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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 November 2005 (03.11.2005)

PCT

(10) International Publication Number
WO 2005/104176 A2

(51) International Patent Classification⁷: H01J 49/00

(21) International Application Number: PCT/US2005/013716

(22) International Filing Date: 22 April 2005 (22.04.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/830,779 23 April 2004 (23.04.2004) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **LIU, Zhen** [FR/US]; 37 Roundabend Road, Tarrytown, New York 10591 (US). **SAHU, Sambit** [IN/US]; 551 Kennicut Hill Road, Mahopac, New York 10541 (US). **SILBER, Jeremy I.** [US/US]; 237 West 109th Street, Apt. 3C, New York, New York 10025 (US).

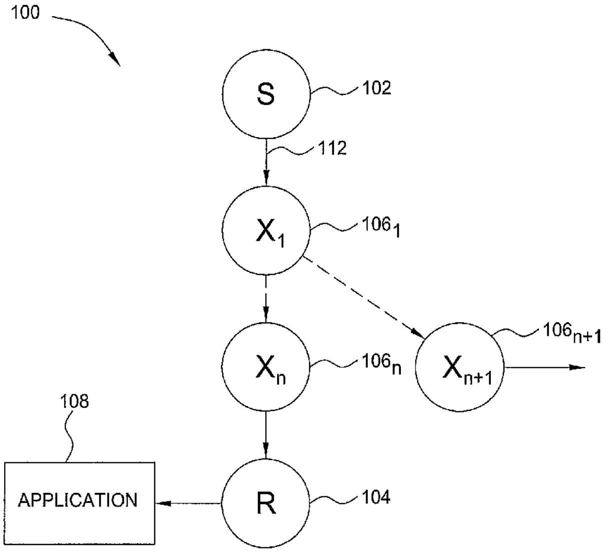
(74) **Agent: TONG, Kin-Wah**; 595 Shrewsbury Avenue, Shrewsbury, New Jersey 07702 (US).

(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, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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, 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).

[Continued on next page]

(54) **Title:** METHOD AND APPARATUS FOR FAILURE RESILIENT FORWARDING OF DATA OVER A COMPUTER NETWORK



(57) **Abstract:** In one embodiment, the present invention is a method and an apparatus for failure-resilient forwarding of data over a computer network. In one embodiment, a marker is introduced into the data stream, e.g., at the sending node, and allows, in turn, forwarding nodes and/or receivers to efficiently track data stream reception. The marker functions as a checkpoint for the data transport process, and is identified and indexed at each forwarding node and receiver. Each receiver saves the marker prior to delivering data to an application, thereby designating a point in the data stream at which all preceding data is confirmed to have been delivered to the application. Thus, if a forwarding node fails, the receiver may request stream data from an alternate forwarding node by specifying to the alternate forwarding node to provide data starting from the marker.

WO 2005/104176 A2

Published:

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

METHOD AND APPARATUS FOR FAILURE RESILIENT FORWARDING OF DATA OVER A COMPUTER NETWORK

BACKGROUND

[0001] The present invention relates generally to computer systems and computer networks, and relates more particularly to content delivery over computer networks. Specifically, the present invention relates to a method and apparatus for adaptive forwarding of data over a computer network.

[0002] Figure 1 is a schematic illustration of one embodiment of a system 100 for forwarding data over a network. A wide range of end-to-end computing applications (including overlay networks, end-system multicast, proxy servers, network address translation and protocol tunneling, among others) use intermediaries, or forwarding nodes $106_1 - 106_n$ (*e.g.*, computing devices or routers), to route a stream 112 of data from a sender 102 (*e.g.*, a server) to one or more receivers 104. Receivers 104 may in turn deliver the data to one or more computing applications *108*.

[0003] A typical problem with a system such as the system 100 is that a failure or disruption at any forwarding node disrupts the end-to-end chain, resulting in incomplete data delivery to the receiver(s). This is especially troublesome for large networks, as the probability of node failure increases with the number of forwarding nodes implemented. Conventional solutions for addressing node failure in a forwarding network include source-based repair such as a Transmission Control Protocol/Internet Protocol (TCP/IP) session between the data source and the receiver, packet number-based retransmission requests, and various application- and content-specific resiliency schemes (*e.g.*, resuming File Transport Protocol at a specific byte offset from the start of a file, or resuming a video transmission at a specific frame number). However, these conventional solutions are subject to a number of limitations, including scalability limitations and the inability to adapt for use over a network using heterogeneous transports or delivering generic (non-content-specific) data streams. Accordingly, they are not appropriate for delivering data over an adaptively changing network using multiple point-to-point protocols in a content-independent manner.

[0004] Thus, there is a need for a method and apparatus for failure-resilient forwarding of data over a computer network.

SUMMARY OF THE INVENTION

[0005] In one embodiment, the present invention is a method and an apparatus for failure-resilient forwarding of data over a computer network. In one embodiment, a marker is introduced into the data stream, *e.g.*, at the sending node and, in turn, allows forwarding nodes and/or receivers to efficiently track data stream reception. The marker functions as a checkpoint for the data transport process, and is identified and indexed at each forwarding node and receiver. Each receiver saves the marker prior to delivering data to an application, thereby designating a point in the data stream at which all preceding data is confirmed to have been delivered to the application. Thus, if a forwarding node fails, the receiver may request stream data from an alternate forwarding node by specifying to the alternate forwarding node to provide data starting from the marker.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] So that the manner in which the above recited embodiments of the invention are attained and can be understood in detail, a more particular description of the invention, briefly summarized above, may be had by reference to the embodiments thereof which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of this invention and are therefore not to be considered limiting of its scope, for the invention may admit to other equally effective embodiments.

[0007] Figure 1 is a schematic illustration of one embodiment of an end-to-end computing network;

[0008] Figure 2 is a flow diagram illustrating one embodiment of a method for enabling failure-resilient forwarding of data from a sender to one or more receivers according to the present invention;

[0009] Figure 3 is a table illustrating one method of distributing content using the system illustrated in Figure 2;

[0010] Figure 4 is a flow diagram illustrating one embodiment of a method for recovering lost data in a data stream; and

[0011] Figure 5 is a high level block diagram of the present failure-resilient forwarding system that is implemented using a general purpose computing device.

[0012] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION

[0013] The present invention is a method and an apparatus for failure-resilient forwarding of data over a computer network. In one embodiment, a trigger condition, such as forwarding node failure, performance degradation, query, resource use imbalance and the like, initiates a network adaptation to correctly resume transmission reception of a data stream.

[0014] Figure 2 is a flow diagram illustrating the flow of data through one embodiment of a method 200 for enabling failure-resilient forwarding of data from a sender to one or more receivers according to the present invention. The method 200 is initialized at step 202 and proceeds to step 204, where a sending node or a forwarding node (*e.g.*, sender 102 or any one of the forwarding nodes 106 of FIG. 1) obtains (in the case of the sender) or receives (in the case of the forwarding node) at least a portion of a data stream. In one embodiment, the data stream may simply be a portion or an arbitrarily sized data segment of a much larger data stream. Namely, a sequence of "chunks" or "portions" of the larger data stream is being sent from the sending node to a receiving node. In step 206, the sender or a forwarding node injects a marker into the portion of the data stream, and forwards the "marked" data stream, with or without further modification, to one or more next recipient nodes in the network (*e.g.*, one or more subsequent forwarding nodes or receiving nodes) via a point-to-point reliable transport protocol (*e.g.*, a protocol that is packet loss resilient, such as TCP/IP and the like). The marker designates a reference point in the generic data stream and in one embodiment is a recognizable bit field with a unique identifier. The marker may be recognized by reserved bit sequences, fixed inter-marker offsets, or an offset specified by a prior marker. Thus, markers may be periodically injected into the data stream, so that a plurality of marked data streams is transported through the network.

[0015] After injecting the marker in step 206, the method 200 branches off into at least one of two possible subsequent processes. In steps 208 – 210, the method executes steps in accordance with the function of a forwarding node; in steps 209-214, the method 200 executes steps in accordance with the function of a receiving node.

[0016] In step 208, the method 200 inquires if the recipient of the marked data stream is a forwarding node. If the method 200 determines that the recipient is forwarding node, the method 200 proceeds to step 210, where the method 200 inspects the received data stream, stores the data in a local buffer of the forwarding node, and creates or updates a marker index at the forwarding node. In one embodiment, the marker index that the method 200 updates comprises two key components: (1) a record of the most recently received marker; and (2) a record of each marker previously received and stored by the forwarding node. Once the method 200 has updated the marker index, the method 200 forwards the marked data stream to the next recipient(s) (*e.g.*, one or more other forwarding nodes or receivers) in the network. The marked data stream is processed by the next recipient node(s) starting at the point in the method 200 just following step 206, as indicated by the loop from step 210. Thus, all forwarding nodes receive the marked data stream, relay the marked data stream to the next forwarding nodes or receivers, and index the markers.

[0017] Figure 3 is a schematic illustration of one embodiment of a marker index 300 according to the present invention, such as the marker index updated by the method 200 in step 210 of Figure 2. In one embodiment, the marker index 300 is a table. As illustrated, the marker index 300 stores, for each marker (*e.g.*, markers M_1 – M_3), the marker's unique identifier and its position in the local buffer. As will be further described below with reference to Figure 4, this stored information may be used to recover data lost, for example, due to a forwarding node failure.

[0018] Referring back to Figure 2, if the method 200 concludes at step 208 that the recipient of the marked data stream is not a forwarding node, the method 200 terminates.

[0019] Also after injecting the marker in step 206, the method 200 inquires in step 209 if the recipient is a receiving node. If the method 200 concludes that the recipient is a receiving node, the method 200 proceeds to step 212 and queues the stream data received by the receiver until the marker is encountered. In step 214, the method 200 saves the marker and delivers the queued data (*i.e.*, all undelivered, non-marked data preceding the marker in the data stream) to a process desiring the original data stream (*e.g.*, an application or a storage process). Alternatively, if the method 200 concludes in step 208 that the recipient is not a receiving node, the method 200 terminates.

[0020] In one embodiment, one or more nodes are both forwarding and receiving nodes. That is, a node may be adapted to both receive data for delivery to an application, and also to forward the received data on to another node. Thus, the node is capable of executing both the forwarding and the receiving methods contained within the method 200. Thus, although the forwarding and receiving processes (*e.g.*, steps 208-210 and 209-214, respectively) are designated by sequential reference numerals, the reference numerals do not connote an order in which the processes occur. Therefore, those skilled in the art will appreciate that the forwarding and receiving methods are executed independently, and that the methods may actually occur simultaneously, or may occur one after the other in any order. Thus, the sequence of the reference numerals as they apply to steps 208-216 is not intended to be limiting in any sense.

[0021] Thus, the markers injected into the data stream represent checkpoints for the data transport process. By saving the markers at the receivers, the method 200 designates points in the data stream where all preceding data has been delivered, reliably and in order, to the waiting application. The method 200 also serves the function of designating points in the data stream where any succeeding data has yet to be delivered. This saved marker information may be used to recover data lost, for example, due to a forwarding node failure.

[0022] Figure 4 is a flow diagram illustrating one embodiment of a method 400 for recovering lost data in a data stream. For example, the method 400 may be executed in the event that a forwarding node (*e.g.*, a forwarding node 106 of Figure 1) fails (*e.g.*, due to disconnection from the network or power failure) and thus ceases to forward data to subsequent recipients. The method 400 is initialized at step 402 and proceeds to step 404, where the method 400 identifies a forwarding node failure and connects a receiver (or subsequent forwarding node) to an alternate forwarding node, or a "backup node" (*e.g.*, a node that preceded the failed node in the routing path). Alternatively, the method 400 may connect the receiver to any "sister" node of the failed forwarding node that is still receiving the data stream. In one embodiment, the backup node is selected for efficiency. For example, if the failed node is node X_n in Figure 1, then the backup node can be selected to be node X_1 or node X_{n+1}. The selection of the proper node can be based on distance, delay, computational cost and the like.

[0023] The method 400 then proceeds to step 406, where the method 400 requests, from the backup node, the stream data starting from the last marker, *M*, saved by the receiver. In an alternative embodiment, the method 400 may request the stream data starting from a specified position after the last marker *M* (*e.g.*, three bits after the marker *M*). The request includes the unique identifier for the marker *M*. In step 407, the method 400 inquires if the backup node will accept the request presented in step 406. If the backup node rejects the request, the method 400 returns to step 404 and connects to another backup node. Alternatively, if the backup node accepts the request in step 407, the method 400 enables the backup node to look up the marker *M* in the backup node's marker index. If the marker *M* is present, the backup node begins sending the marked data stream, using the location of the marker *M* in its local buffer as the starting point. In one embodiment, any data residing in the local buffer past the point of the marker *M* is discarded.

[0024] In step 408, the method 400 resets a queue "write pointer" for the receiver to a position immediately following the marker *M*. The method 400 also erases data following the write pointer in the local buffer, and the receiver will now start queuing data over the new connection from the new forwarding node. As the marked data stream arrives at the receiver over the new connection from the backup node, the arriving data stream overwrites any data following the marker *M* in the receiver's local buffer. In an alternative embodiment, the method 400 may request discrete portions of the marked data stream from multiple backup nodes.

[0025] At step 410, the method 400 inquires if the next marker, *M+1*, has arrived at the receiver. If the next marker *M+1* has arrived, the method 400 delivers data queued by the receiver (minus the marker *M*) to an application requesting the data at step 412. If the next marker *M+1* has not arrived, the method 400 continues to queue data over the new connection from the backup node. Those skilled in the art will recognize that steps 408-412 of the method 400 are steps typically executed by a receiver node; they have been discussed here, in the context of the method 400, to illustrate the method by which the receiver mode may implement such steps in conjunction with the recovery of lost data.

[0026] The method 400 is therefore able to repair failures accurately and efficiently by resuming data transmission at the point of interruption. Moreover, as the repair only requires communication with a nearby forwarding/backup node, repair paths are short and network load is fairly distributed. The method 400 also works at the application layer with any reliable point-to-point transport protocol, can leverage existing point-to-point protocols, and may allow reframing and multi-protocol forwarding. Thus, the method 400 works independently of transport protocols, as well as independently of data stream content.

[0027] Figure 5 is a high level block diagram of the present failure-resilient forwarding system that is implemented using a general purpose computing device 500. In one embodiment, a general purpose computing device 500 comprises a processor 502, a memory 504, a failure-resilient forwarding mechanism or module 505 and various input/output (I/O) devices 506 such as a display, a keyboard, a mouse, a modem, and the like. In one embodiment, at least one I/O device is a storage device (*e.g.*, a disk drive, an optical disk drive, a floppy disk drive). It should be understood that the failure-resilient forwarding mechanism 505 can be implemented as a physical device or subsystem that is coupled to a processor through a communication channel.

[0028] Alternatively, the failure-resilient forwarding mechanism 505 can be represented by one or more software applications (or even a combination of software and hardware, *e.g.*, using Application Specific Integrated Circuits (ASIC)), where the software is loaded from a storage medium (*e.g.*, I/O devices 506) and operated by the processor 502 in the memory 504 of the general purpose computing device 500. Thus, in one embodiment, the failure-resilient forwarding mechanism 505 and the associated methods described herein with reference to the preceding Figures can be stored on a computer readable medium or carrier (*e.g.*, RAM, magnetic or optical drive or diskette, and the like).

[0029] Although the methods described herein have been discussed with reference to system recovery from node failures, those skilled in the art will appreciate that the present invention may have other applications in the field of content delivery. For example, the present invention may be implemented to assure reliable data delivery with any network reconfiguration, and for any reason. Other reconfiguration techniques may include finding a backup node using a centralized or distributed registry of nodes (*e.g.*, a known server or a Domain Name Service (DNS) lookup), a distributed hash table lookup, or a broadcast search, among others. Other reasons for network reconfiguration may include responding to performance degradation, optimization of network resource utilization and load balancing, among others.

[0030] Thus, the present invention represents a significant advancement in the field of content delivery. A method and apparatus are provided that enable efficient, failure-resilient forwarding of data over a network. The network is able to accurately and efficiently resume data transmission at the point of interruption, without transmitting redundant or out-of-order data to a receiver. To an application requesting data from a sender, the failure and recovery of the system are substantially transparent. Moreover, the methods of the present invention are not application specific, but may be adapted for use with any type of data stream, regardless of content, and with any type of reliable transport protocol.

[0031] While foregoing is directed to the preferred embodiment of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

CLAIMS

1. A method for forwarding a data stream over a network from a sender node to one or more receiver nodes, the method comprising the steps of:

receiving a marked stream having a portion of the data stream and one or more markers;

storing one or more of said markers and at least a subsequent portion of said marked stream in a buffer;

forwarding said marked stream to one or more next recipient nodes; and

responding to a request for stream data stored in the buffer by initiating data transfer at a specified marker, wherein said request is for the delivery of stream data subsequent to the specified marker.

2. The method of claim 1, wherein said marker is a recognizable bit field having a unique identifier.

3. The method of claim 1, wherein said marker is injected into said portion of the data stream by the sender node.

4. The method of claim 1, wherein said marker is injected into the data stream by an intermediate node between the sender node and the one or more receiver nodes.

5. The method of claim 1, wherein the marked data stream is sent using point-to-point reliable transport protocol.

6. The method of claim 5, wherein said point reliable transport protocol is TCP/IP protocol.

7. The method of claim 1, wherein the one or more next recipient nodes are intermediate nodes between the sender node and the one or more receiver nodes, wherein the intermediate nodes are adapted for forwarding the marked data stream to at least the one or more receiver nodes.

8. The method of claim 1, wherein said step of storing said one or more markers comprises:

recording said marker in the marked data stream as a most recently received marker; and

updating a marker index, wherein the marker index records markers that were previously received.

9. The method of claim 8, wherein the marker index records, for each marker received, a unique identifier and a position in the buffer.

10. The method of claim 1, wherein the one or more next recipient nodes are the one or more receiver nodes.

11. The method of claim 1, further comprising the steps of:

delivering the portion of the data stream to an application.

12. The method of claim 11, wherein the step of delivering the portion of the data stream comprises delivering a portion of the data stream starting from one of said markers.

13. The method of claim 11, wherein the step of delivering the portion of the data stream comprises delivering a portion of the data stream starting from a point in the data stream that is after one of said markers.

14. The method of claim 1, wherein the one or more next recipient nodes are adapted for both forwarding the marked data stream and for receiving the marked data stream for delivery to one or more applications.

15. The method of claim 14, wherein the one or more next recipient nodes are adapted for performing the forwarding and receiving steps simultaneously.

16. The method of claim 14, wherein the one or more next recipient nodes are adapted for performing the forwarding and receiving steps in any order.

17. A method for resuming interrupted transport of data over a network from a sender node to a receiver node, the method comprising the steps of:

connecting to an intermediate node in the network; and

requesting data from the intermediate node starting from a last marker saved by the receiver node.

18. The method of claim 17, further comprising the step of:

resetting a receiver queue write pointer to a position immediately following the last saved marker; and

queuing data from the intermediate node to the receiver node.

19. The method of claim 17, further comprising the step of:

delivering data queued by the receiver when a next marker is received from the intermediate node, wherein the queued data is delivered to an application.

20. The method of claim 17, wherein the step of requesting data from the intermediate node comprises the steps of:

specifying a unique identifier for the marker; and

looking up the marker in a local buffer of the intermediate node.

21. The method of claim 20, further comprising the step of:

discarding any data in the local buffer received after the marker.

22. The method of claim 17, further comprising the step of:

overwriting data following the marker in a local buffer of the receiver node, wherein the data is overwritten with the data being received from the intermediate node.

23. A system for forwarding data over a network, comprising:

means for sending stream data;

means for receiving the stream data;

means for designating and tracking reference points in the stream data; and

means for receiving the stream data from an alternate sending means by specifying a point at which the alternate sending means should begin sending the stream data,

24. A computer readable medium containing an executable program for forwarding a data stream over a network from a sender node to one or more receiver nodes, where the program performs the steps of:

receiving a marked stream having a portion of the data stream and one or more markers;

storing said one or more markers and said marked stream in a buffer; and

forwarding said marked stream to one or more next recipient nodes; and

responding to a request for stream data stored in the buffer by initiating data transfer at a specified marker, wherein said request is for the delivery of stream data subsequent to the specified marker.

25. The computer readable medium of claim 24, wherein said marker is a recognizable bit field having a unique identifier.

26. The computer readable medium of claim 24, wherein the one or more next recipient node are intermediate nodes between the sender node and the one or more receiver nodes, wherein the intermediate nodes are adapted for forwarding the marked data stream to the one or more receiver nodes.

27. The computer readable medium of claim 24, wherein said step of storing said one or more markers comprises:

recording said marker in the marked data stream as a most recently received marker; and updating a marker index, wherein the marker index records markers that were previously received.

28. The computer readable medium of claim 27, wherein the marker index records, for each marker received, a unique identifier and a position in the buffer.

29. The computer readable medium of claim 24, wherein the one or more next recipient nodes are the one or more receiver nodes.

30. The computer readable medium of claim 24, wherein the one or more next recipient nodes are adapted for both forwarding the marked data stream and for receiving the marked data stream for delivery to one or more applications.

31. The computer readable medium of claim 30, wherein the one or more next recipient nodes are adapted for performing the forwarding and receiving steps simultaneously.

32. The computer readable medium of claim 30 wherein the one or more next recipient nodes are adapted for performing the forwarding and receiving steps in any order.

33. A computer readable medium containing an executable program for resuming interrupted transport of data over a network from a sender node to a receiver node, where the program performs the steps of:

connecting to an intermediate node in the network; and

requesting data from the intermediate node starting from a last marker saved by the receiver node.

34. The computer readable medium of claim 33, further comprising the step of:

resetting a receiver queue write pointer to a position immediately following the last saved marker; and

queuing data from the intermediate node to the receiver.

35. The computer readable medium of claim 33, further comprising the step of:

delivering data queued by the receiver when a next marker is received from the intermediate node, wherein the queued data is delivered to an application.

36. The computer readable medium of claim 33, wherein the step of requesting data from the intermediate node comprises:

specifying a unique identifier for the marker; and

looking up the marker in a local buffer of the intermediate node.

37. A method for providing content delivery optimization for a network over which a data stream is transported from a sender node to one or more receiver nodes, the method comprising the steps of:

injecting one or more markers into the data stream to create a marked stream comprising a portion of the data stream and said one or more markers;

storing said marked data stream and said marker;

receiving a data request from at least one of the one or more receiver nodes soliciting a portion of the data stream identified by a location of the marker; and

forwarding a portion of the data stream, starting from the location specified by the data request.

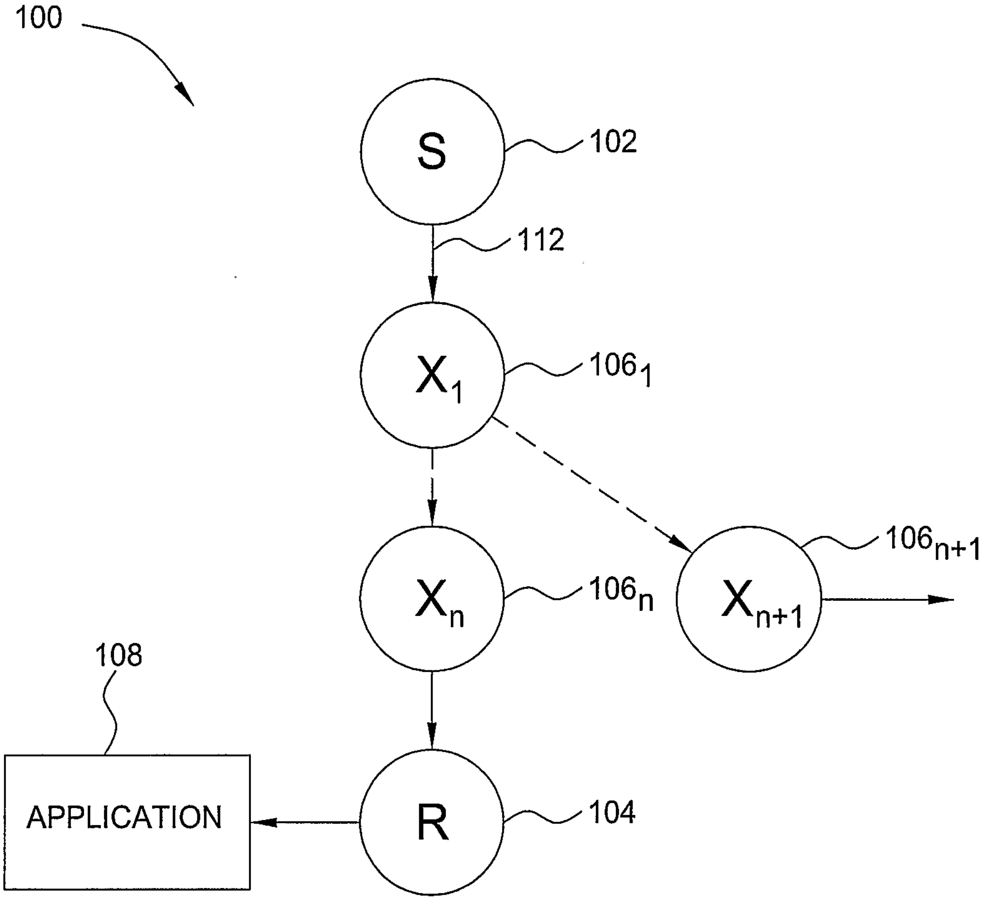


FIG. 1

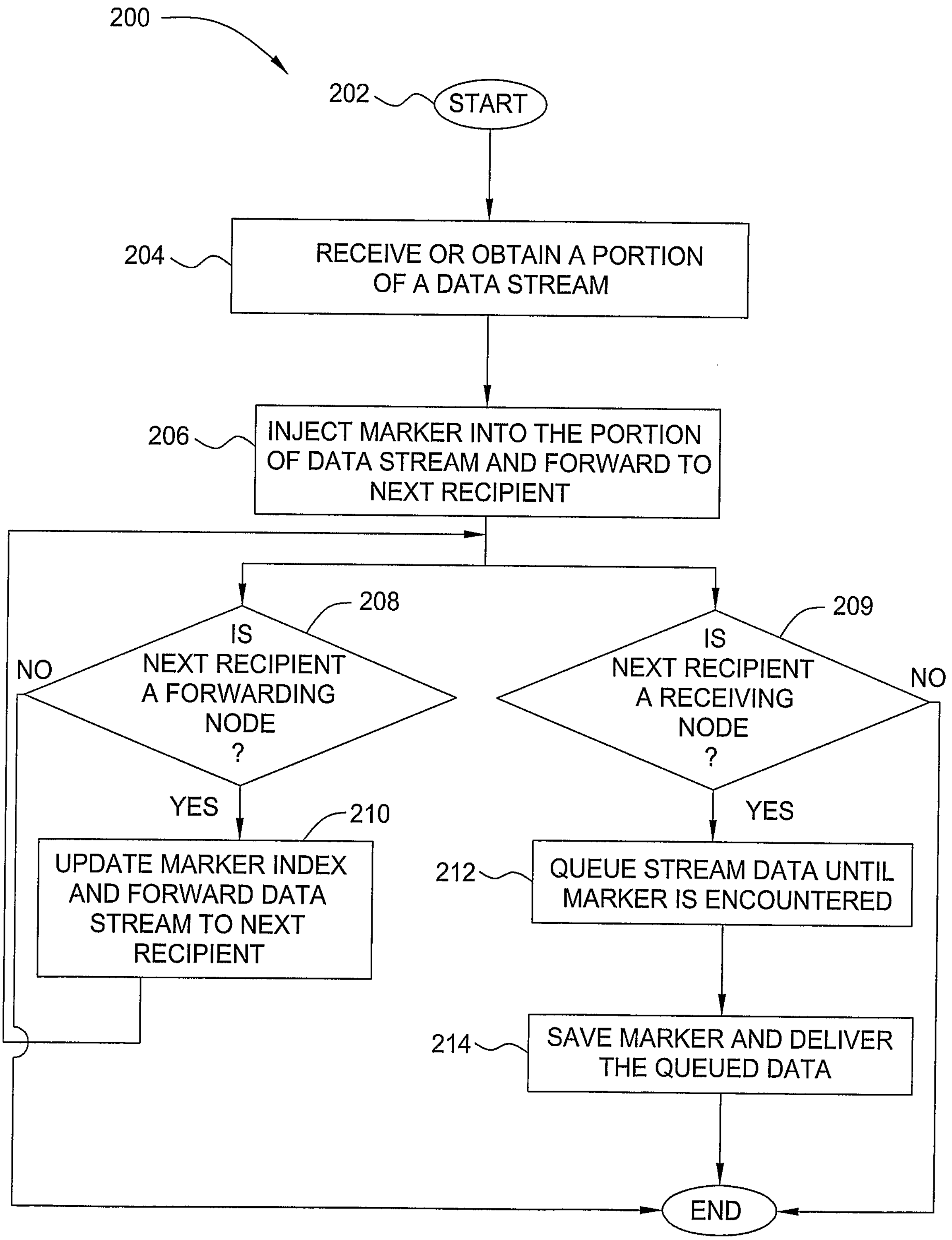

200
202
START
204
RECEIVE OR OBTAIN A PORTION OF A DATA STREAM
206
INJECT MARKER INTO THE PORTION OF DATA STREAM AND FORWARD TO NEXT RECIPIENT
208
IS NEXT RECIPIENT A FORWARDING NODE ?
NO
YES
210
UPDATE MARKER INDEX AND FORWARD DATA STREAM TO NEXT RECIPIENT
209
IS NEXT RECIPIENT A RECEIVING NODE ?
NO
YES
212
QUEUE STREAM DATA UNTIL MARKER IS ENCOUNTERED
214
SAVE MARKER AND DELIVER THE QUEUED DATA
END

FIG. 2

300

MARKER	IDENTIFIER	LOCAL BUFFER POSITION
M_1	1	L_1
M_2	2	L_2
M_3	3	L_3

FIG. 3

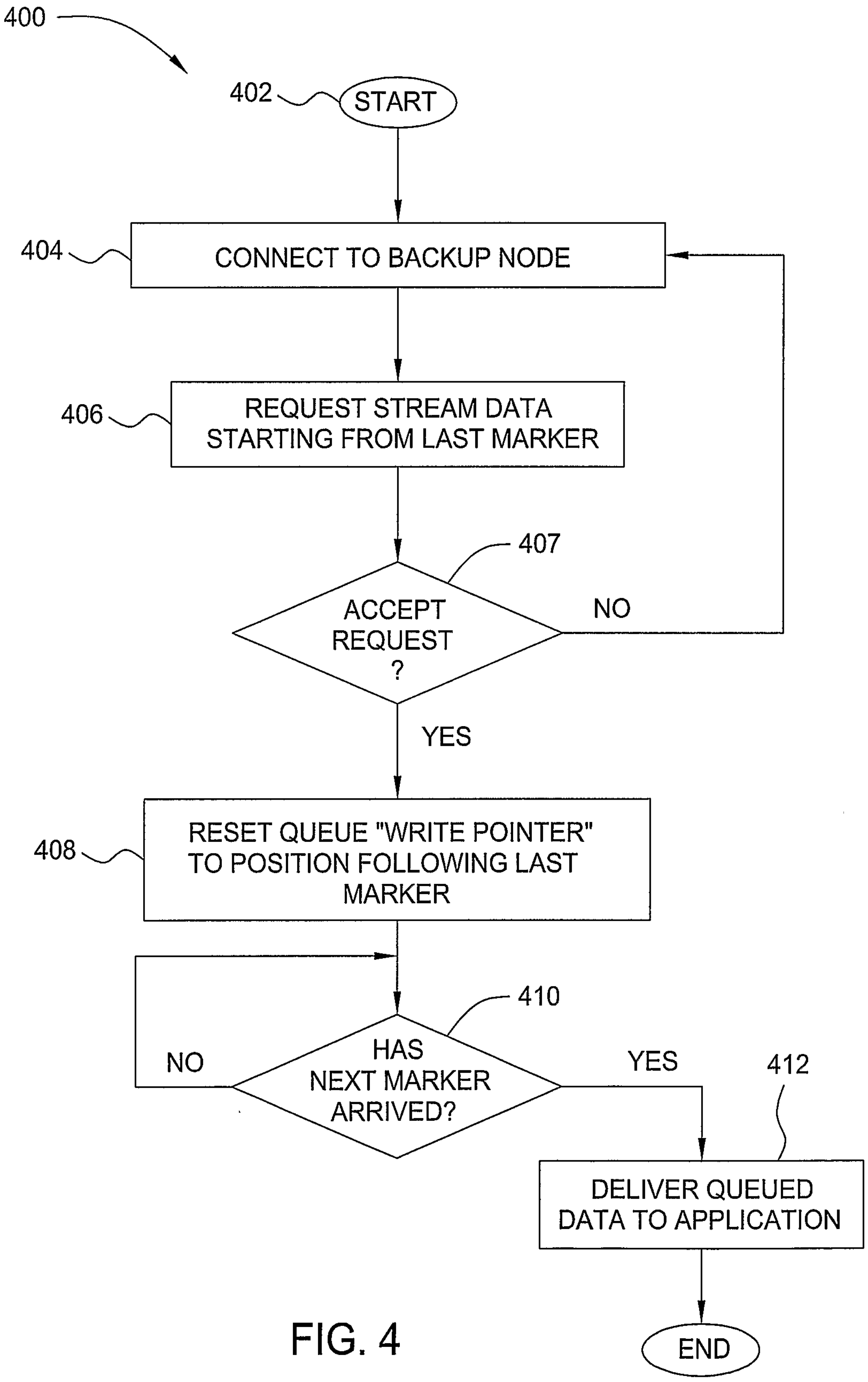

400
402
START
404
CONNECT TO BACKUP NODE
406
REQUEST STREAM DATA STARTING FROM LAST MARKER
407
ACCEPT REQUEST ?
NO
YES
408
RESET QUEUE "WRITE POINTER" TO POSITION FOLLOWING LAST MARKER
410
HAS NEXT MARKER ARRIVED?
NO
YES
412
DELIVER QUEUED DATA TO APPLICATION
END

FIG. 4

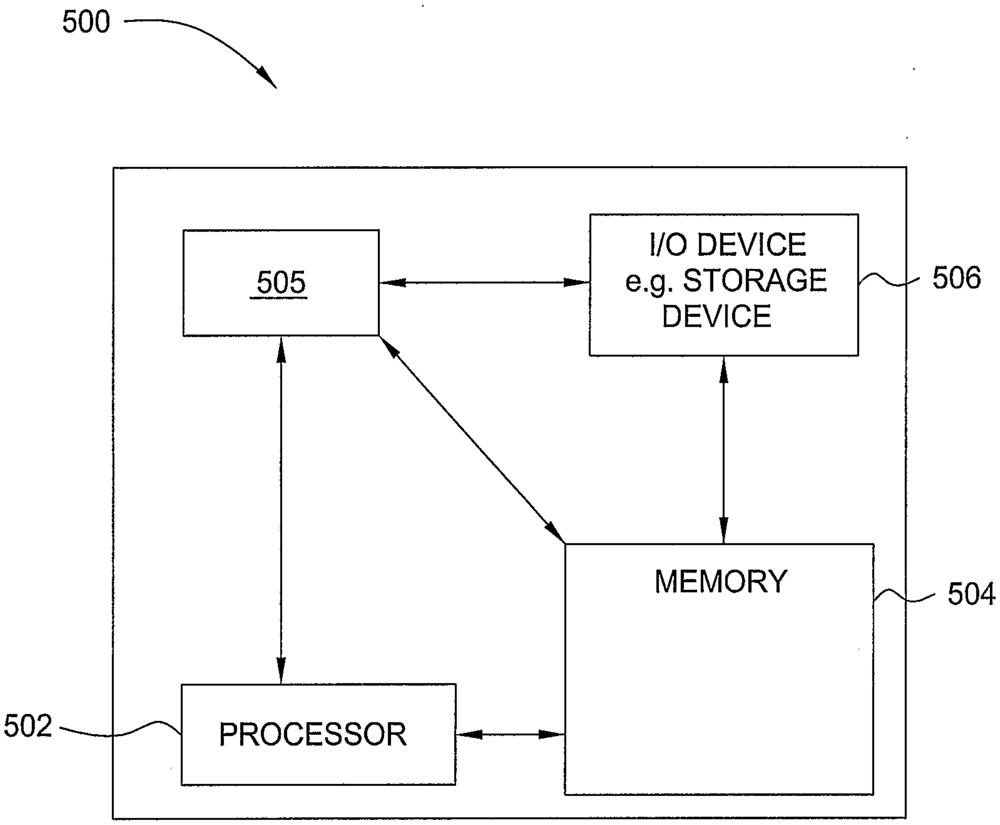

500
505
I/O DEVICE e.g. STORAGE DEVICE
506
MEMORY
504
502
PROCESSOR

FIG. 5