



(43) International Publication Date
26 May 2006 (26.05.2006)

PCT

(10) International Publication Number
WO 2006/055832 A2

(51) International Patent Classification:
H04L 12/26 (2006.01)

(21) International Application Number:

PCT/US2005/041947

(22) International Filing Date:

18 November 2005 (18.11.2005)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

60/629,331 19 November 2004 (19.11.2004) US

(71) Applicant (for all designated States except US):
NORTHROP GRUMMAN CORPORATION [US/US];
1840 Century Park East, Los Angeles, California
90067-2199 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **POETKER, John James** [US/US]; 14762 Shirley Bohn, Mt. Airy, Maryland 21771 (US). **JONES, Jeffrey Allen** [US/US]; 3324 N. Berkshire Court, Ijamsville, Maryland 21574 (US). **FOLKART, Michael Bruce** [US/US]; 613 Warfield Drive, Rockville, Maryland 20850 (US). **SHUMAY, Jerry Leroy** [US/US]; 2271 Glenmore Terrace, Rockville, Maryland 20850 (US).

(74) Agents: **ROSENBLOOM, Brian S.** et al.; ROTHWELL, FIGG, ERNST & MANBECK, 1425 K STREET, NW, Suite 800, Washington, District Of Columbia 20005 (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, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, 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.

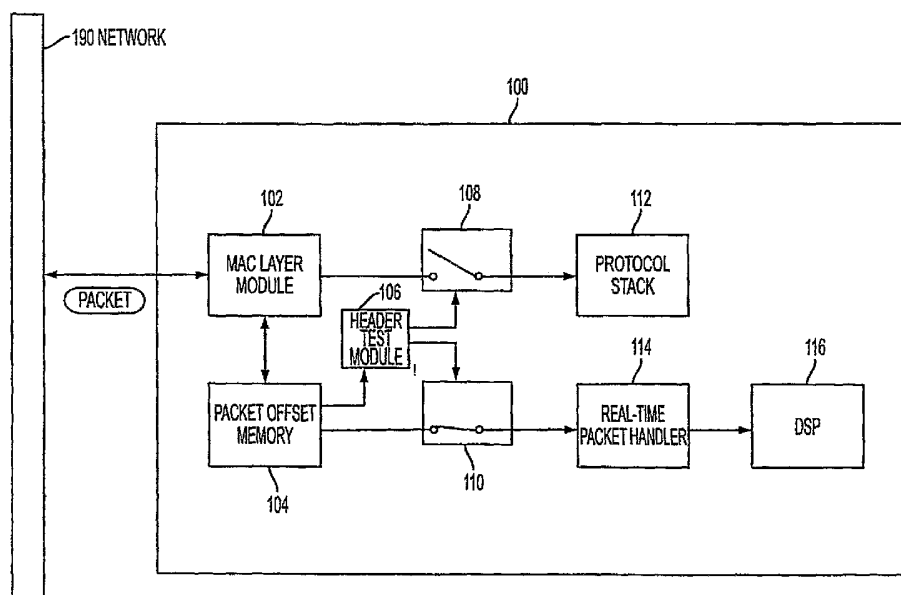
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT,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

[Continued on next page]

(54) Title: REAL-TIME PACKET PROCESSING SYSTEM AND METHOD



(57) Abstract: In some embodiments, the invention may include a combination of real-time packet detection, processing and routing. When implemented in a distributed architecture, such systems can yield a low cost, high availability and/or secure network capable of switching real-time data and delivering the quality of service expected in mission critical systems.



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.

REAL-TIME PACKET PROCESSING SYSTEM AND METHOD

[001] This application claims the benefit of U.S. Provisional Patent Application No. 60/629,331, filed on November 19, 2004, the entire contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

[002] Field of the invention

[003] The present invention relates generally to the processing of packets in packet switching systems, including, e.g., TCP/IP packets. Some preferred embodiments involve audio, video and/or multimedia packet processing systems and methods.

[004] Discussion of the Background

[005] Packet switching systems have found minimal acceptance in mission critical applications. This is largely due to the fact that packet switching systems may experience low real-time data quality under heavy network loading and propagation delays due to limited network bandwidth.

[006] In addition to network bandwidth concerns, conventional packet switched network nodes need to receive and process all network traffic to determine if the packet is of interest. To examine a packet, a node processor needs to host at least part of a network protocol software stack. In large systems, this requirement can impact the cost of network nodes

because powerful microprocessors are needed to handle the processing load.

[007] However, packet switching systems have characteristics that can be advantageous in mission critical applications. For example, such systems do not require the distribution of critical timing reference signals to all nodes to accommodate audio and real-time data distribution. In addition, illustrative packet switching systems may require only a standard type network connection between system nodes (e.g., an Ethernet connection) which can have the effect of simplifying installation, reducing infrastructure costs and significantly lowering life cycle costs.

[008] What is desired, therefore, are systems and methods to overcome the above described and/or other limitations of packet switching nodes so that they are better suited for critical applications.

SUMMARY OF THE INVENTION

[009] Some preferred embodiments of the invention described in this patent application overcome limitations of packet switched networks in mission critical applications. In some embodiments, the invention may include a combination of real-time packet detection, processing and routing. When implemented in a distributed architecture, such systems can yield a low cost, high availability and/or secure network capable of switching real-time data and delivering the quality of service expected in mission critical

systems. These significant improvements can be accomplished using standard network protocols and infrastructure assuring compatibility with both existing networks and future deployments of packet switched systems.

[0010] A system according to one embodiment of the invention includes: a media access control layer module for receiving packet data from a network; a packet memory for storing the packet data received from the network, the packet data including packet header data; a real-time packet handler; a processor executing a protocol stack; and a packet header test module that examines the packet header data to determine whether the packet header data indicates that the payload associated with the packet header contains real-time data, wherein the header test module routes the packet data to the processor if the module determines that the packet header does not indicate that the payload contains real-time data and routes the packet data to a real-time packet handler if the module determines that the packet header indicates that the payload contains real-time data.

[0011] In some embodiments, the real-time packet handler comprises a controller for filtering and processing the packet data and a queue, coupled to a digital signal processor, for storing the filtered and processed packet data. In some embodiments, the real-time packet handler is configured to (a) determine the source address included in the packet data, (b)

determine whether the source address is included in a list of predetermined source addresses; and (c) determine whether the source address is associated with an active channel, wherein the handler modifies the packet data and stores the modified packet data in said queue if the source address is associated with an active channel and the source address is included in said list of predetermined source addresses.

[0012] A method according to one embodiment of the invention includes the steps of: (a) receiving packet data from a network, the packet data comprising packet header data and packet payload data; (b) examining the packet header data; (c) determining whether the packet payload data includes real-time data based on the packet header data; (d) providing the packet data to a processor running a protocol stack if the packet payload data does not include real-time data; and if the packet payload data includes real-time data then, (e) storing the packet data in a queue coupled to a digital signal processor if the packet data passes through a filter and the source address is associated with an active channel; and (f) storing the packet data in said queue if the packet data passes through said filter and the payload of the packet data does not consist of comfort noise and/or silence.

[0013] The above and other features and advantages of the present invention, as well as the structure and operation of preferred embodiments of the present

invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The accompanying drawings, which are incorporated herein and form part of the specification, help illustrate various embodiments of the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use embodiments of the invention. In the drawings, like reference numbers indicate identical or functionally similar elements.

[0015] FIG. 1 is a functional block diagram of a computer system 100 according to one embodiment.

[0016] FIG. 2 is a functional block diagram of a real-time packet handler according to one embodiment.

[0017] FIG. 3 is a flowchart illustrating a process according to one embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0018] FIG. 1 is a functional block diagram of a computer system 100 according to one embodiment of the invention. System 100 includes a media access control (MAC) layer module 102. MAC layer module 102 receives packets from and transmits packets onto network 190. When MAC layer module 102 receives packet data from network 190, MAC layer module 102 forms a packet and stores the packet data in packet offset memory 104.

[0019] A header test module 106 examines the packet data stored in memory 104 to determine whether the packet being received by MAC layer 102 is a real-time packet (i.e., a packet that requires special processing). For example, header test module 106 may examine the appropriate packet header to determine if the packet contains a real-time payload (e.g., voice payload).

[0020] If header test module 106 determines that the packet is not a real-time packet, then it closes switch 108 so that the packet can be routed to a processor (e.g., a RISC processor, a CISC processor or other processor) that is executing a conventional protocol stack 112 (e.g., a TCP/IP protocol stack) for processing. If header test module 106 determines that the packet is a real-time packet, then it closes switch 110 so that the packet can be routed to a real-time packet handler 114. Switches 108 and 110 may be hardware or software switches. Real-time packet handler 114 may be configured to filter and modify the packets that it receives and to present the filtered and modified packets to a digital signal processor (DSP) 116 that processes the payload of the packet.

[0021] Referring now to FIG. 2, FIG. 2 is a functional block diagram of real-time packet handler 114. Handler 114 may have a packet memory 202 for storing the packet received from packet offset memory 104. Handler 114 may also include a controller 204

that is configured to process the packet stored in memory 202.

[0022] Referring now to FIG. 3, FIG. 3 is a flowchart illustrating a process 300 according to one embodiment that may be performed by controller 204. Process 300 may begin in step 302, where controller 204 determines the source address of the packet. In step 304, controller may determine whether the source address falls within a range of predefined addresses. For example, the range may include 65,536 addresses. If the determined source address is not within the range, then controller 204 may drop the packet (step 306). If the source address is within the range, then controller 204 may compare the source address to a table 206 that stores addresses of specific interest to an application or user (step 308). If controller 204 determines that there is no match between the source address and an address in table 206, then controller 204 may drop the packet (step 306).

[0023] If, on the other hand, controller 204 determines that the source address of the packet is included in table 206, then controller 204 will continue to process the packet (i.e., control may pass to step 312).

[0024] In step 312, controller 204 may determine whether the source address is associated with an active channel. For example, in some embodiments, a source address is associated with an active channel if the source address is stored in an active channels table

208. In some embodiments, there may be a limit to the number of active channels. For example, in one embodiment, the DSP 116 can process only 128 channels at a given point in time. Thus, in this embodiment, the number of active channels should not exceed 128. The active channels table maps source addresses to channel numbers.

[0025] If, in step 312, controller 204 determines that the source address is associated with an active channel, then process 300 may proceed to step 314. If, on the other hand, controller 204 determines that the source address is not associated with an active channel value, then process 300 may proceed to step 330.

[0026] In step 314, controller 204 modifies the header of the packet. For example, controller 204 may remove unnecessary information from the header (e.g., unnecessary RTP definitions are removed) and insert into the header the channel number that is associated with the source address of the packet. As discussed above, table 208 associates channel numbers with source addresses. That is, each source address stored in table 208 may be associated with a unique channel number. In some embodiments, after step 318, the header of the packet no longer contains all RTP definitions but only the channel assignment value and control bits.

[0027] In step 316, controller examines the payload of the packet to determine whether the payload consists of either comfort noise and/or silence. If controller

204 determines that the payload consists of either comfort noise and/or silence, then controller 204 may remove the channel assignment from the table 208 (step 318). That is, controller 204 may modify table 208 so that the source address of the packet is not associated with the value of an active channel.

[0028] If controller 204 determines that the payload includes something other than comfort noise and/or silence, then process 300 may proceed to step 320. Additionally, process 300 proceeds to step 320 after step 318.

[0029] In step 320, the now completely formed but modified packet header and payload is then stored in a queue 210. Queue 210 may include one or more first-in, first-out (FIFO) queues. For example, in some embodiments, queue 210 includes two FIFO queues so that while controller 204 writes a packet to one of the queues the DSP 116 can read a packet from the other queue.

[0030] In some embodiments, the packet or a portion of the packet (e.g., the packet payload) being processed by controller 204 is encrypted. In such embodiments, table 206 or 208 may associate each source address in the table with a key that is used to decrypt the packet. In this embodiment, before controller 204 writes a packet to queue 210, controller 204 uses the key associated with the source address of the packet to decrypt the packet or portion thereof that is encrypted.

[0031] Referring now to step 330, in step 330 controller 204, controller 204 examines the payload of the packet to determine whether the payload consists of either comfort noise and/or silence. If controller 204 determines that the payload consists of either comfort noise and/or silence, then controller 204 may drop the packet (step 332).

[0032] If controller 204 determines that the payload includes something other than comfort noise and/or silence, then process 300 may proceed to step 334.

[0033] In step 334, controller 204 associates the source address with a channel and modifies the header of the packet. For example, controller 204 may remove unnecessary information from the header and insert into the header the channel number that is associated with the source address of the packet. In some embodiments, after step 334, the header of the packet no longer contains all RTP definitions but only the channel assignment value and control bits. After step 334, control may pass to step 320.

[0034] In preferred embodiments of the invention, modules 102, 106 and 114 are implemented in hardware, but this is not a requirement. For example, modules 102, 106 and 114 may be implemented using one or more field programmable gate arrays (FPGAs) and/or application specific integrated circuits (ASICs). Additionally, protocol stack 112 is preferably implemented in software that executes on a general

purpose processor (e.g., a RISC process or other processor).

[0035] While various embodiments/variations of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

[0036] Additionally, while the process described above and illustrated in the drawings is shown as a sequence of steps, this was done solely for the sake of illustration. Accordingly, it is contemplated that some steps may be added, some steps may be omitted, the order of the steps may be re-arranged, and some steps may be performed simultaneously.

What is claimed is:

1. A system for processing packets containing real-time audio and/or video data, comprising:
 - a media access control layer module for receiving packet data from a network;
 - a packet memory for storing the packet data received from the network, the packet data including packet header data;
 - a real-time packet handler;
 - a processor executing a protocol stack;
 - a packet header test module that examines the packet header data to determine whether the packet header data indicates that the payload associated with the packet header contains real-time data, wherein the header test module routes the packet data to the processor if the module determines that the packet header does not indicate that the payload contains real-time data and routes the packet data to a real-time packet handler if the module determines that the packet header indicates that the payload contains real-time data, wherein
 - the real-time packet handler comprises a controller for filtering and processing the packet data and a queue, coupled to a digital signal processor, for storing the filtered and processed packet data, and
 - the real-time packet handler is configured to (a) determine the source address included in the packet data, (b) determine whether the source address is

included in a list of source addresses; and (c) determine whether the source address is associated with an active channel, wherein the handler modifies the packet data and stores the modified packet data in said queue if the source address is associated with an active channel and the source address is included in said list of source addresses.

2. The system of claim 1, wherein the handler is further configured to determine whether the payload of the packet data consists of comfort noise and/or silence.

3. The system of claim 2, wherein, if a data structure associates the source address with an active channel and if the handler determines that the payload of the packet data consists of comfort noise and/or silence, then the handler modifies the data structure so that the source address is not associated with the active channel.

4. The system of claim 2, wherein, if a data structure does not associate the source address with an active channel and if the handler determines that the payload of the packet data does not consist of comfort noise and/or silence, then the handler modifies the data structure so that the source address is associated with an active channel.

5. The system of claim 1, wherein the header test module is implemented using an FPGA.

6. The system of claim 5, wherein the real-time packet handler is implemented using an FPGA.

7. The system of claim 6, wherein the processor that executes the protocol stack is a RISC or CISC processor.

8. The system of claim 1, wherein the handler determines whether the source address is associated with an active channel only after the handler determines that the source address is included in said list of source addresses.

9. The system of claim 1, wherein, prior to determining whether the source address is included in said list, said handler determines whether the source address is within a predetermined range of addresses.

10. The system of claim 1, wherein, prior to the modifying the packet data and storing the modified packet data in said queue, the handler decrypts at least a portion of the packet data using a key associated with the source address.

11. A method for processing packets containing real-time audio and/or video data, comprising:

(a) receiving packet data from a network, the packet data comprising packet header data and packet payload data;

(b) examining the packet header data;

(c) determining whether the packet payload data includes real-time data based on the packet header data;

(d) providing the packet data to a processor running a protocol stack if the packet payload data does not include real-time data; and

if the packet payload data includes real-time data then,

(e) storing the packet data in a queue coupled to a digital signal processor if the packet data passes through a filter and the source address is associated with an active channel; and

(f) storing the packet data in said queue if the packet data passes through said filter and the payload of the packet data does not consist of comfort noise and/or silence.

12. The method of claim 11, wherein step (e) is performed regardless of whether the payload of the packet data consists of comfort noise and/or silence.

13. The method of claim 11, wherein step (f) occurs regardless of whether the source address is associated with an active channel.

14. The method of claim 11, wherein the packet data passes through said filter if the source address included in the packet header data is included in a list of addresses.

15. The method of claim 11, wherein the packet data passes through said filter if the source address included in the packet header data is included in a list of addresses and is within a predetermined address range.

16. The method of claim 11, further comprising determining whether the packet payload data consists of comfort noise and/or silence.

17. The method of claim 16, wherein, if the packet payload data consists of comfort noise and/or silence and a data structure associates the source address with an active channel, then the method further includes modifying the data structure so that the data structure does not associate the source address with an active channel.

18. The method of claim 16, wherein, if the packet payload data includes voice data and a data structure does not associate the source address with an active channel, then the method further includes modifying the data structure so that the data structure associates the source address with an active channel.

19. The method of claim 11, wherein said queue comprises one or more first-in, first-out queues.

20. The method of claim 11, wherein the digital signal processor is configured to read packets from said queue.

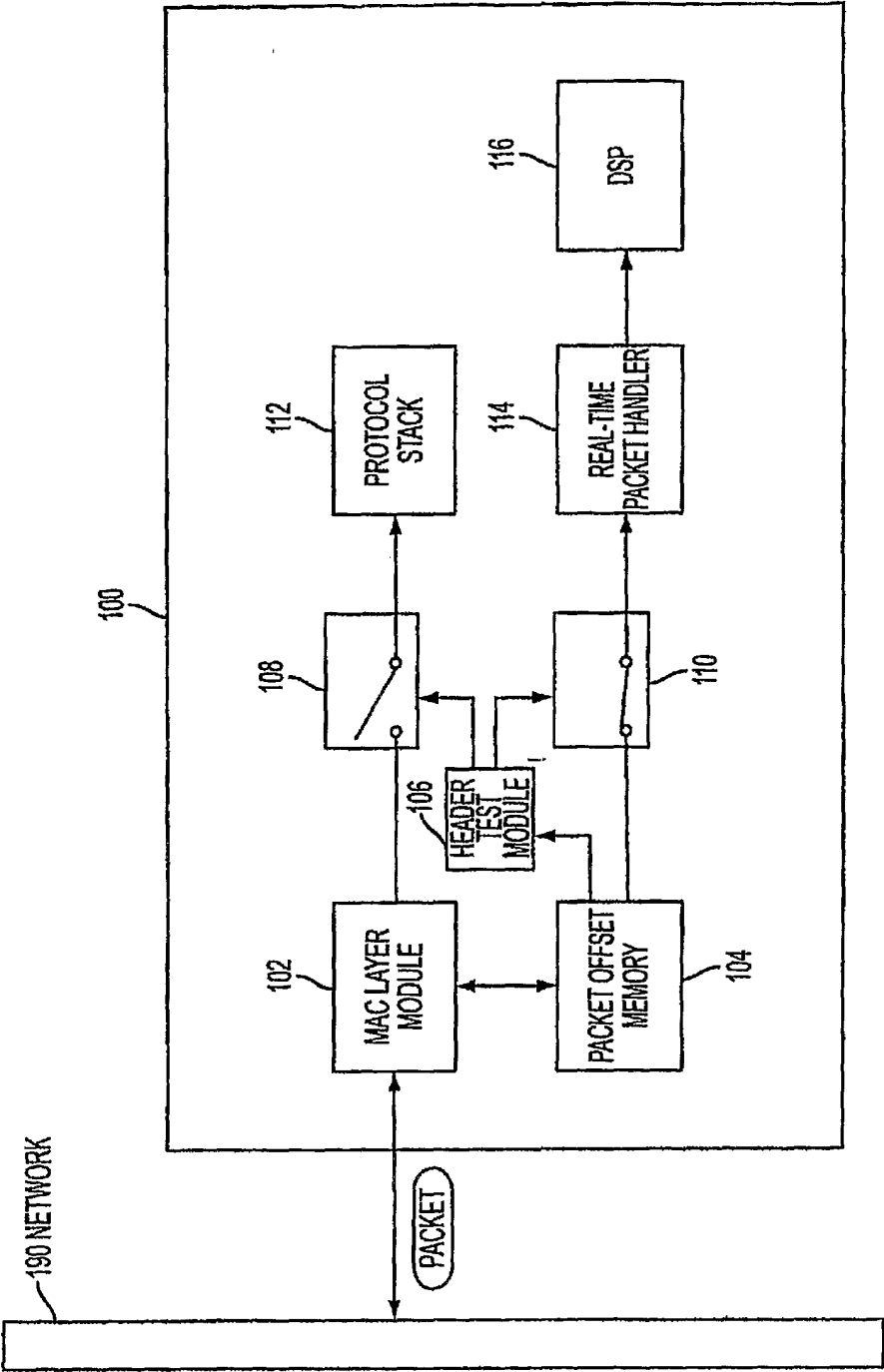


FIG. 1

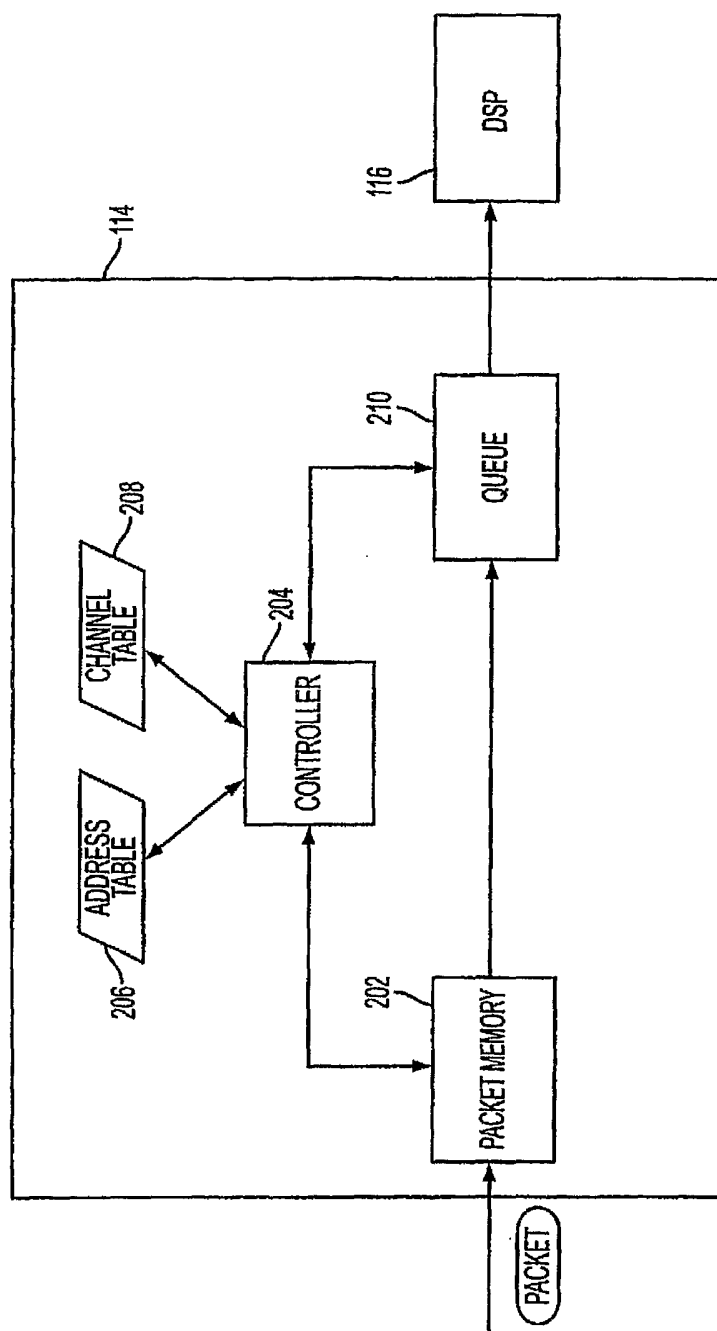


FIG. 2

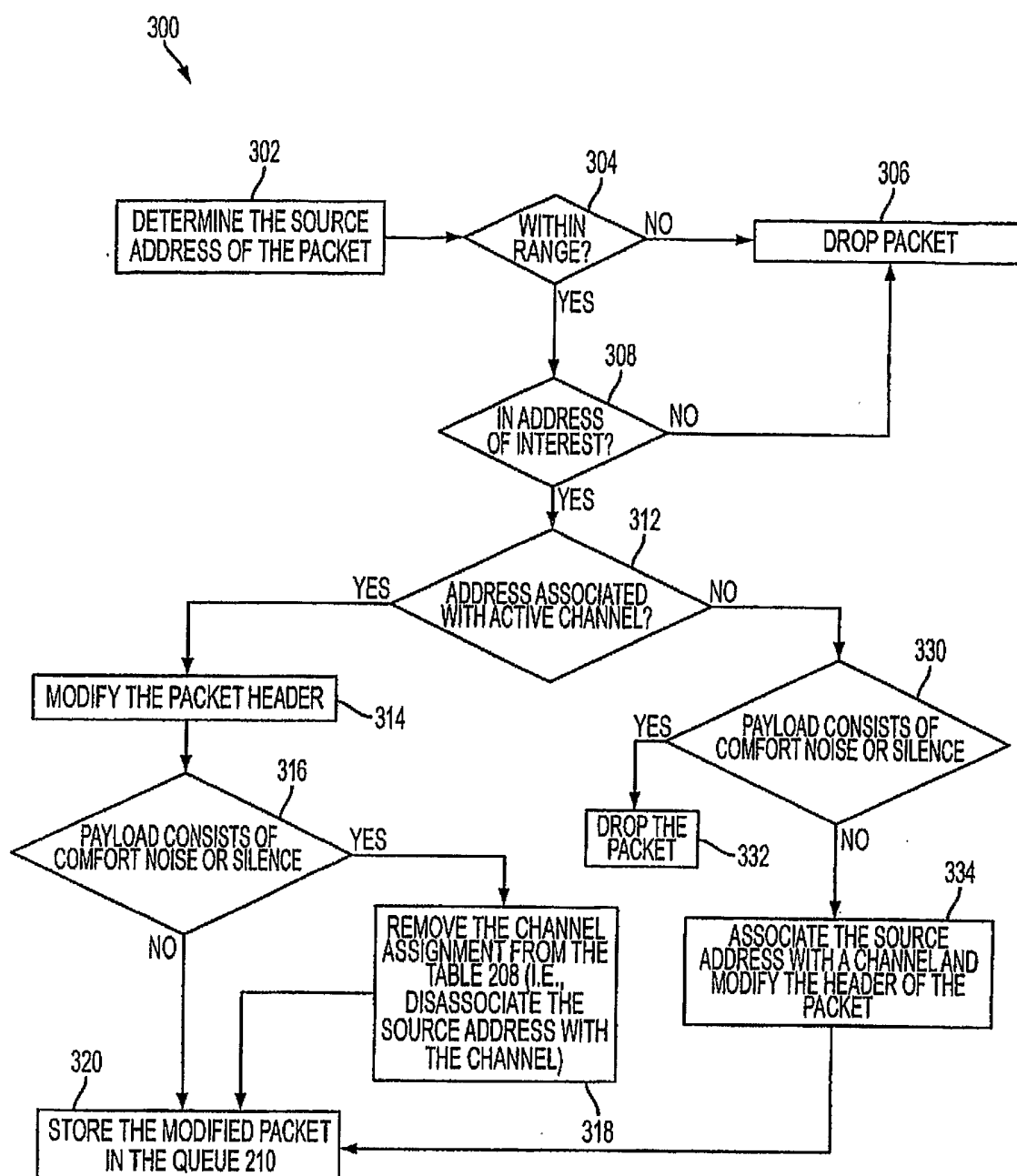


FIG. 3