

(51) International Patent Classification:
H04L 1/16 (2006.01)(21) International Application Number:
PCT/US2014/062196(22) International Filing Date:
24 October 2014 (24.10.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).(72) Inventor: **SHERLOCK, Derek, A.**; 3404 E. Harmony Road, Mail Stop 35, Fort Collins, CO 80528 (US).(74) Agents: **SHOOKMAN, Jeb, A.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) Title: END-TO-END NEGATIVE ACKNOWLEDGMENT

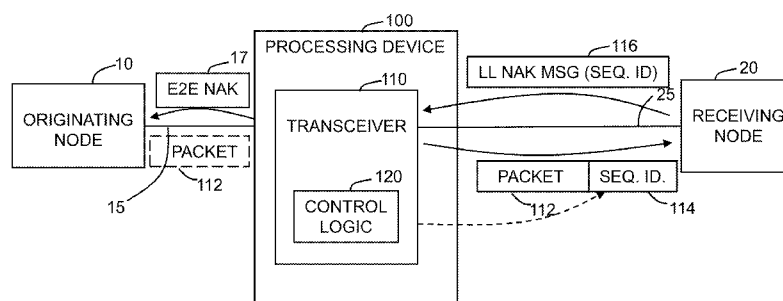


FIG. 1

(57) **Abstract:** A processing device includes a transceiver to be coupled to a link and control logic coupled to the transceiver. The control logic is to assign a unique sequence identifier to each packet to be transmitted across the link to a receiving node and transmit packets via the transceiver across the link to the receiving node. Each packet is to have a unique sequence identifier. The control logic also is to receive a message from the receiving node, the message containing the sequence identifier of a packet not correctly received by the receiving node. Based on the received message, the control logic is to cause an end-to-end negative acknowledgment (E2E NAK) packet to be transmitted to an originating node of the packet that was not correctly received.

END-TO-END NEGATIVE ACKNOWLEDGMENT

BACKGROUND

[0001] Many network fabrics include various switches, routers, etc. to receive and forward packets to thereby permit originating endpoint nodes to send packets to destination endpoint nodes. A link is a communication pipeline between adjacent nodes in the fabric. The nodes on either end of a link may include endpoint nodes which produce and consume traffic, and intermediate nodes which propagate traffic from one node to another node. Endpoint nodes may comprise, for example, central processing units (CPUs), memory devices, storage devices, peripherals, accelerators, renderers, graphical display devices, etc. Intermediate nodes may comprise, for example, switches, routers, proxies, translators, repeaters, protocol converters, etc. A packet may be transmitted from a transmitting node to a receiving node over a dedicated link between such nodes.

[0002] Some networks employ end-to-end (E2E) retry, link level retry (LLR), or both, to ensure that a packet arrives at its intended destination without error. LLR addresses transient CRC-detectable packet corruption due to electrical interference for packets crossing an individual link. The LLR mechanism is implemented across each link independently of the other links. LLR assigns unique sequence numbers to individual packets transmitted across a given link so that if an individual packet experiences an error during transmission over a given link, the receiving node can inform the transmitting node for that link can be made aware of that fact (via a link level negative acknowledgment (NAK) message from the receiving node) and can retry (i.e., resend) the packet again across the link. Each transmitting node stores copies of its outgoing packets in they need to be re-sent. If a receiving node detects an error with a packet received from a transmitting node, the receiving node sends the link level NAK message back across the link to the transmitting node. The link level NAK message includes the sequence number of the packet that had the error. The transmitting node responds to the link level NAK message by resending the identified packet.

[0003] E2E retry may be used in combination with LLR, or alone. E2E retry permits an originating endpoint node to resend a packet if an acknowledgment of that packet is not timely received from the destination endpoint node. E2E can protect against component or link failures along a route, by migrating the retry to an alternate route.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] For a detailed description of various examples, reference will now be made to the accompanying drawings in which:

[0005] Figure 1 shows a system in accordance with various examples;

[0006] Figure 2 shows another system in accordance with various examples;

[0007] Figure 3 shows sequence identifiers in storage of a transceiver in accordance with various examples;

[0008] Figure 4 shows sequence identifiers, originating node identifiers, and unique packet identifiers in storage of a transceiver in accordance with various examples;

[0009] Figure 5 shows an example of a non-transitory computer-readable storage device in accordance with various examples;

[0010] Figure 6 shows an example of a non-transitory computer-readable storage device in accordance with various examples;

[0011] Figure 7 illustrates a processing device in accordance with example;

[0012] Figure 8 illustrates a transceiver usable in the processing device of Figure 7 in accordance with an example;

[0013] Figure 9 illustrates a transceiver usable in the processing device of Figure 7 in accordance with another example;

[0014] Figure 10 illustrates an endpoint node in accordance with an example;

[0015] Figure 11 shows a method in accordance with various examples; and

[0016] Figure 12 shows another method in accordance with various examples.

DETAILED DESCRIPTION

[0017] For LLR to work, each transmitting node maintains local storage for its outgoing packets in case a receiving node detects an error and requests the transmitting node to retry a particular packet. The amount of local storage can be large and generally scales with the link speed (packet sending rate) and the

round-trip latency across the link being driven (i.e., the total time taken for a packet to cross the link, and for a corresponding acknowledgement to cross back in the other direction).

[0018] The disclosed implementations avoid the need for as much local storage. In at least some implementations, sufficient local storage in a node is provided for storage of identification information of the originating node for a given packet and the identity of the packet. Local storage in the node for the packet itself is not needed or provided. If a transmitting node receives a link-level NAK for a particular packet, rather than resending the requested packet, the transmitting node sends an end-to-end negative acknowledgment (E2E NAK) packet back to the endpoint node that originated the packet in the first place (referred to herein as the “originating node”). The E2E NAK packet may be routed back to the originating node along the same path (but in the reverse direction) that the original packet took from the originating endpoint node to the transmitting node. At that point, the E2E retry mechanism of the network will cause the originating node to resend the packet through the fabric. In other words, the nodes of the network rely on the E2E retry mechanism to thereby avoid having to store copies of packets pending a potential link level NAK of any given packet.

[0019] Figure 1 illustrates one example of a processing device 100 which receives a packet 112 from an originating node 10 across a link 15 and transmits the packet 112 across link 25 to a receiving node 20. The processing device 100 may receive the packet directly from the originating node 10 or indirectly via other processing devices. The originating node 10 originates (e.g., creates) the packet and the receiving node 20 may consume the packet or forward it on to another node in the network. The processing device 100 may be implemented as a switch, router, proxy, translator, repeater, protocol converter, or other type of network device. The processing device 100 may be an intermediate node in a network fabric along a path between originating and destination endpoint nodes. While Figure 1 illustrates the processing device 100 coupled to the originating node 1, the processing device 100 may be coupled to the originating node through one or more additional intermediate nodes (e.g., other processing

devices 100). The processing device 100 receives and forwards packets, generally bidirectionally.

[0020] In the example of Figure 1, the processing device 100 is coupled to the originating and receiving nodes 10, 20 via links 15 and 25, respectively. The processing device 100 includes a transceiver 110 coupled to link 25. Control logic 120 is part of, or otherwise accessible to, the transceiver 110. An additional transceiver may be included as well as to couple to link 15. The transceiver 110 may include a receiver and transmitter and thus facilitate bidirectional communications across link 25 to the receiving node 20. Similar bidirectional communications are possible as well between processing device 100 and originating node 10 across link 15. In some implementations, the processing device 100 may include multiple transceivers 110—one to support each link to which the processing device is to be coupled.

[0021] The control logic 120 may be implemented as a programmable hardware processor that executes machine instructions, a discrete circuit or any other type of logic that implements the functionality described herein. For example, the control logic 120 may control the operation of the transceiver 110 to send packets, received from originating node 10, across link 25 to the receiving node 20. The control logic 120 may assign a unique sequence identifier to each packet to be transmitted across link 25 to receiving node 20. Figure 1 shows an example of a packet 112 to which a sequence identifier 114 has been included. The packet 112, which may have been received from the originating node 10, and its sequence identifier are transmitted across link 25 to the receiving node 20. The sequence identifier 114 uniquely differentiates each packet 112 from other packets transmitted across link 25 so that if the receiving node 20 receives and detects an error with a packet, the receiving node 20 can use that packet's unique sequence identifier to notify the processing device 100 that the packet was found to be corrupted upon receipt.

[0022] In some implementations, the sequence identifiers of packets traversing a particular link are specific only to that particular link. For example, if a packet has "5" for its sequence identifier as the packet is transmitted across link 25, the sequence identifier 5 does not follow that packet as it is forwarded on by the

receiving node 20 across other links. A packet that has the same sequence identifier on two different links is coincidental.

[0023] The control logic 120 may transmit packets (e.g., packets including sequence identifiers) to be transmitted across link 25 to the receiving node 20, which may also include a transceiver that functions similar to the transceiver 110. Any given packet transmitted by transceiver 110 to receiving node 20 may be received by the receiving node with an error detected by the receiving node (e.g., by the transceiver in the receiving node). If the receiving node 20 detects an error with a packet received from the processing device 100, the receiving node 20 may send a link level NAK message (LL NAK MSG) 116 to the processing device. For example, if packet 112 with its unique sequence identifier 114 is received by the receiving node 20 and detected to have an error, the receiving node may send back a link level NAK message 116 containing the sequence identifier 114 of packet 112. If a packet is not determined to have an error, the receiving node 20 may send a link level acknowledgment (ACK) to transceiver 110.

[0024] While some processing devices 100 might resend the packet associated with the sequence identifier contained in the retry message, in accordance with the disclosed examples, transceiver 110 may not include sufficient storage to store a copy of packet 112 in, for example, a retry buffer and thus may not resend the designated packet. Instead, based on the link level NAK message, the control logic 120 of the processing device 100 may generate and transmit an E2E NAK packet 17 back to the originating node 10 (i.e., the node that originated the packet 112) detected to have an error. The E2E NAK packet 17 contains information that can be used to uniquely identify a particular packet (e.g., the packet 112 detected to have an error by the receiving node 20 in this example). The E2E NAK packet 17 indicates to the originating node that the packet associated with the E2E NAK packet was not correctly received at its intended destination endpoint node, which may have been receiving node 20 or a node further downstream from node 20. The E2E NAK packet 17 may not indicate where along the path from the originating node 10 to the destination node that an error with the packet was detected. The originating node 10 may respond to the

E2E NAK packet 17 by causing the packet to be resent through network which may include processing device 100 or other processing devices. That is, the originating node 10 may cause the packet to be resent along the same path or a different path.

[0025] LLR includes a receiving node detecting an error with a packet and sending a link level NAK message to the transmitting node so that the transmitting node will resend the packet on the link. In accordance with the disclosed examples, however, the link level NAK message is implemented but the transmitting node does not respond to the retry message by resending the requested packet. Instead, the transmitting node sends an E2E NAK packet back to the originating node to cause the network's E2E retry process to re-send the packet back into the network fabric.

[0026] Figure 2 shows another example of a processing device 150 coupled to the originating node 10 over link 15 and to the receiving node 20 over link 25. The block diagram of Figure 2 is similar to that of Figure 1. One difference, however, is that the processing device 150 in Figure 2 includes a transceiver 152 and control logic 160, and the transceiver 152 includes storage 154. Storage 154 may be volatile storage (e.g., random access memory) or non-volatile storage. The control logic 160 may be implemented in much the same way, and may perform much the same operations, as described above for control logic 120 of Figure 1. Another difference in Figure 2 is that Figure 2 illustrates that processing device 150 couples to another intermediate node 13 via link 15 and that the intermediate node 13 couples to the originating node 10. Thus, when the processing device 150 generates and transmits an E2E NAK packet 17, the E2E NAK packet 17 is routed through the intermediate node 13 back to the originating node 10.

[0027] As noted above, receiving node 20 may also include a transceiver which functions similar to receiver 110. Thus, the transceiver in node 20 may send packets to processing device 100, receive link level ACKs and link level NAKs from the processing device, and respond in much the same way as explained above regarding the transceiver 110 of processing device 100.

[0028] The example of Figure 2 illustrates that the transceiver 152 of the processing device has, or has access to, storage 154. Storage 154 may not be large enough, or may not be used, to store packets (e.g., packet 112) that are forwarded on across link 25 to receiving node 20. Without the use of local storage to store copies of transmitted packets, the processing device is unable to respond to a link level NAK message to retrieve and resend a packet detected to have an error by the receiving node 20.

[0029] The processing device 150 may have some storage such as storage 154 to store sequence identifiers for the packets transmitted across the link 25. Thus, the transceiver 152 can store the sequence numbers of packets it transmits in case the processing device receives a link level NAK message 116 associated with any of those packets. Figure 3 illustrates that storage 154 may store one or more sequence identifiers for the various packets transmitted across link 25, although additional information may be included in storage 154 as well. If and when the receiving node 20 sends an acknowledgment message back to the processing device 150 that certain packets have been correctly received (no errors), the control logic 160 may remove the sequence numbers from storage 154 corresponding to the packets that are identified as having been successfully transmitted to receiving node 20.

[0030] Figure 4 illustrates another example of the contents of storage 154. In this example, the storage 154 includes, for each packet transmitted across link 25, any or all of: the sequence identifier 162 of the packet, an identifier 164 of the node that originated the packet (originating node identifier) and a unique packet identifier 166. The unique packet identifier identifies the particular packet which needs to be re-sent by the E2E mechanism, since a given originating endpoint may have multiple outstanding packets at the same time. The unique identifier, for example, may be the same identifier used within the E2ED ACK packets in the E2E retry scheme to identify successfully delivered packets. The unique packet identifier may be a purpose-added serial number generated by the originating endpoint node and carried in every packet, or it may be derived from some other field(s) in the packet header which are already known to be unique – for example, a protocol-layer transaction identifier in architectures where this is guaranteed to

be unique. The originating node identifier may include an address (e.g., an internet protocol (IP) address of the originating node or some other identifier by which the processing device can determine to where to send to the E2E NAK packet 17. The E2E NAK packet 17 may be sent to the originating node identifier of the packet determined not have been received correctly by the receiving node 20, and may contain the unique packet identifier associated with that packet.

[0031] Figure 5 illustrates an example of an implementation of control logic 120, 160. In this example, a processing resource 172 is coupled to a non-transitory, computer readable storage device 174. The non-transitory, computer readable storage device 174 may include volatile storage (e.g., random access memory), non-volatile storage (e.g., solid state storage, magnetic storage, optical storage, etc.) or combinations thereof. The storage device 174 contains machine executable instructions that may be executed by the processing resource 172.

[0032] The processing resource 172 may include an application specific integrated circuit (ASIC), a field-programmable gate array (FPA), discrete logic, a single hardware processor, multiple hardware processors, a single computer, or a network of computers. The machine instructions include various software modules 176, 178, 180, and 182 that, when executed by the processing resource 172, cause the processing resource to perform various operations as described below. The software modules 176-182 may be implemented as separate modules or various groups of all of the software modules may be implemented as a single software module.

[0033] The sequence identifier assignment module 176 causes the processing resource 172 to assign a unique sequence identifier to each packet to be transmitted across a link to a receiving node (e.g., across link 25 to receiving node 20). The sequence identifiers may be numbers and the processing resource 172 may assign each new packet a sequence identifier by incrementing a sequence identifier used for a preceding packet.

[0034] If and when the receiving node 20 detects an error with a packet transmitted by the processing device 100, 150, the receiving node sends a link level NAK message 116 including the sequence identifier of the packet detected to have an error. The message reception module 280 causes the processing

resource 172 to receive the link level NAK message 116 from the receiving node 20 over link 25. Based on the received link level NAK message, the originating node identification module 182 causes the processing resource 172 to determine the identity of the node that originated the packet identified by the link level NAK message. The packet transmission module 178 then causes the processing resource to cause the transceiver 110, 152 to transmit an E2E NAK packet to the originating node, and not to retry the packet on the link 25. In some implementations, the originating node identification module 182 causes the processing resource 172 to determine the identity of the originating node by examining the storage 154. For example and with reference to Figure 4, the originating node identification module 182 causes the processing resource 172 to obtain the originating node identifier 164 corresponding to the sequence identifier 162 contained in the link level NAK message 116.

[0035] The packet transmission module 178 also may cause the processing resource 172 to cause the transceiver 110, 152 to transmit packets across link 25 to the receiving node 20—packets that may subsequently be determined by the receiving node to have errors as explained above.

[0036] Some networks may implement virtual channels over their physical links. Virtual channels allow a multiplicity of independent channels of communication (i.e., independent streams of packets) to share a single physical link, and by extension, to share an end-to-end route. Virtual channels are independent of each other in terms of storage allocation in queues, independent in flow control, and independent in competing for access to the physical links that they share. Thus, packets on one virtual channel do not head-of-line block packets on another virtual channel. At the link level, multiple virtual channels may be implemented by time multiplexing of the physical link between the virtual channels, and by communicating independent flow-control information for each virtual channel across each link. Endpoint protocols obey conventions with regard to which virtual channels are used to carry which packet types – for example to avoid potential deadlocks. Since E2E NAK packets are routable packets in their own right, and potentially victims of head-of-line blocking, E2E NAK packets should obey virtual channel conventions consistent with that used

when endpoint nodes generate new packets. For example, E2E NAK packets may travel on the same virtual channel that is used by the endpoint nodes to send end-to-end packet delivery acknowledgement packets. This would be a virtual channel that has no protocol layer dependencies upon the channel carrying the packets being protected.

[0037] Figure 6 illustrates another example of a non-transitory, computer readable storage device 200. The software modules depicted in Figure 6 includes some of the same software modules as in Figure 5. Figure 6, however, shows that the non-transitory, computer readable storage device 200 includes a virtual channel determination module 202. The virtual channel determination module 202 causes the processing resource 172 to determine over which virtual channel the processing resource is to cause the E2E NAK packet 17 to be transmitted. If the network implements a virtual channel for control packets, then the virtual channel determination module 202 may causes the processing resource 172 to use the control packet virtual channel to send the E2E NAK packet.

[0038] Figure 7 illustrates an example of a processing device that includes a processing device core 205 coupled to six transceivers 206a-206f, although the number of transceivers may vary. Each transceiver 206a-206f couples to its own link 207a-207f as shown. A packet received on any one link 207a-207f may be provided by that transceiver to the processing device core 205 for a determination as through which other transceiver/link to forward the packet.

[0039] Figure 8 illustrates an example of at least some of a transceiver 206a-f. The illustrative transceiver includes a receiver (RX) 208 and transmitter (TX) 209 which provide connectivity to that transceiver's link. The RX 208 sends incoming packets to the processing device core 205 for a determination, for example, as to how to forward on the packets. Packets received by other transceivers that the processing device core 205 determines should be transmitted out by a particular transceiver are provided by the core to the sequence identifier append unit 212. A sequence identifier is determined for, and appended to, the outgoing packet by the sequence identifier append unit 212, and the resulting packet with sequence

identifier is provided the TX 209 for transmission to another node over the transceiver's link.

[0040] Packet tracker storage 213 provides storage, as explained above, for information from which the transceiver is able to determine the originating node for a given packet. If the transceiver receives a link level ACK message via RX 208 from another node indicating one or more packets were received without an error, the RX 208 causes the corresponding entries for those packets in the packet tracker storage 213 to be removed.

[0041] If, however, a packet is determined to have an error by a node to which the TX 209 sent the packet, the RX 208 will receive a link level NAK message indicating the existence of the error. The RX may respond to the link level NAK message by causing the E2E NAK generator 214 to generate an E2E NAK packet. The E2E NAK generator 214 may access the packet tracker storage 213 to determine the identity of the source of the packet experiencing the error in order to generate the E2E NAK packet. The E2E NAK generator 214 provides the E2E NAK packet to the processing device core 205. The processing device core 215 then causes the E2E NAK packet to be transmitted by another transceiver back towards the target originating node so that the originating node can cause the packet to be re-sent through the fabric.

[0042] Figure 9 illustrates another example of a transceiver. The illustrative transceiver includes RX 208 and TX 209 which provide connectivity to that transceiver's link as explained above. A link level negative acknowledgment/acknowledgment (NAK/ACK) generator 210 receives an indication from the RX 208 as to whether an error was detected for an incoming packet. In other implementations, the link level NAK/ACK generator 210 itself determines whether an error occurred. The link level NAK/ACK generator 210 causes the TX 209 to send back across the link level NAK and link level ACK messages as appropriate depending on whether or not packet errors are detected.

[0043] The send to processing device core block 215 provides packets received via RX 208 to the processing device core 205 for a determination, for example, as to how to forward on the packets. Packets that the processing device core 205

determines should be transmitted out by the transceiver are provided by the core to the receive from processing device core block 211. A sequence identifier is determined for, and appended to, the outgoing packet by the sequence identifier append unit 212 and the resulting packet with sequence identifier is provided the TX 209 for transmission.

[0044] Packet tracker storage 213 provides storage, as explained above, for information from which the transceiver is able to determine the originating node for a given packet. If the transceiver receives a link level ACK message via RX 208 from another node indicating one or more packets were received without an error, the RX 208 causes the corresponding entries for those packets in the packet tracker storage 213 to be removed.

[0045] If, however, a packet is determined to have an error by a node to which the TX 209 sent the packet, the RX 208 will receive a link level NAK message indicating the existence of the error. The RX may respond to the link level NAK message by causing the E2E NAK generator 214 to generate an E2E NAK packet. The E2E NAK generator 214 may access the packet tracker storage 213 to determine the identity of the source of the packet experiencing the error in order to generate the E2E NAK packet. The E2E NAK generator 214 provides the E2E NAK packet to the processing device core 205 via the send to processing device core block 215. The processing device core 215 then causes the E2E NAK packet to be transmitted by another transceiver back towards the target originating node so that the originating node can cause the packet to be re-sent through the fabric.

[0046] Figure 10 illustrates an endpoint node 199. The endpoint node of the example of Figure 10 includes a transceiver 206 such as that described above coupled to endpoint logic 201. The send to/from processing device core blocks have been relabeled as send to endpoint logic block 215a and receive from endpoint logic block 211a, but generally, the transceiver 203 functions in much the same way as described above.

[0047] The endpoint logic 201 includes a protocol decoder 216 which decodes packets received by the transceiver 203. Healthy received packets are provided to the endpoint protocol agent 217 for consumption and to an E2E ACK generator

218 for generation of an E2E ACK to be transmitted back to the originator of the packet. The endpoint protocol agent 217 may generate new packets which may be provided to the E2E retry buffer 219, where a copy is stored against the possibility that future resends may be needed.

[0048] If the transceiver 203 receives an E2E NAK packet (which indicates that a packet originated by the endpoint node 199 was determined to have an error by the fabric), the protocol decoder 216 sends a signal to the E2E resend unit 220 to resend that packet indicated by the E2E NAK packet.

[0049] The E2E resend unit 220 may include timers that determine when a threshold amount of time has elapsed after the endpoint node 199 has transmitted a particular packet with neither an E2E NAK packet nor an E2E ACK packet indicating successful receipt of the original packet. If the original packet was determined by an intermediate node to have an error, as described above, the originating endpoint node will receive an E2E NAK packet indicating an error with that particular packet. If, however, the E2E NAK packet is unable to be transmitted all of the way back to the endpoint node (e.g., due to transmission problems with the fabric, interference, etc.), then the timer functionality present in the E2E resend unit 220 will cause the packet to be resent anyway.

[0050] Figure 11 illustrates a method in accordance with various examples. The operations of the method may be performed in the order shown, or in a different order. Further, two or more of the operations may be performed in parallel rather than sequentially.

[0051] At 230, the method includes assigning a sequence identifier to a packet that originated from an originating node. This operation may be performed the processing resource 172 executing the sequence identifier assignment module 176 as explained previously.

[0052] At 232, the method includes transmitting the packet across a link from a transmitting node to a receiving node. For example, a packet 112 may be transmitted from processing device 100, 150 to receiving node 20.

[0053] At 234, the method includes detecting an error in the packet at the receiving node. This operation may be performed by computing a cyclic redundant check (CRC) value based on the received message and comparing

the computed CRC value against a CRC value contained in the packet itself. A mismatch of CRC values indicates the presence of an error.

[0054] At 236, the method includes receiving a message (e.g., a link level NAK message 116) from the receiving node 20 indicating an error having occurred with the packet and the sequence identifier of the packet experiencing the error. At 238, the method includes transmitting, by the transmitting node (e.g., processing device 100, 150), an E2E NAK packet to the originating node to cause the originating node to retransmit the packet through the network.

[0055] Figure 12 illustrates another example of a method including operations 230, 232, 234, 236, 240, 242, 244, and 246. The operations of the method may be performed in the order shown, or in a different order. Further, two or more of the operations may be performed in parallel rather than sequentially. Operations 230-236 are the same as described above regarding Figure 10 and thus are not repeated here.

[0056] At 240, the method includes, based on receiving the message from the receiving node, retrieving an identifier of the originating node from storage in the transmitting node (e.g., storage 154). At 242, the method includes determining which virtual channel to use for the E2E NAK packet out of a plurality of virtual channels used in the network.

[0057] At 244, the method includes, rather than retrying the packet across the same link again, transmitting an E2E NAK packet to the originating node (over the determined virtual channel) to cause the originating node to retransmit the packet through the network. At 246, the method further includes retransmitting the packet. The packet may be retransmitted across a path that does not include the link over which the original packet was sent and received in error (e.g., link 25).

[0058] The above discussion is meant to be illustrative of the principles and various embodiments of the present invention. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

CLAIMS

What is claimed is:

1. A processing device, comprising:
a transceiver to be coupled to a link; and
control logic accessible to the transceiver, the control logic to:
assign a unique sequence identifier to each packet to be transmitted across the link to a receiving node;
transmit packets via the transceiver across the link to the receiving node, each packet to have a unique sequence identifier;
receive a message from the receiving node, the message containing the sequence identifier of a packet not correctly received by the receiving node; and
based on the received message, cause an end-to-end negative acknowledgment (E2E NAK) packet to be transmitted to an originating node of the packet not correctly received.
2. The processing device of claim 1 wherein each assigned sequence identifier does not follow the packet across multiple links.
3. The processing device of claim 1 further including storage for sequence identifiers but not for packets transmitted across the link.
4. The processing device of claim 3 wherein the storage is further for storing originating node identifiers and unique packet identifiers, each originating node identifier uniquely to identify the originating node that originated a given packet and each unique packet identifier to identify that particular packet, and wherein the E2E NAK packet includes the unique packet identifier associated with a packet not having been correctly received by the receiving node.
5. The processing device of claim 1 wherein the control logic is to transmit the E2E NAK packet on a virtual channel.

6. A transceiver, comprising:
 - a receiver to couple to a link;
 - a transmitter to couple to the link;
 - a sequence identifier append unit to assign a unique sequence identifier to each packet to be transmitted across the link by the transmitter;
 - an end-to-end negative acknowledgment (E2E NAK) generator coupled to the receiver and to generate an E2E NAK packet based on a message received over the link by the receiver and containing the sequence identifier of a packet determined to have an error; and
 - a packet tracker storage to include information usable to identify an originating node of each packet transmitted by the transmitter and unique packet information to identify each such packet;wherein, the E2E NAK generator is to access the packet tracker storage to determine an identify of the originating node of a packet indicated by the message as not correctly received and to generate and transmit an end-to-end negative acknowledgment (E2E NAK) packet to the originating node and not retry the packet on the link on the link.
7. The transceiver of claim 6 wherein the information in the packet tracker storage includes, for each packet transmitted by the transmitter, a sequence identifier of the packet, an originating node identifier, and a unique packet identifier.
8. The transceiver of claim 6 wherein the E2E NAK packet also includes a unique packet identifier to identify the packet not correctly received.
9. The transceiver of claim 6 wherein the message received by the receiver is a link level NAK message.

10. A method, comprising:
 - assigning a sequence identifier to a packet, wherein the packet originated from an originating node;
 - transmitting the packet across a link from a transmitting node to a receiving node;
 - detecting an error in the packet at the receiving node;
 - receiving a message from the receiving node indicating an error having occurred with the packet and the sequence identifier of the packet experiencing the error; and
 - transmitting, by the transmitting node, an end-to-end negative acknowledgment (E2E NAK) packet to the originating node to cause the originating node to retransmit the packet.
11. The method of claim 10 further comprising retransmitting the packet across a path that does not include the link.
12. The method of claim 10 further comprising, based on receiving the message from the receiving node, retrieving an identifier of the originating node from storage in the transmitting node.
13. The method of claim 12 further comprising generating the E2E NAK packet to include the identifier of the originating node and a unique identifier of the packet for which the error was detected.
14. The method of claim 10 further comprising not retrying the packet, by the transmitting node, across the link.
15. The method of claim 10 further comprising determining which virtual channel to use for the E2E NAK packet out of a plurality of virtual channels.

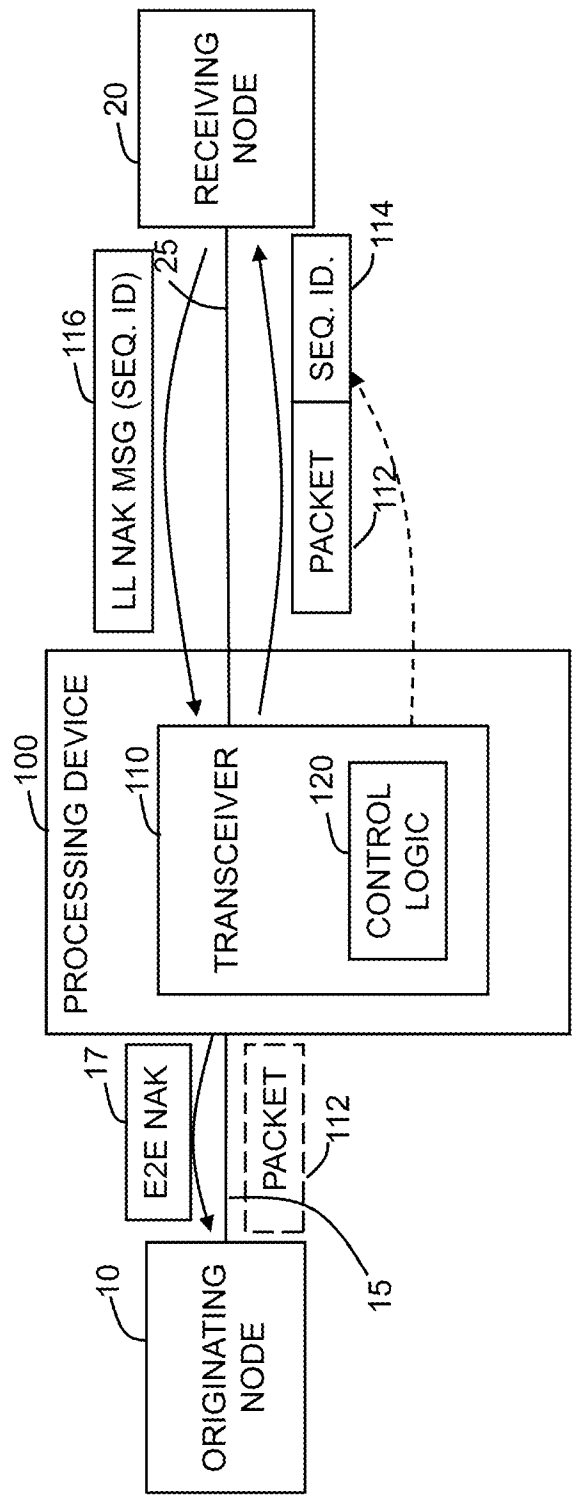


FIG. 1

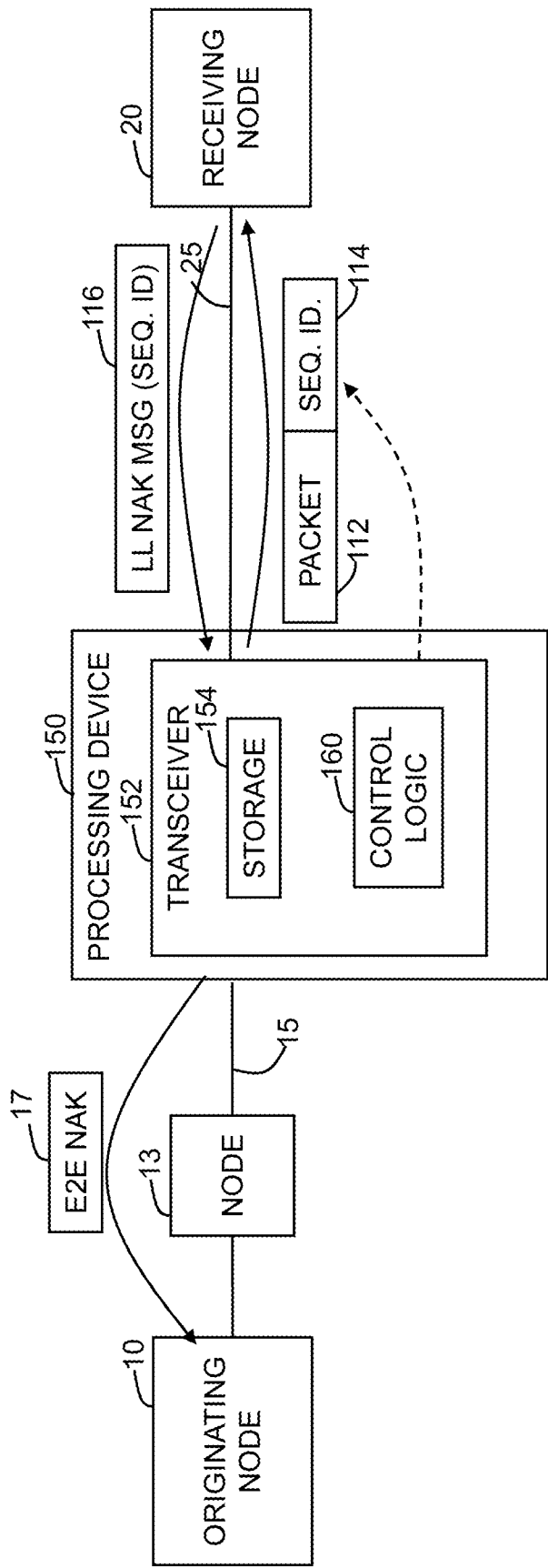


FIG. 2

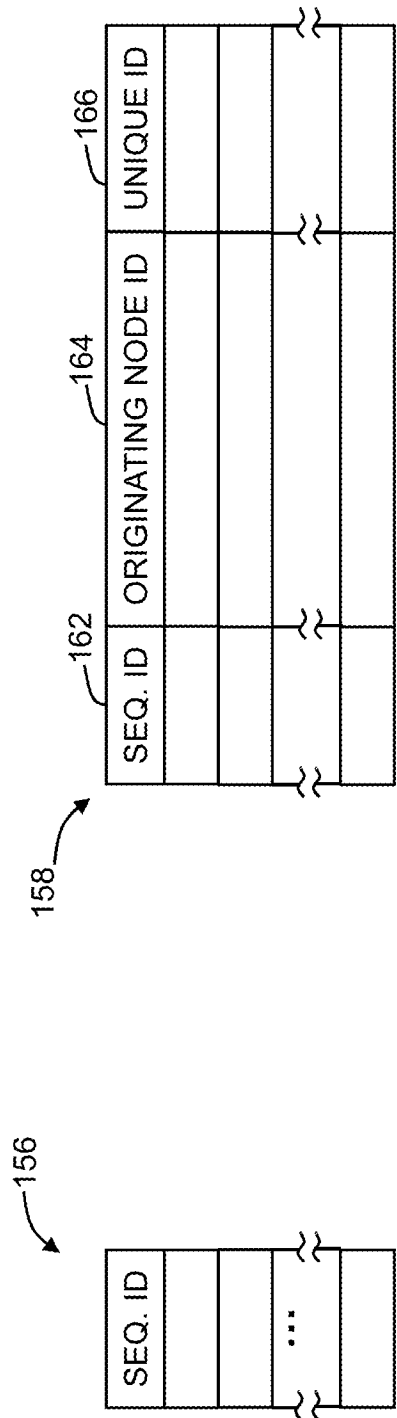


FIG. 3

FIG. 4

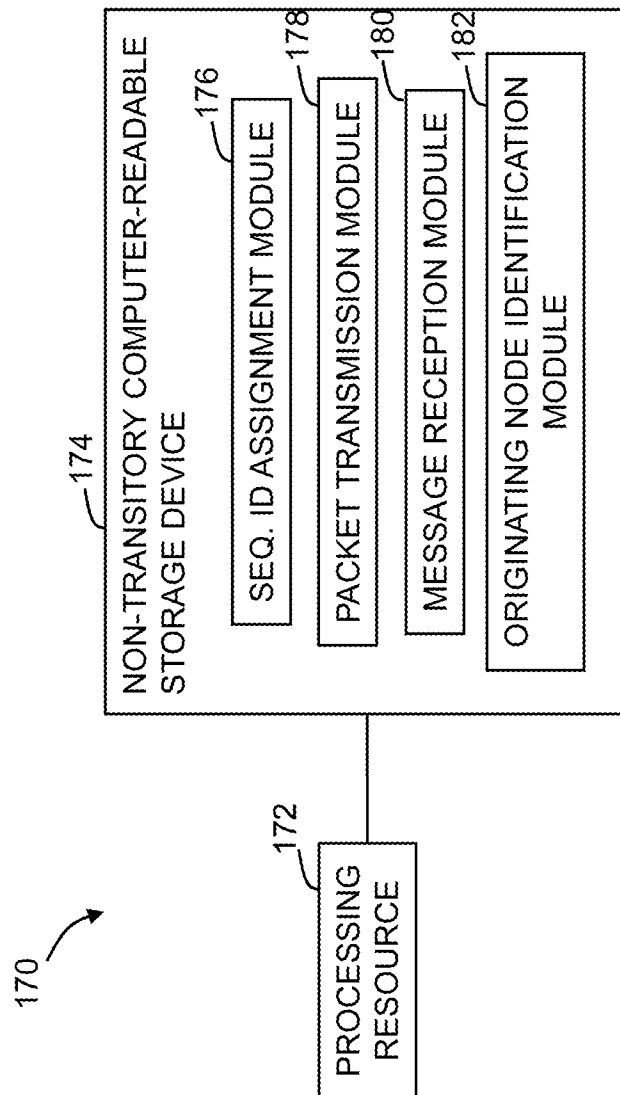


FIG. 5

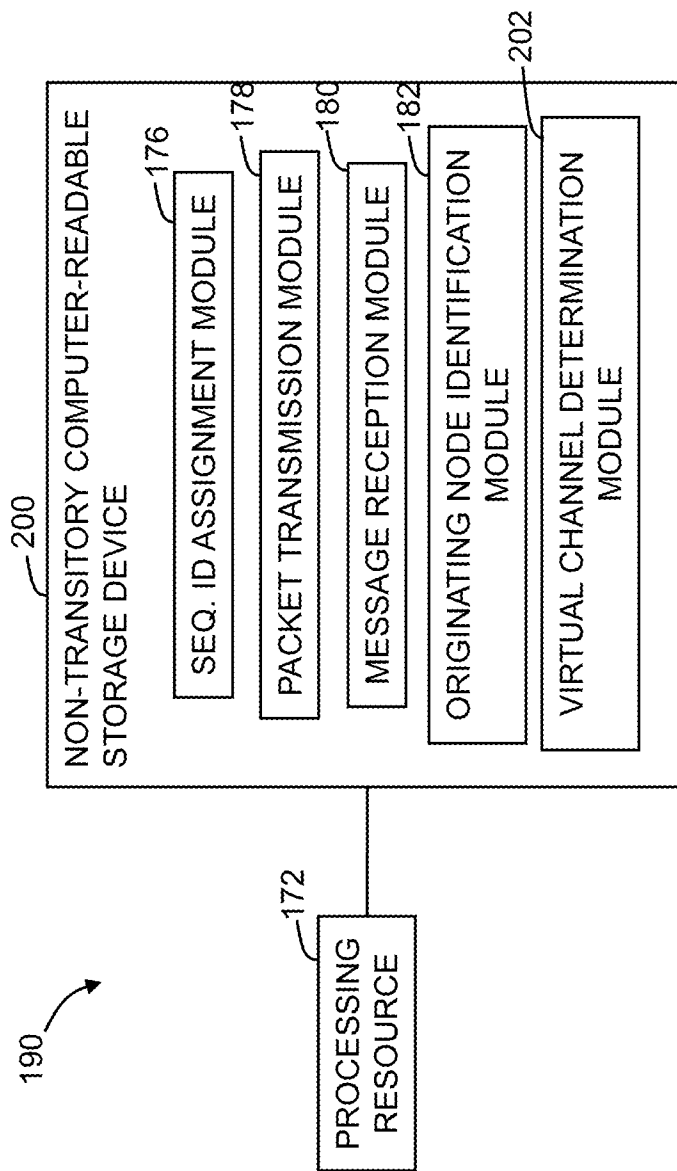


FIG. 6

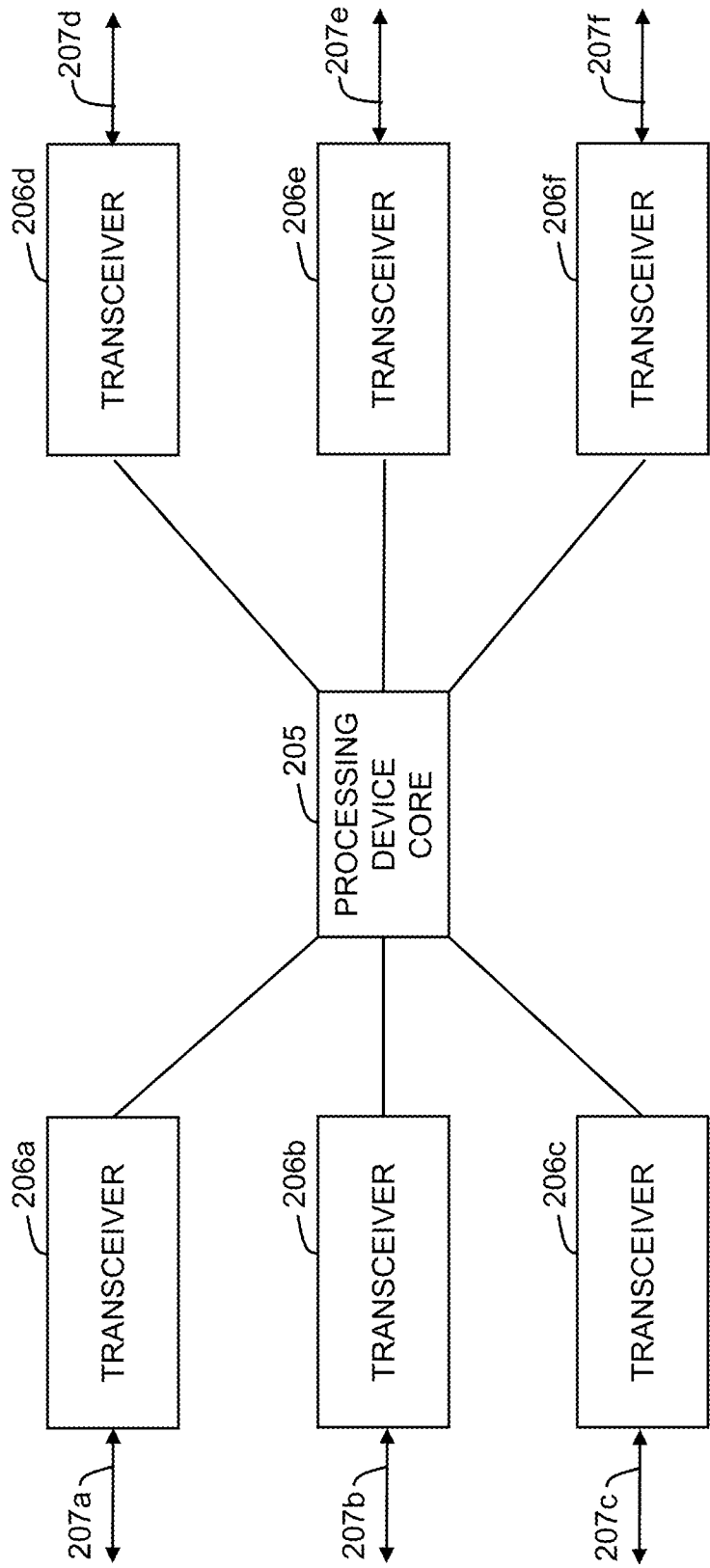


FIG. 7

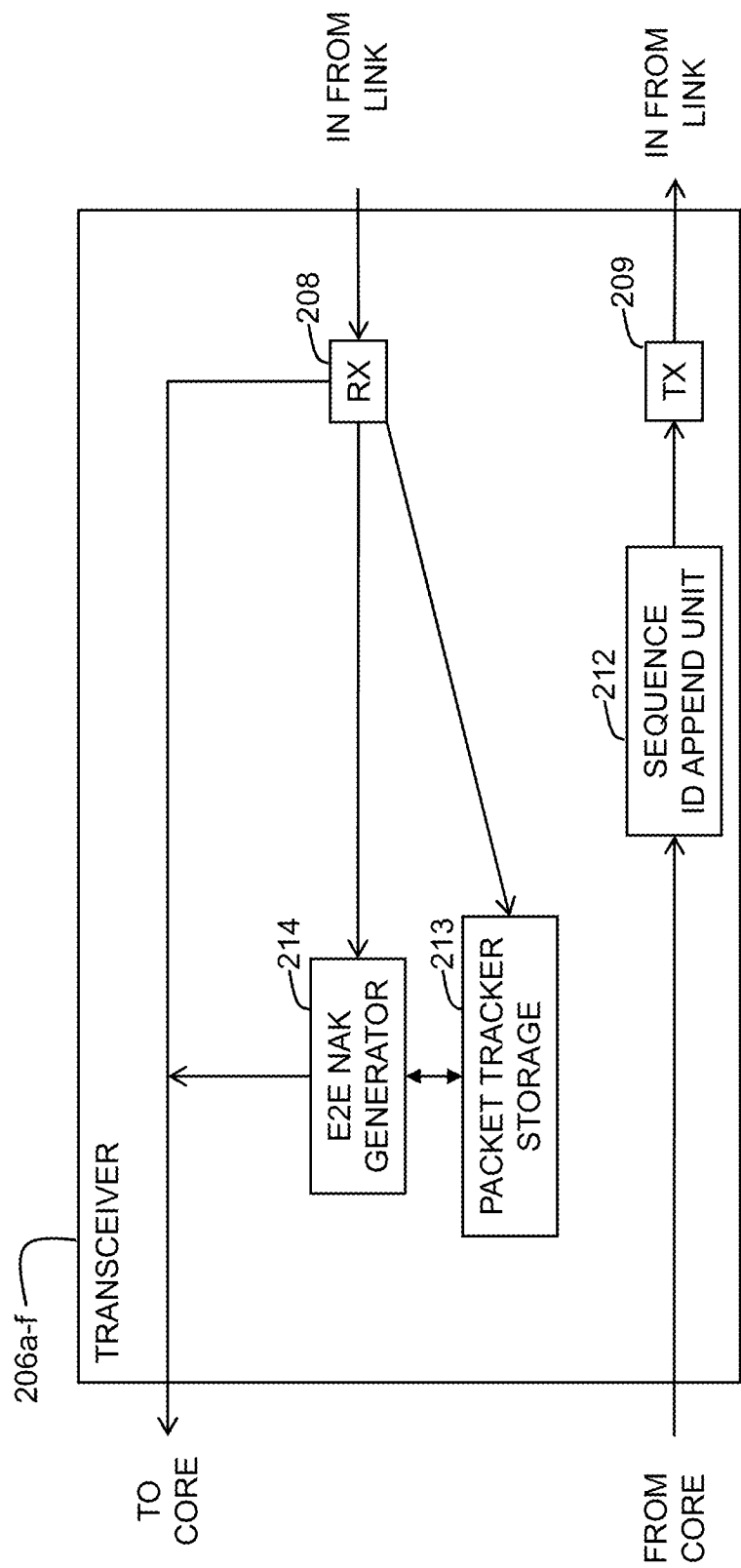


FIG. 8

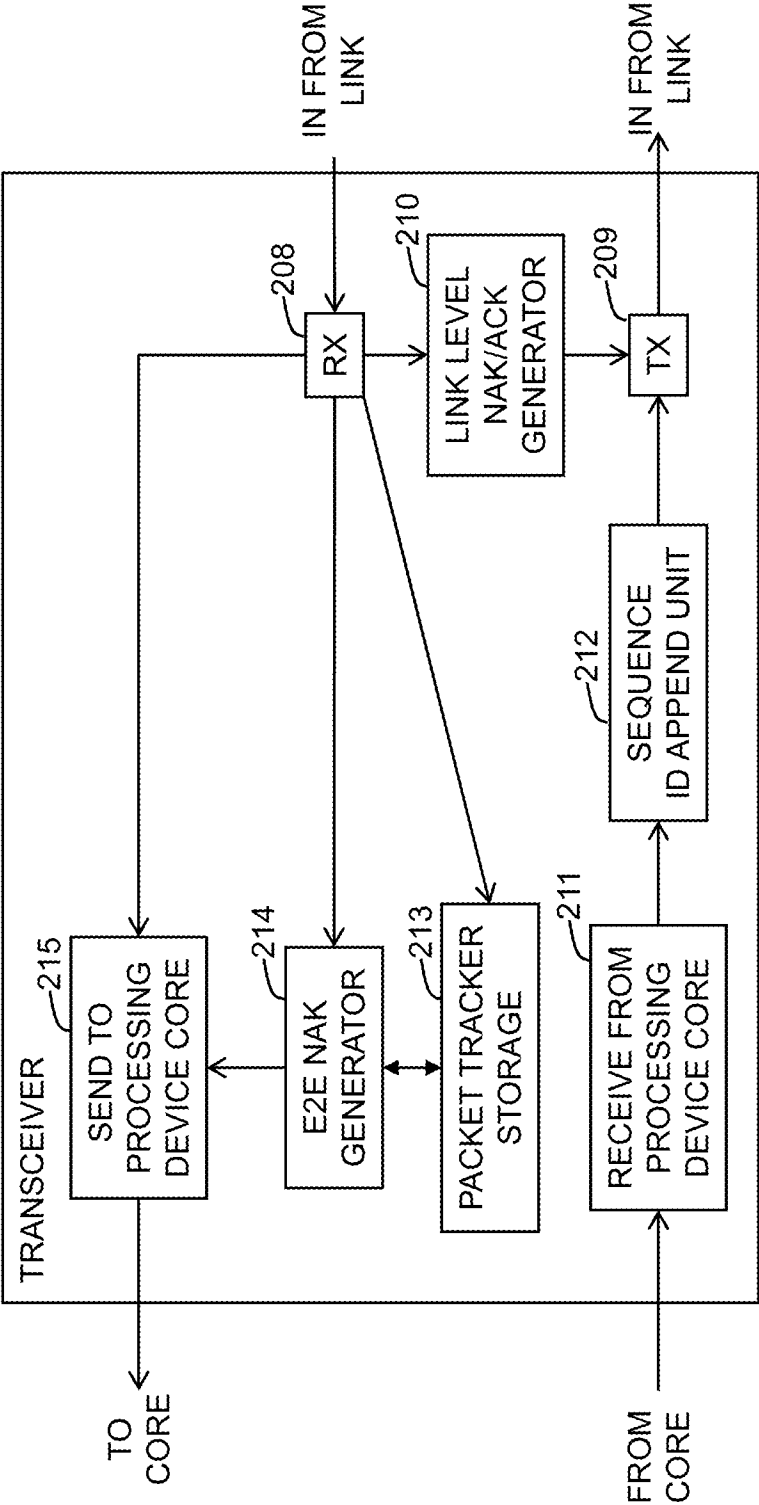


FIG. 9

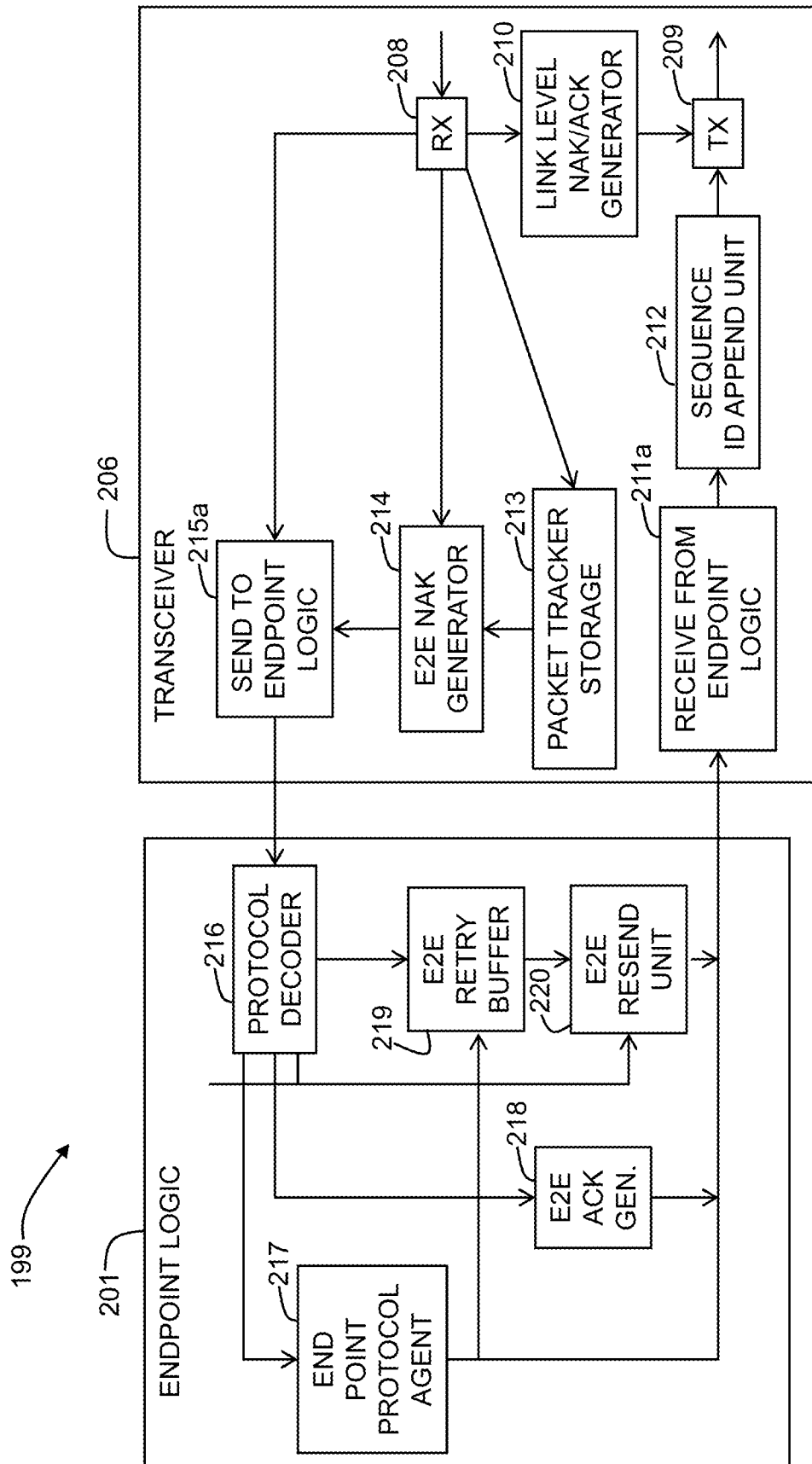


FIG. 10

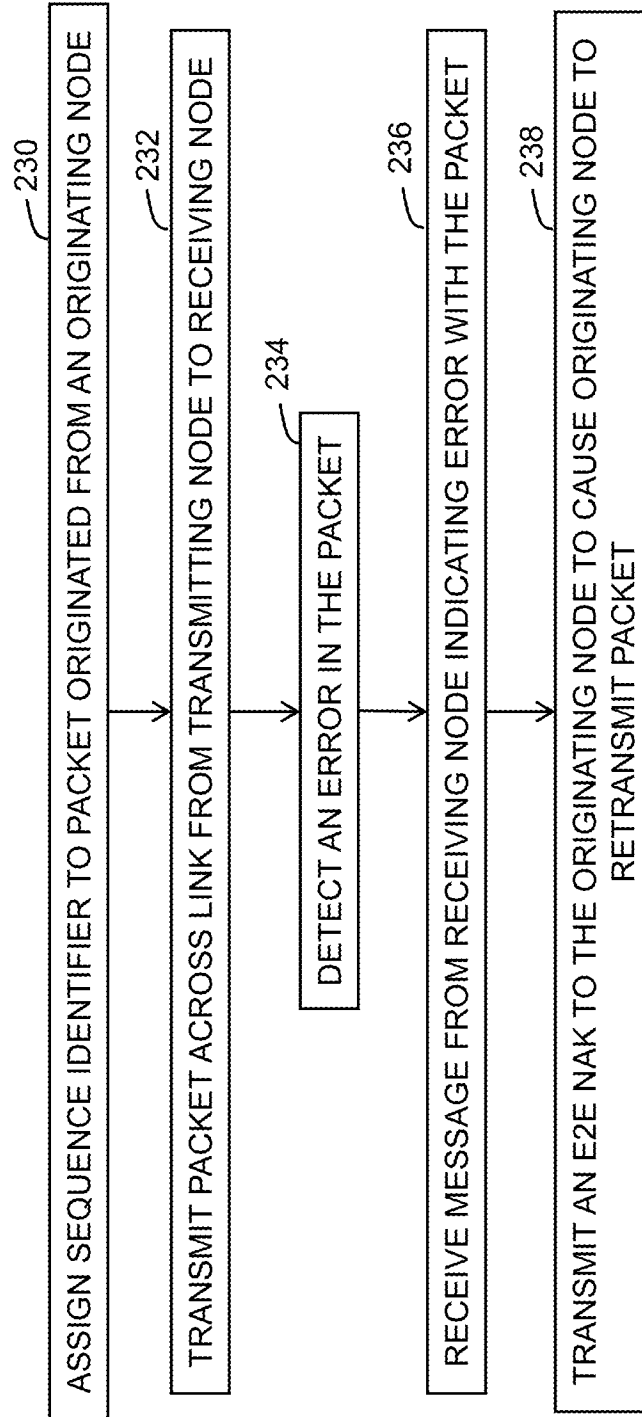


FIG. 11

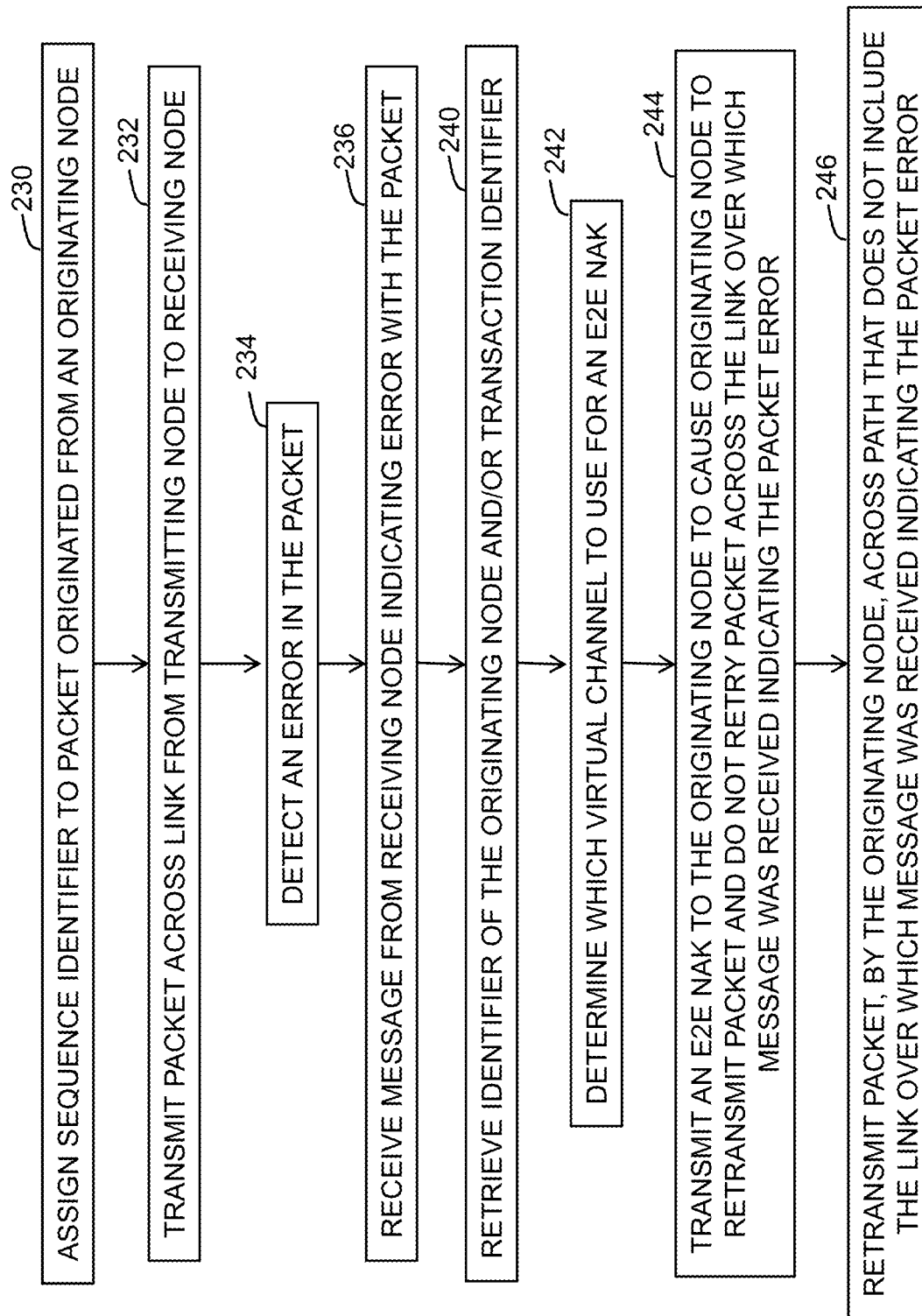


FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/062196**A. CLASSIFICATION OF SUBJECT MATTER****H04L 1/16(2006.01)i**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
H04L 1/16; H04L 1/00; H04Q 11/04; H04L 12/26; H04L 12/50; H04L 29/14; H04L 12/861Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & keywords: end-to-end negative acknowledgment, unique sequence identifier, link level NAK**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0213055 A1 (PARIMAL BAJPAI et al.) 23 August 2012 See paragraphs [0017]-[0053]; and figures 1-5C.	1-15
Y	US 2014-0269731 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 18 September 2014 See paragraphs [0031]-[0058]; and figures 3-7.	1-15
A	US 8169908 B1 (DAVID SLUITER et al.) 01 May 2012 See column 3, line 22 - column 6, line 56; and figures 1-6.	1-15
A	US 6181704 B1 (KEN DROTTAR et al.) 30 January 2001 See column 3, line 46 - column 7, line 60; and figures 1-4.	1-15
A	US 2013-0128884 A1 (CASIMER M. DECUSATIS et al.) 23 May 2013 See paragraphs [0024]-[0065]; and figures 1-10.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family


Date of the actual completion of the international search

30 June 2015 (30.06.2015)

Date of mailing of the international search report

30 June 2015 (30.06.2015)

Name and mailing address of the ISA/KR


 International Application Division
 Korean Intellectual Property Office
 189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
 Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KIM, Do Weon

Telephone No. +82-42-481-5560



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/062196

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0213055 A1	23/08/2012	US 2010-0232437 A1 US 8169914 B2 US 9007915 B2	16/09/2010 01/05/2012 14/04/2015
US 2014-0269731 A1	18/09/2014	CN 104052574 A US 2015-0016276 A1	17/09/2014 15/01/2015
US 8169908 B1	01/05/2012	None	
US 6181704 B1	30/01/2001	AT 346473 T AU 1999-52421 A1 CN 1135895 C CN 1233194 C CN 1324555 A CN 1510964 A CN 1764099 A CN 1764099 C DE 69934124 D1 DE 69934124 T2 EP 1108343 A1 EP 1108343 A4 EP 1108343 B1 HK 1041413 A1 WO 00-13455 A1	15/12/2006 21/03/2000 21/01/2004 21/12/2005 28/11/2001 07/07/2004 26/04/2006 30/07/2008 04/01/2007 11/10/2007 20/06/2001 25/02/2004 22/11/2006 03/12/2004 09/03/2000
US 2013-0128884 A1	23/05/2013	CN 103959722 A US 2013-0128721 A1 US 9007904 B2 US 9007905 B2 WO 2013-072801 A1	30/07/2014 23/05/2013 14/04/2015 14/04/2015 23/05/2013