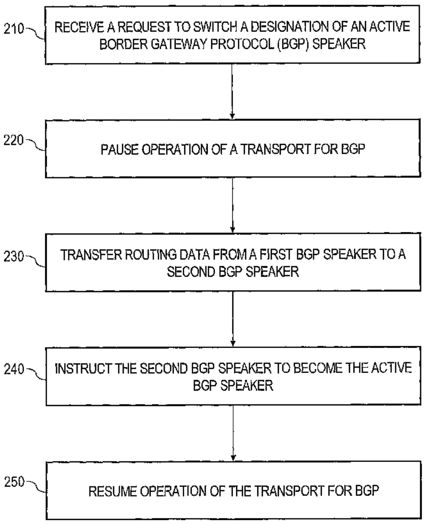
(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR,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

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

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

(54) **Title:** METHOD FOR RECOVERY OF A CONTROLLED FAILOVER OF A BORDER GATEWAY PROTOCOL SPEAKER


210
RECEIVE A REQUEST TO SWITCH A DESIGNATION OF AN ACTIVE BORDER GATEWAY PROTOCOL (BGP) SPEAKER
220
PAUSE OPERATION OF A TRANSPORT FOR BGP
230
TRANSFER ROUTING DATA FROM A FIRST BGP SPEAKER TO A SECOND BGP SPEAKER
240
INSTRUCT THE SECOND BGP SPEAKER TO BECOME THE ACTIVE BGP SPEAKER
250
RESUME OPERATION OF THE TRANSPORT FOR BGP

(57) **Abstract:** A method and apparatus for recovering from a controlled failover of a BGP speaker is provided. A user sends, to a network element, a request to switch a designation of an active Border Gateway Protocol (BGP) speaker of the network element from a first BGP speaker to a second BGP speaker. After receiving the request, the network element pauses operation of a transport for BGP. Routing data, which describes a state of a first routing information base (RIB) maintained by the active BGP speaker, is transferred from the first BGP speaker to the second BGP speaker. Thereafter, the first BGP speaker may instruct the second BGP speaker to become the active BGP speaker. After the second BGP speaker becomes the active BGP speaker, the second BGP speaker resumes operation of the BGP transport.

WO 2007/047153 A2

METHOD FOR RECOVERY OF A CONTROLLED FAILOVER OF A BORDER GATEWAY PROTOCOL SPEAKER

FIELD OF THE INVENTION

[0001] The present invention generally relates to recovering from a controlled failover of a BGP speaker on a network element.

BACKGROUND

[0002] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

[0003] Border Gateway Protocol (BGP) is a path vector routing protocol for inter-Autonomous System routing. The function of a BGP-enabled network element (a BGP host or peer) is to exchange network reachability information with other BGP-enabled network elements. The most commonly implemented version of BGP is BGP-4, which is defined in *RFC1771* (published by the Internet Engineering Task Force (IETF) in March 1995).

[0004] To exchange routing information, two BGP hosts first establish a BGP peering session by exchanging BGP OPEN messages. The BGP hosts then exchange their full routing tables. After this initial exchange, each BGP host sends to its BGP peer or peers only incremental updates for new, modified, and unavailable or withdrawn routes in one or more BGP UPDATE messages. A route is defined as a unit of information that pairs a network destination with the attributes of a network path to that destination. The attributes of the network path include, among other things, the network addresses (also referred to as address prefixes or just prefixes) of the computer systems along the path.

[0005] A BGP host stores information about the routes known to the BGP host in a Routing Information Base (RIB). Depending on the particular software implementation of BGP, a RIB may be represented by one or more routing tables. When more than one routing table represents a RIB, the routing tables may be logical subsets of information stored in the same physical storage space, or the routing tables may be stored in physically separate storage spaces.

[0006] As networks grow more complex and the number of BGP routes maintained by a particular network element increase, the consequences of a BGP host device, or the BGP process executing on the BGP host device, becoming inoperable are more severe. For example, in some scenarios, when a BGP host fails, the BGP host loses all information about routes maintained by the failed BGP host. Thus, recovery of the failed BGP host may require retransmission of a large amount of route information from other BGP hosts and the re-computation of a large amount of network reachability information by the recovering BGP host. During the retransmission period, the failed BGP host cannot route network traffic. Therefore, vendors of network gear and their customers wish to deploy BGP in a high availability manner.

[0007] One approach for deploying BGP in a high availability manner is referred to as "stateful switchover" or SSO. SSO is typically implemented with network elements that have dual route processors, each of which can host separate but duplicate instances of various software applications. One route processor is deemed Active and the other is deemed Standby. In one implementation of SSO, processes or "speakers" periodically transfer (in a process referred to as "checkpointing") a copy of large amounts of data, from one or more routing tables, from the Active BGP speaker to the Standby BGP speaker. In this way, the Standby BGP speaker may operate, using the same routes as previously used by the Active BGP speaker, when the Active BGP speaker becomes inoperable. Consequently, all data accumulated by the Active BGP speaker must be transferred to the Standby BGP speaker before the Standby BGP speaker can start processing BGP UPDATE messages or perform other substantive BGP functions.

[0008] However, this bulk data transfer approach is inefficient and does not scale as the volume of routes maintained by the Active BGP speaker increases. For example, the data structures that are transferred must be converted to messages for purposes of inter-process communications. Therefore, all data structures have to be flattened, i.e., pointers present in the data structures cannot be sent in the form of pointers. Further, as data structures change between versions of software, new messages and converter functions are necessary to provide SSO support between the different versions.

[0009] Some implementations of BGP SSO attempt to limit that amount of data that is transferred from the Active BGP speaker to the Standby BGP speaker at a single time by transmitting data that identifies a change made to the Active BGP speaker, from the Active BGP speaker to the Standby BGP speaker, as soon as the change is made to the Active BGP speaker. However, such an approach requires a large amount of overhead in updating the RIB of the Active BGP speaker because the RIB of the Standby BGP speaker must be updated synchronously with the RIB of the Active BGP speaker.

[0010] Another approach for deploying BGP in a high availability manner is referred to as "graceful restart." The graceful restart approach involves, for example, two different BGP hosts, denoted host A and host B herein. According to the graceful restart approach, if host A determines that host B may have become inoperable, host A starts a first timer that reflects the amount of time in which host A must receive a communication from host B before host A concludes that host B has become inoperable. If host A does receive a communication from host B before the expiration of the first timer, then host A starts a second timer that reflects the amount of time in which host B must send all BGP UPDATE messages to host A. On the other hand, if host A does not receive a communication from host B before the expiration of the first time, then host A updates the RIB it maintains to reflect that host B is not reachable.

[0011] Unfortunately, as a result of the time involved in updating the RIBs of each BGP speaker through the exchange of BGP UPDATE messages, the graceful restart approach requires several minutes or more before host A and host B are both updated after one of the hosts comes back online. Further, it is possible that host A would not be notified of a topology change in the network, because host B will not be able to communicate any BGP UPDATE messages to host A if host B is down.

[0012] Thus, there is a clear need for an improved technique for recovering from the failover of a BGP speaker on a network element which does not experience the disadvantages discussed above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Embodiments of the present invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0014] FIG. 1 is a block diagram that illustrates an overview of a network element that may be used to implement an embodiment;

[0015] FIG. 2 is a flowchart illustrating the functional steps performed by an embodiment;

[0016] FIG. 3 is a graphical depiction of various ways of transferring routing data according to several embodiments; and

[0017] FIG. 4 is a block diagram that illustrates a computer system upon which an embodiment may be implemented.

DETAILED DESCRIPTION

[0018] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

FUNCTIONAL OVERVIEW

[0019] Embodiments operate, at least in part, under the recognition that the amount of time required to recover from the failover of a Border Gateway Protocol (BGP) speaker may be reduced when the failover of the BGP speaker is the result of a controlled event. A controlled event refers to any event, initiated by a user, which requires shutting down a BGP speaker, including but not limited to a hardware upgrade, a software upgrade, or turning on a feature which requires restarting a BGP session maintained by the BGP speaker. Embodiments provide a mechanism for recovering from a controlled failover of a BGP speaker. Such a recovery may be performed by a user sending a command to a network element to pause a transport of BGP messages to the network element, transferring the BGP state and the transport state of an active BGP speaker to a standby BGP speaker, instructing the standby BGP speaker to become the active BGP speaker, and thereafter unpausing the transport of BGP messages to the network element. In this way, the active BGP speaker may be shut down in a controlled manner and the standby BGP speaker may synchronize to the active BGP speaker while minimizing the time in which the network element does not process BGP messages from peers.

[0020] More particularly, in one embodiment, a user may send, to a network element, a request to switch a designation of an active BGP speaker of the network element from a first BGP speaker to a second BGP speaker. In response to receiving the request, the network element pauses operation of a transport for BGP. For example, TCP may serve as a transport, and the network element may decide not to advance the window size of TCP segments sent to a peer. After the peer sends, to the network element, an amount of data up to size indicated by the window size, the peer will not send additional TCP segments to the network element until the network element advertises a positive window size. In this way, the network element may cause the peer to cease sending to the network element TCP segments carrying BGP messages. Other techniques for pausing the operation of a transport for BGP may also be used.

[0021] Routing data, which describes a state of a first routing information base (RIB) maintained by the first BGP speaker, is transferred to the second BGP speaker. Any of several transfer approaches may be used. As a result of transferring the routing data, a second RIB, maintained by the second BGP speaker, becomes synchronized to the first RIB. After the second RIB is synchronized to the first RIB, the first BGP speaker may instruct the second BGP speaker to become the active BGP speaker. After the second BGP speaker becomes the active BGP speaker, the second BGP speaker resumes operation of the BGP transport.

[0022] Having described a high level overview of one embodiment, an architecture overview of another embodiment will be presented below.

ARCHITECTURE OVERVIEW

[0023] FIG. 1 is a block diagram that illustrates system 10 that may be used to implement an embodiment. FIG. 1 depicts network elements 100 and 150 and console 120. A network element refers to any machine, such as a router or a switch, which resides on a network. Network elements 100 and 150 may exchange communications with each other over communications link 152. For example, network elements 100 and 150 may exchange TCP segments carrying BGP messages. Communications link 152 may be implemented by any medium or mechanism that provides for the exchange of data between network elements 100 and 150.

[0024] An administrator may wish to configure a particular network element, such as network element 100. In doing so, the administrator may use console 120 to issue commands to network element 100. Console 120 may be implemented by any machine that allows a user to transmit data to a particular network element over communications link 122, e.g., console 120 may be implemented by a computer operationally connected to network element 100 or a web browser that is indirectly coupled to network element 100. Communications link 122 may be implemented by any medium or mechanism that provides for the exchange of data between console 120 and network element 150. Examples of communications links 122 and 152 include, without limitation, a network such as a Local Area Network (LAN), Wide Area Network (WAN), Ethernet or the Internet, or one or more terrestrial, satellite or wireless links.

[0025] Network element 100 comprises an operating system 102. Operating system 102 is software designed to control the hardware of network element 100 in order to allow users and application programs to make use of it. For example, operating system 102 may receive a command from console 120 over communications link 122, and thereafter send the command to a software application executing on network device 100 for processing.

[0026] In one embodiment, network element 100 is a dual route processor network element that participates in a packet-switched network. Each of the dual route processors of network element 100 can host separate but duplicate instances of various software applications relating to packet routing or forwarding. To illustrate, as shown in FIG. 1, one of the dual route processors of network element 100, and the applications and data supported by it, is identified as route processor 110A and the other dual route processor, and the applications and data supported by it, is identified as route processor 110B.

[0027] In one embodiment, route processor 110A hosts a TCP module 120A that runs or controls a TCP high-availability (HA) process 140A and a BGP application 130A. Route processor 110B hosts a TCP module 120B, BGP application 130B, and TCP HA process 140B.

TCP modules 120A, 120B and TCP HA processes 140A, 140B provide transport control protocol functionality. BGP applications 130A and 130B provide border gateway protocol functionality. In alternative embodiments not depicted in FIG. 1, route processor 110A and route processor 110B may host other protocols and applications. For example, SCTP may serve as a transport. Thus, TCP modules 120A, 120B, TCP HA processes 140A, 140B, and BGP applications 130A, 130B are shown only as examples and not as requirements.

[0028] BGP Applications 130A and 130B each comprise switchover logic 132A and 132B respectively. Switchover logic 132A and 132B are responsible for recovering from a controlled failover of a BGP speaker. The operation of switchover logic 132A and 132B shall be described in greater detail below with reference to FIGS. 2 and 3.

[0029] Route processor 110A and route processor 110B may be arranged in a redundant or fault-tolerant configuration. To illustrate, route processor 110A may initially be designated an "active" route processor. Thus, BGP application 130A would be correspondingly designated an "active" BGP speaker. Similarly, route processor 110B may initially be designated a "standby" route processor. Thus, BGP application 130B would be correspondingly designated a "standby" BGP speaker. When route processor 110A is active, and route processor 110B is standby, all BGP functionality at network element 100 is performed by route processor 110A, and in the case that route processor 110A becomes inoperable (either for a controlled event or an unplanned event), then route processor 110B assumes the position of the active route processor, and subsequently performs all BGP functionality at network element 100.

[0030] Route processor 110A and route processor 110B each comprises a Routing Information Base (RIB) 160A and 160B respectively. RIB 160A stores all the routes known to BGP application 130A, and RIB 160B stores all the routes known to BGP application 130B.

[0031] An inter-process communication (IPC) service 104 is coupled between route processor 110A and route processor 110B. The IPC 104 provides a means for route processor 110A and route processor 110B to exchange data and instructions.

[0032] The arrangement of FIG. 1 represents just one possible context for applying the approaches described herein. The approaches herein are generally applicable to any other context in which BGP services are supported by a network element having a dual processor.

[0033] Having described the architecture overview of an embodiment, an approach for recovering from a controlled failover of a BGP speaker is presented below.

RECOVERING FROM A CONTROLLED FAILOVER OF A BGP SPEAKER

[0034] The process of recovering from a controlled failover of a BGP speaker according to an embodiment is described below with reference to FIG. 2. For purposes of illustrating a clear example, and not as a limitation, the description of the steps of FIG. 2 makes reference to the system 10 of FIG. 1. While the steps of FIG. 2 are depicted in a particular sequence, the particular sequence of steps depicted in FIG. 2 is merely to facilitate the explanation of embodiments of the invention. In other embodiments, the sequence of steps in recovering from a controlled failover of a BGP speaker may differ from those depicted in FIG. 2. For example, as explained below, in some embodiments the step of transferring routing data (step 230) may be performed in parallel to one or more other steps depicted in FIG. 2.

[0035] Initially, in step 210, a request to switch a designation of an active border gateway protocol (BGP) speaker from a first BGP speaker to a second BGP speaker is received. For example, route processor 110A may be currently designated as the active route processor. The request of step 210 may be a request to switch the designation of an active route processor from route processor 110A to route processor 110B. For ease of explanation, the steps of FIG. 2 are described below assuming that the request of step 210 is a request to switch the designation of an active route processor from route processor 110A to route processor 110B.

[0036] Switching the designation of an active route processor correspondingly switches the designation of an active BGP speaker. To illustrate, if the request of step 210 is a request to switch the designation of an active route processor from route processor 110A to route processor 110B, then BGP application 130B becomes the active BGP speaker. In other words, the active BGP speaker is the BGP speaker of the active route processor, and so a request may request to switch a designation of an active BGP speaker by requesting the designation of an active route processor be switched from route processor 110A to route processor 110B. Other embodiments may expressly request that a designation of an active BGP speaker be switched from one particular BGP speaker (such as BGP application 130A) to another particular BGP speaker (such as BGP application 130B).

[0037] A user, such as an administrator, may issue the switchover request of step 210 using console 120. For example, the request of step 210 may be implemented as a command to shut down the route processor 110A, and cause route processor 110B to become the active route processor. In an embodiment, the command may be a Command Line Interface (CLI) command that a user issues through console 120. The command may be sent over communications link 122 and subsequently received by the operating system 102 of network element 100. The operating system 102, upon receiving the command, communicates the command to switchover logic associated with the active BGP speaker. For example, if BGP application 130A is the active BGP speaker, then operating system 102, upon receiving the command, communicates the command to switchover logic 132A. After the request to switch the designation of the active

BGP speaker is received, processing proceeds to step 220. Alternately, a program or process may issue request 210.

[0038] In step 220, the operation of a transport of BGP to network element 100 is paused. The operation of a transport of BGP to network element 100 may be paused using a variety of different approaches. Three such approaches will be discussed herein, although any approach for pausing the operation of a transport of BGP to network element 100 may be employed by embodiments of the invention. For ease of explanation, the performance of step 220 shall be explained below with reference to pausing operation of a transport of BGP from network element 150 to network element 100, although the techniques below for pausing the operation of a transport for BGP may be used with any number of peers of network element 100.

[0039] In an embodiment, a transport-layer protocol may be used to pause the operation of the transport of BGP. Networks use transport-layer protocols to regulate the movement of data packets among network elements, such as network elements 100 and 150. Transmission Control Protocol ("TCP") is a foundational protocol that provides transport-layer services. TCP is defined in Internet Engineering Task Force (IETF) Request for Comments (RFC) 793. TCP may be used to carry BGP messages between BGP peers. A software implementation of TCP is often known as a TCP stack. A TCP stack may be implemented in a router, switch, or other network element of network infrastructure, or in a personal computer, server, workstation, or other end station device. The term "host" encompasses both network elements and end station devices. Two devices that have respective TCP stacks and communicate according to TCP are often called TCP peers or peer hosts. The basic unit of communication under TCP is a segment, which is encapsulated in an Internet Protocol (IP) packet when IP is the network layer protocol.

[0040] Data communication protocols, such as TCP, often provide a way for a receiving host to signal a sending host that the receiving host cannot accept more data, for example, because receiving buffers of the receiving host are full. To provide such signaling, TCP defines a segment attribute termed the window size. When a receiving host receives a data segment but cannot accept further data thereafter, the receiving host sends an acknowledgment TCP segment ("ACK segment") that contains a window size value of zero. A sending host, upon receiving a window size value of zero, understands that the receiving host cannot accept more data, and therefore the sending host must not send any more data to the receiving host.

[0041] However, the sending host needs a way to determine, subsequently, when the receiving host is again ready to receive data. In one approach, a sending host sends a probe segment that contains one byte of additional data. The purpose of the probe segment is to induce the receiving host to send back another ACK segment that contains an updated window size value. If the receiving host truly cannot accept more data, then the one byte of data in the probe segment may be lost, and re-transmission will be required.

[0042] According to a first approach (the "closing the window upon consumption" approach) for pausing the operation of a transport for BGP to network element 100, network element 100 may cease advancing the window size advertised by TCP segments sent from network element 100 to peers, such as network element 150. In other words, the advertised window size is decremented upon receipt of data from network element 150, and the window size is no longer incremented upon capacity of network element 100 becoming available for receiving data. Over time, as network element 150 sends data to network element 100, the advertised window size of TCP segments sent from network element 100 to network element 150 will decrease, until the window size shrinks to zero. The act of sending an amount of data to a host equal to the amount of data advertised by a window size from that host is referred to as "consuming" the window size.

[0043] After network element 150 has consumed the window size, network element 100 will advertise a window size of zero in TCP segments sent to network 150. Receipt of the TCP segment advertising a window size of zero by network element 150 will cause network element 150 to cease transmitting TCP segments to network element 100. In effect, network element 150 will become "flow blocked," or unable to send additional TCP segments to network element 100. By flow blocking network element 150, network element 100 may pause the operation of the transport for BGP from network element 150 to network element 100.

[0044] A second approach (the "not acknowledging TCP segments" approach) for pausing the operation of a transport for BGP to network element 100 involves network element 100 ceasing to acknowledge TCP segments received from network element 150. In the "not acknowledging TCP segments" approach, network element 150 will continue retransmitting the unacknowledged TCP segments. As explained below, after the standby BGP speaker of network element 100 becomes the active BGP speaker of network element 100, the active BGP speaker of network element 100 will resume acknowledging the TCP segments from network element 150.

[0045] According to a third approach ("the message approach") for pausing the operation of a transport for BGP to network element 100, switchover logic 132A of network element 100 may send a message to network element 150. The message, received by network element 150, may inform network element 150 that network element 100 can no longer receive any BGP messages or transport segments from network element 150. Network element 100 may also, in step 220, send other messages to other peers to pause the operation of a transport of BGP from the other peers to network element 100.

[0046] To illustrate, in an embodiment employing the message approach, a TCP segment is sent, from network element 100 to network element 150, which advertises a window size of zero. Further, network element 100 does not send any ACK segments to network element 150 in response to any probe segments sent by network device 150 to network device 100. Receipt of the TCP segment advertising a window size of zero by network element 150 will cause network element 150 to become flow blocked. The message approach differs from the closing the window upon consumption approach in that, in the message approach, the message is immediately sent regardless of what prior window size was previously advertised, but not consumed.

[0047] In an embodiment, the switchover logic associated with the active BGP speaker may instruct the TCP module of the active route processor to send the TCP segment advertising a window size of zero to network element 150. For example, in an embodiment, switchover logic 132A may instruct TCP module 120A to send a TCP segment advertising a window size of zero to network element 150. Further, switchover logic 132A may instruct TCP HA process 140 to not send any ACK segments to network element 150 while the transport of BGP is paused.

[0048] In some contexts, it may not be desirable to advertise a window size of zero if a larger window was previously advertised which the peer has not yet consumed (otherwise known as "reneging a window size"). For example, reneging on a window size may cause a peer to abort a connection. After a message that pauses operation of a transport of BGP is sent to the peer, processing proceeds to step 230.

[0049] In step 230, routing data that describes a state of a first routing information base (RIB) is transferred from a first BGP speaker to a second BGP speaker. For example, routing data that describe the current state of RIB 160A may be transferred from route processor 110A to route processor 110B. The routing data received at route processor 110B may be used to update RIB 160B. Thus, after all the routing data is transferred in step 230, RIB 160B becomes synchronized to RIB 160A.

[0050] Embodiments of the invention may perform step 230 in a variety of ways. Several approaches for performing step 230 shall be discussed below with reference to FIG. 3. Some embodiments of the invention may transfer routing data prior to the performance of step 210 or in parallel to other steps depicted in FIG. 3.

[0051] In an embodiment, during step 230, TCP connection data that describes the state of TCP connections maintained by the first BGP speaker may also be transferred. Non-limiting examples of TCP connection data includes the socket number of a TCP session, a file descriptor, and the state of a TCP session with a peer. In an embodiment, when the TCP connection data is transferred from route processor 110A to route processor 110B, the TCP state is frozen (i.e., the state no longer changes) at route processor 110A. TCP HA Process 140B is updated to reflect the TCP connection data transferred to route processor 110B. In this way, route processor 110B may transmit TCP segments to peers of network element 100 using the transferred connection data once route processor 110B is instructed to become the active route processor.

[0052] After switchover logic 132A determines that all the routing data has been transferred to route processor 110B (in other words, RIB 160B is synchronized to RIB 160A), then processing proceeds to step 240.

[0053] In step 240, the standby BGP speaker is instructed to become the active BGP speaker. Step 240 may be performed by switchover logic 132A instructing switchover logic 132B to cause BGP application 130B to become the active BGP speaker. Switchover logic 132A may instruct switchover logic 132B over IPC 104.

[0054] As explained above, the identity of the active BGP speaker corresponds to the active route processor, so in an embodiment, the performance of step 240 may be performed by switchover logic 132 instructing route processor 110B that route processor 110B is to become the active route processor.

[0055] In an embodiment, after BGP application 130B receives the instruction, from BGP application 130A, to become the active BGP speaker, BGP application 130B becomes the active BGP speaker. BGP application 130B sends an acknowledgement to BGP application 130A that BGP application 130B is preparing to become the active BGP speaker.

[0056] In an embodiment, after BGP application 130A receives the acknowledgement from BGP application 130B, BGP application 130A shuts down. Also, in other embodiments, in addition to BGP application 130A shutting down in response to receiving the acknowledgement, other applications of route processor 110A, or route processor 110A itself, may also shut down. After route processor 110A, or any applications supported by route processor 110A, shuts down, a user may perform activities on route processor 110A, such as installing new hardware or software. After the performance of step 240, processing proceeds to step 250.

[0057] In step 250, operation of the transport for BGP to network element 100 is resumed. As in step 220, the performance of step 250 may be performed using a variety of different approaches. Naturally, the particular approach taken to pause the operation of a transport for BGP performed in step 220 will affect how the operation of the transport for BGP is resumed in the performance of step 250.

[0058] In an embodiment employing the closing the window upon consumption approach or the message approach, the performance of step 250 may be performed by network element 110 transmitting to network element 150 a TCP ACK segment that contains an updated window size value that is positive. The updated positive window size value reflects the amount of data that the second BGP speaker is capable of receiving from the peer. As a result of network element 150 receiving a TCP segment advertising a positive window size, network element 150 will resume sending TCP segments to network element 100 in accordance with the updated window size.

[0059] In an embodiment employing the "not acknowledging TCP segments" approach, the operation of the transport for BGP may be resumed in step 250 by the active BGP speaker of network element 100 acknowledging the TCP segments from network element 150.

[0060] Advantageously, embodiments allow a user to submit a request to the network element 100 to cause the active BGP speaker to switch from route processor 110A to route processor 110B in a manner that minimizes the disruption to BGP processing by network elements 100 and 150. Further, the active BGP speaker may be switched, in embodiments, from route processor 110A to route processor 110B in a matter of seconds, as opposed to the several minutes typical of prior approaches. Also, the approaches described herein scale for use with RIBs storing a large volume of routes.

TRANSFERRING ROUTING DATA

[0061] Embodiments may perform step 230 in a variety of ways. FIG. 3 is a graphical depiction of various options for transferring routing data according to several embodiments. An embodiment may transfer routing data in step 230 of FIG. 2 by performing any of the options depicted in FIG. 3. The options of FIG. 3 are not meant to depict all the ways in which embodiments may transfer routing data from one BGP speaker to another BGP speaker, as other ways, not depicted in FIG. 3, may be employed. Further, as explained below, the options depicted in FIG. 3 are not meant to be mutually exclusive, e.g., the performance of step 340 may also involve the performance of step 310 or step 330.

[0062] According to option 310, routing data is transferred by a direct memory transfer between route processors. For example, option 310 may be performed by route processor 110A transferring routing data to route processor 110B by direct memory transfer.

[0063] According to option 320, routing data is transferred by multi-casting received BGP messages carried in TCP segments to both the first BGP speaker and the second BGP speaker. In such an embodiment, the BGP application 130A and BGP application 130B may both listen to the same socket of TCP HA process 140A. Thus, in effect, each TCP segment received by

TCP HA process 140A is sent to BGP application 130A and to BGP application 130B, thereby allowing RIB 160A and 160B to be updated simultaneously.

[0064] According to option 330, routing data is transferring in a BGP peering session. In effect, BGP application 130A may interact with BGP application 130B using a BGP peering session as if BGP application 130B was located remotely across the Internet. In an embodiment, conventional BGP UPDATE messages may be used to transferring the routing data in a BGP peering session.

[0065] According to option 340, a portion of the routing data is transferred prior to the performance of step 210. Prior to receiving the request of step 210, a first portion of routing data may be transferred from BGP application 130A to BGP application 130B in a process referred to as a "lazy transfer." In a lazy transfer, routing data is transferred whenever the route processor 110A has available processing resources that are not otherwise utilized. A lazy transfer may be performed by a direct memory transfer (as in option 310) or using a BGP peering session (as in option 330). In this way, BGP application 130B may update RIB 160B whenever routing data is transferred to the BGP application 130B in a lazy transfer.

[0066] At the time the request of step 210 is received by the network element 100, a second portion of routing data, corresponding to all the routes of RIB 160A that have not yet been transferred in a lazy transfer, is transferred to BGP application 130B. The second portion of routing data may also be transferred using a direct memory transfer or in a BGP peering session. Once BGP application 130B receives the second portion of routing data, BGP application 130B updates RIB 160B to become synchronized to RIB 160A.

TRANSMITTING BGP KEEPALIVE MESSAGES TO PEERS

[0067] In an embodiment, the active BGP speaker continues to send one or more BGP KEEPALIVE messages to peers during the sending and transferring steps. For example, in an embodiment, BGP application 130A continues to send one or more BGP KEEPALIVE messages to network element 150 during steps 320 and 330.

[0068] In such an embodiment, a BGP KEEPALIVE message is sent to a particular peer after TCP connection data, that describes the state of a TCP connection to the particular peer, is transferred from the first route processor to the second route processor. Since the TCP state is frozen at route processor 110A when TCP connection data is transferred from route processor 110A to route processor 110B, a BGP KEEPALIVE message may only be sent to a peer after the TCP connection data, for that peer, is transferred. To illustrate, after TCP connection data, for a TCP session associated with network element 150, is transferred from BGP application

130A to BGP application 130B in step 230, network element 100 may send a BGP KEEPALIVE message to network element 150.

[0069] In this way, network element 150 may be informed that network element 100 is still operational. Thus, when network element 150 receives the BGP KEEPALIVE message from network element 100, network element 150 maintains a BGP connection to network element 100 during the time when the standby route processor of network element 100 is restarting as the active route processor of network element 100.

IMPLEMENTING MECHANISMS

[0070] In an embodiment, network elements 100 and 150, as well as console 120, may each be implemented on a computer system. FIG. 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a magnetic disk or optical disk, is provided and coupled to bus 402 for storing information and instructions.

[0071] Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0072] The invention is related to the use of computer system 400 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another machine-readable medium, such as storage device

410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0073] The term “machine-readable medium” as used herein refers to any medium that participates in providing data that causes a machine to operation in a specific fashion. In an embodiment implemented using computer system 400, various machine-readable media are involved, for example, in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic, light, or electromagnetic waves, such as those generated during radio-wave and infra-red data communications. All such media must be tangible to enable the instructions carried by the media to be detected by a physical mechanism that reads the instructions into a machine.

[0074] Common forms of machine-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, an EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0075] Various forms of machine-readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

[0076] Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0077] Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

[0078] Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 418. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418.

[0079] The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

[0080] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method, comprising:

receiving a request to switch a designation of an active Border Gateway Protocol (BGP) speaker of a network element from a first BGP speaker of the network element to a second BGP speaker of the network element, wherein the active BGP speaker processes all BGP messages received at the network element;

in response to receiving the request, pausing operation of a transport for BGP to the network element;

transferring routing data, which describes a state of a first routing information base (RIB), from the first BGP speaker to the second BGP speaker, wherein a second routing information base (RIB), maintained by the second BGP speaker becomes synchronized to the first routing information base (RIB);

instructing the second BGP speaker to become the active BGP speaker; and

after the second BGP speaker becomes the active BGP speaker, resuming operation of the transport to the network element.

2. The method of Claim 1, wherein the first BGP speaker shuts down in response to the first BGP speaker receiving a message, from the second BGP speaker, indicating that the second BGP speaker became the active BGP speaker.

3. The method of Claim 1, wherein the routing data is transferred by a direct memory transfer.

4. The method of Claim 1, wherein transferring the routing data comprises:

prior to receiving the request, transferring first data that identifies a first portion of routes in the first routing information base (RIB) to the second BGP speaker when the first BGP speaker has available processing resources which are not otherwise utilized; and

after receiving the request, transferring second data that identifies all remaining routes, in the first routing information base (RIB) that are not identified in the first data, to the second BGP speaker in a BGP peering session.

5. The method of Claim 1, wherein the routing data is transferred by multi-casting received BGP messages carried in TCP segments to both the first BGP speaker and the second BGP speaker.

6. The method of Claim 1, wherein the step of transferring the routing data is performed, at least in part, in a BGP peering session established between the first BGP speaker and the second BGP speaker.

7. The method of Claim 1, wherein each of the first BGP speaker and the second BGP speaker is associated with a different processor of the network element.

8. The method of Claim 1, further comprising:
transferring, from the first BGP speaker to the second BGP speaker, TCP connection data that describes the state of TCP connections maintained by the first BGP speaker.

9. The method of Claim 8, wherein the active BGP speaker continues to send one or more BGP KEEPALIVE messages to a peer during the steps of pausing operation of the transport for BGP and transferring the routing data.

10. The method of Claim 9, wherein each BGP KEEPALIVE message, of the one or more BGP KEEPALIVE messages, is sent to a particular peer after TCP connection data, that describes the state of a TCP connection to the particular peer, is transferred from the first BGP speaker to the second BGP speaker.

11. The method of Claim 1, wherein the step of pausing operation of the transport comprises ceasing to advance a window size advertised by TCP segments sent from the network element to a peer.

12. The method of Claim 1, wherein the step of pausing operation of the transport comprises ceasing to acknowledge TCP segments received, by the network element, from a peer.

13. The method of Claim 1, wherein the step of resuming operation of the transport comprises acknowledging TCP segments received, by the network element, from a peer.

14. The method of Claim 1, wherein the step of pausing operation of the transport comprises sending, to a peer, a TCP segment that advertises a window size of zero.

15. The method of Claim 1, wherein the step of resuming operation of the transport comprises sending, to the peer, a TCP segment that advertises a positive window size that reflects an amount of data that the second BGP speaker is capable of receiving from a peer.

16. An apparatus, comprising:

means for receiving a request to switch a designation of an active Border Gateway Protocol (BGP) speaker of a network element from a first BGP speaker of the network element to a second BGP speaker of the network element, wherein the active BGP speaker processes all BGP messages received at the network element;

means for pausing operation of a transport for BGP to the network element in response to receiving the request;

means for transferring routing data, which describes a state of a first routing information base (RIB), from the first BGP speaker to the second BGP speaker, wherein a second routing information base (RIB), maintained by the second BGP speaker becomes synchronized to the first routing information base (RIB);

means for instructing the second BGP speaker to become the active BGP speaker; and

means for resuming operation of the transport to the network element after the second BGP speaker becomes the active BGP speaker.

17. The apparatus of Claim 16, wherein the first BGP speaker shuts down in response to the first BGP speaker receiving a message, from the second BGP speaker, indicating that the second BGP speaker became the active BGP speaker.

18. The apparatus of Claim 16, wherein the routing data is transferred by a direct memory transfer.

19. The apparatus of Claim 16, wherein the means for transferring the routing data comprises:

mean for, prior to receiving the request, transferring first data that identifies a first portion of routes in the first routing information base (RIB) to the second BGP speaker when the first BGP speaker has available processing resources which are not otherwise utilized; and

means for, after receiving the request, transferring second data that identifies all remaining routes, in the first routing information base (RIB) that are not identified in the first data, to the second BGP speaker in a BGP peering session.

20. The apparatus of Claim 16, wherein the means for transmitted routing data comprises means for multi-casting received BGP messages carried in TCP segments to both the first BGP speaker and the second BGP speaker.

21. The apparatus of Claim 16, wherein the means for transferring the routing data comprises means for transferring the routing data in a BGP peering session established between the first BGP speaker and the second BGP speaker.

22. The apparatus of Claim 16, wherein each of the first BGP speaker and the second BGP speaker is associated with a different processor of the network element.

23. The apparatus of Claim 16, further comprising:
means for transferring, from the first BGP speaker to the second BGP speaker, TCP connection data that describes the state of TCP connections maintained by the first BGP speaker.

24. The apparatus of Claim 23, further comprising means for the active BGP speaker to continue to send one or more BGP KEEPALIVE messages to a peer while pausing operation of the transport and transferring the routing data.

25. The apparatus of Claim 24, wherein each BGP KEEPALIVE message, of the one or more BGP KEEPALIVE messages, is sent to a particular peer after TCP connection data, that describes the state of a TCP connection to the particular peer, is transferred from the first BGP speaker to the second BGP speaker.

26. The apparatus of Claim 16, wherein the means for pausing operation of the transport comprises means for ceasing to advance a window size advertised by TCP segments sent from the network element to a peer.

27. The apparatus of Claim 16, wherein the means for pausing operation of the transport comprises means for ceasing to acknowledge TCP segments received, by the network element, from a peer.

28. The apparatus of Claim 16, wherein the means for resuming operation of the transport comprises means for acknowledging TCP segments received, by the network element, from a peer.

29. The apparatus of Claim 16, wherein the means for pausing operation of the transport comprises means for sending, to a peer, a TCP segment that advertises a window size of zero.

30. The apparatus of Claim 16, wherein the means for resuming operation of the transport comprises means for sending, to a peer, a TCP segment that advertises a positive

window size that reflects an amount of data that the second BGP speaker is capable of receiving from the peer.

31. An apparatus, comprising:

a network interface that is coupled to the data network for receiving one or more packet flows therefrom;

a processor; and

a machine-readable medium carrying one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps recited in any one of Claims 1-15.

FIG. 1

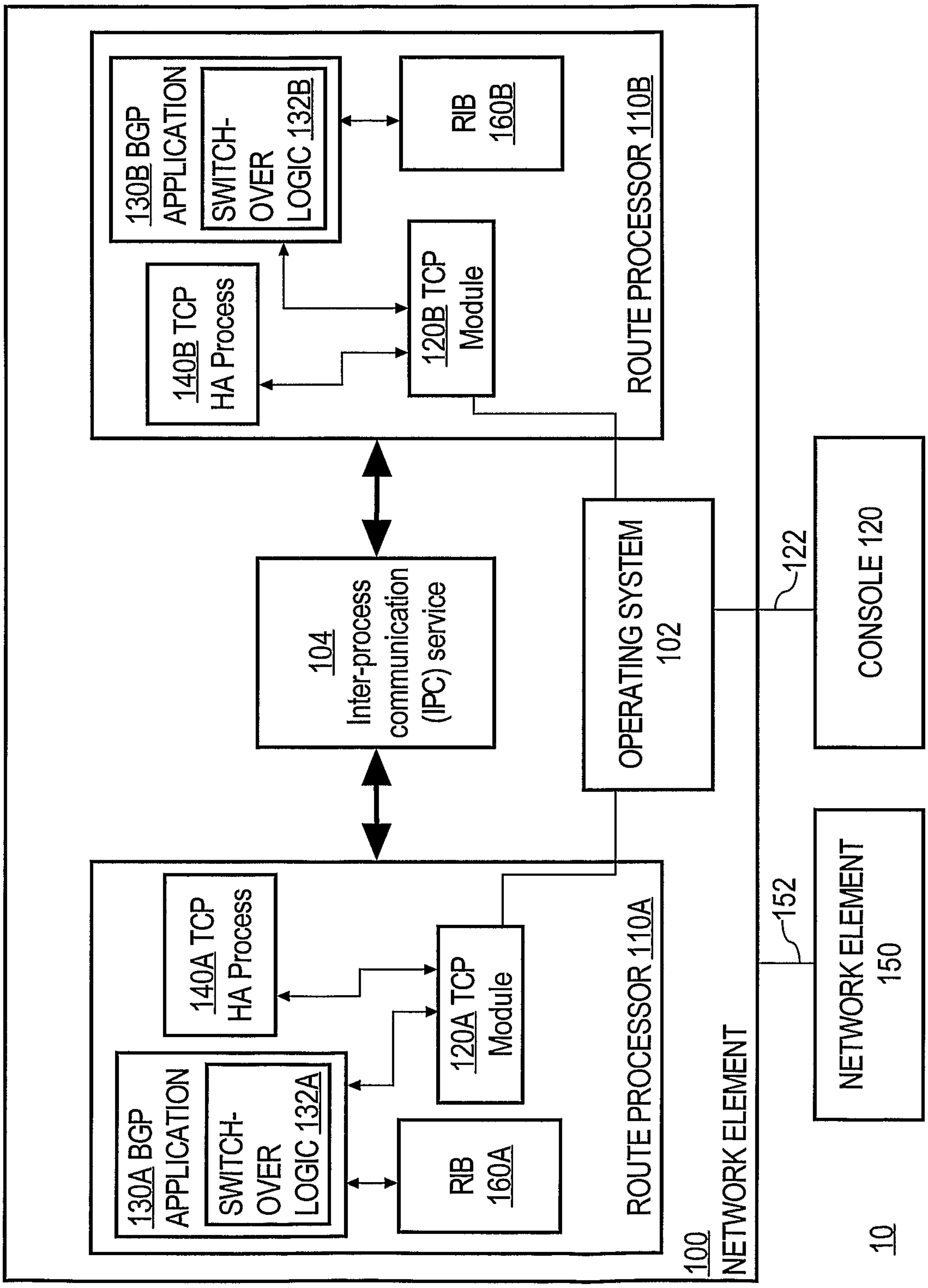

130B BGP APPLICATION
SWITCH-OVER LOGIC 132B
RIB 160B
140B TCP HA Process
120B TCP Module
ROUTE PROCESSOR 110B
104 Inter-process communication (IPC) service
OPERATING SYSTEM 102
122
CONSOLE 120
140A TCP HA Process
120A TCP Module
130A BGP APPLICATION
SWITCH-OVER LOGIC 132A
RIB 160A
ROUTE PROCESSOR 110A
100 NETWORK ELEMENT
152
NETWORK ELEMENT 150
10

FIG. 2

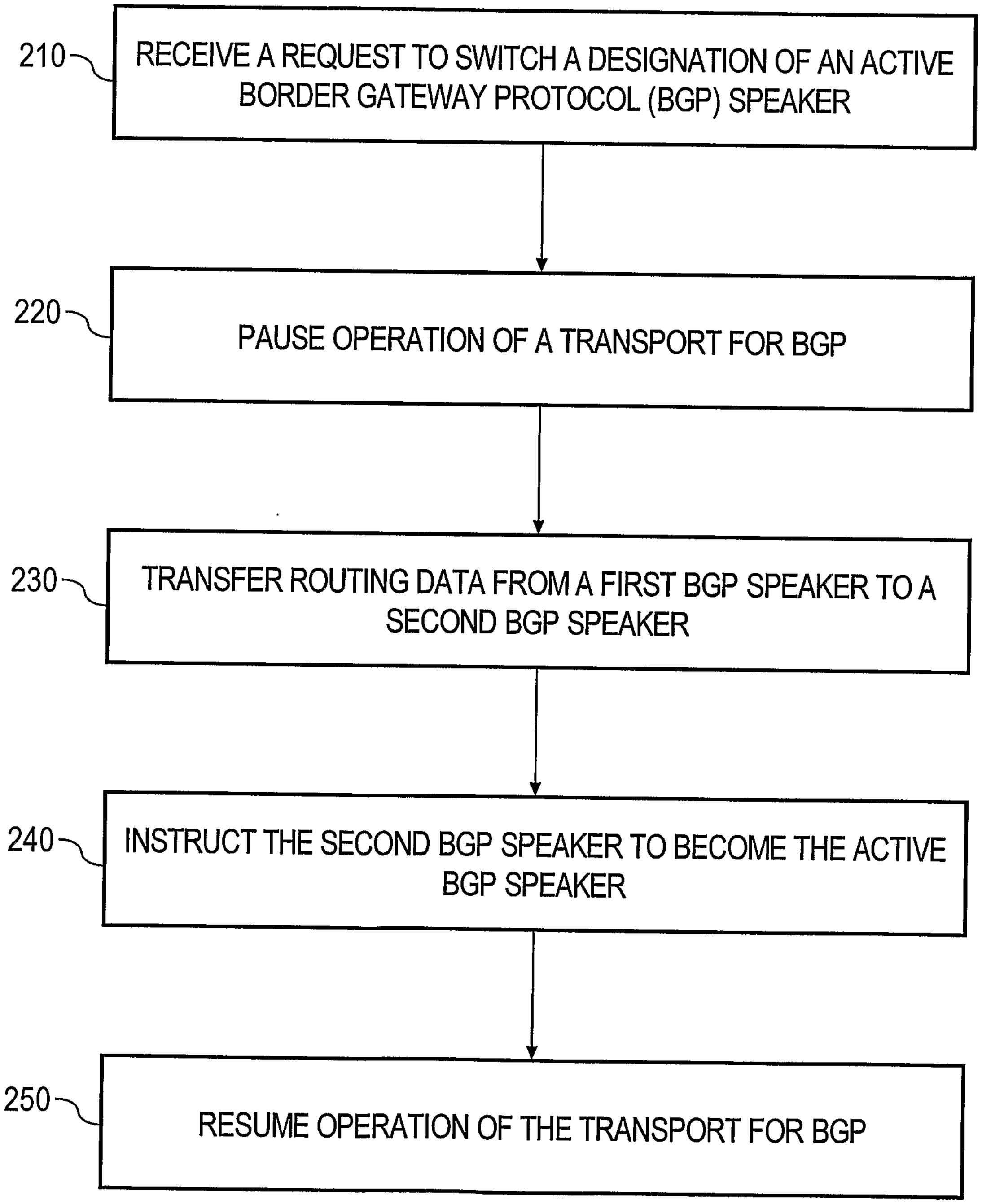

210
RECEIVE A REQUEST TO SWITCH A DESIGNATION OF AN ACTIVE BORDER GATEWAY PROTOCOL (BGP) SPEAKER
220
PAUSE OPERATION OF A TRANSPORT FOR BGP
230
TRANSFER ROUTING DATA FROM A FIRST BGP SPEAKER TO A SECOND BGP SPEAKER
240
INSTRUCT THE SECOND BGP SPEAKER TO BECOME THE ACTIVE BGP SPEAKER
250
RESUME OPERATION OF THE TRANSPORT FOR BGP

FIG. 3

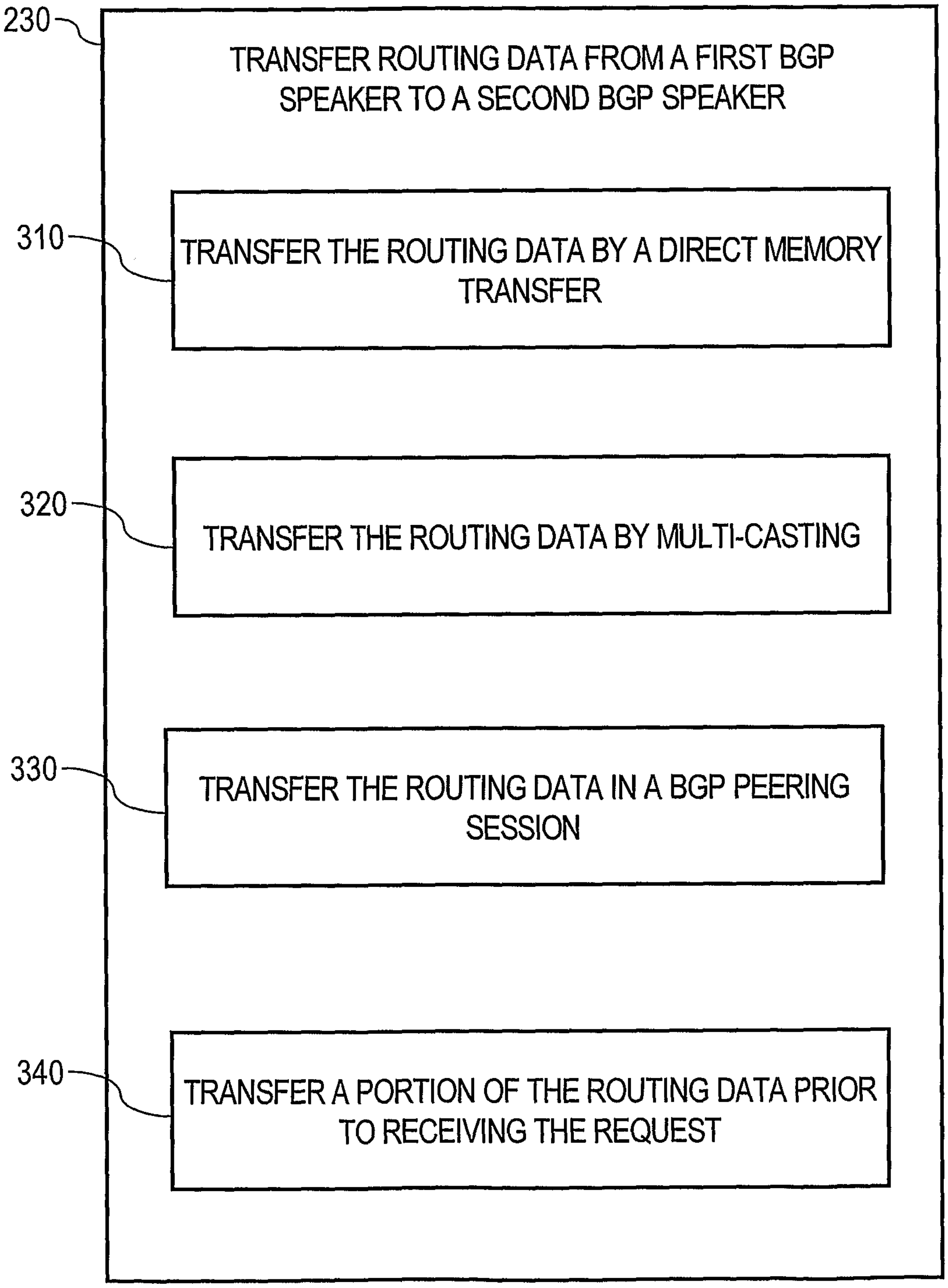
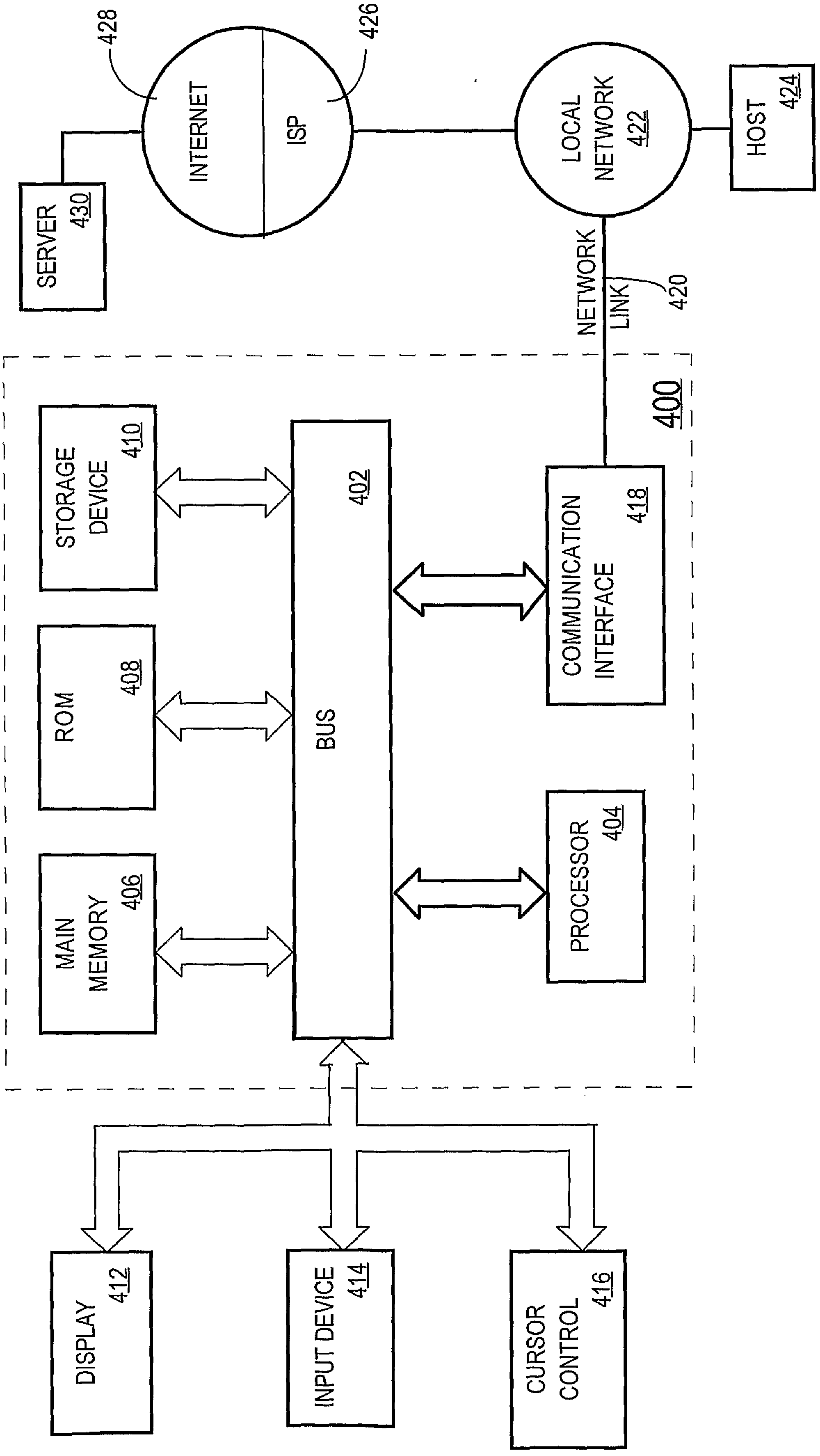

230
TRANSFER ROUTING DATA FROM A FIRST BGP SPEAKER TO A SECOND BGP SPEAKER
310
TRANSFER THE ROUTING DATA BY A DIRECT MEMORY TRANSFER
320
TRANSFER THE ROUTING DATA BY MULTI-CASTING
330
TRANSFER THE ROUTING DATA IN A BGP PEERING SESSION
340
TRANSFER A PORTION OF THE ROUTING DATA PRIOR TO RECEIVING THE REQUEST

FIG. 4


SERVER 430
INTERNET
428
ISP
426
LOCAL NETWORK 422
HOST 424
NETWORK LINK
420
400
STORAGE DEVICE 410
ROM 408
MAIN MEMORY 406
BUS
402
COMMUNICATION INTERFACE 418
PROCESSOR 404
DISPLAY 412
INPUT DEVICE 414
CURSOR CONTROL 416