



(51) International Patent Classification:

H04L 1/16 (2006.01) G06F 13/36 (2006.01)
G06F 12/0831 (2016.01) H04L 1/00 (2006.01)

(21) International Application Number:

PCT/GB2019/052400

(22) International Filing Date:

28 August 2019 (28.08.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

16/116,017 29 August 2018 (29.08.2018) US

(71) Applicant: **ARM LIMITED** [GB/GB]; 110 Fulbourn Road, Cambridge CB1 9NJ (GB).

(72) Inventors: **XU, Zheng**; 110 Fulbourn Road, Cambridge CB1 9NJ (GB). **JALAL, Jamshed**; 110 Fulbourn Road, Cambridge CB1 9NJ (GB).

(74) Agent: **TLIP LTD**; 14 King Street, Leeds, Yorkshire LS1 2HL (GB).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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.

(54) Title: METHOD AND APPARATUS FOR COHERENT INTERCONNECT RECOVERY WITH PROTOCOL LAYER RE-TRANSMISSION

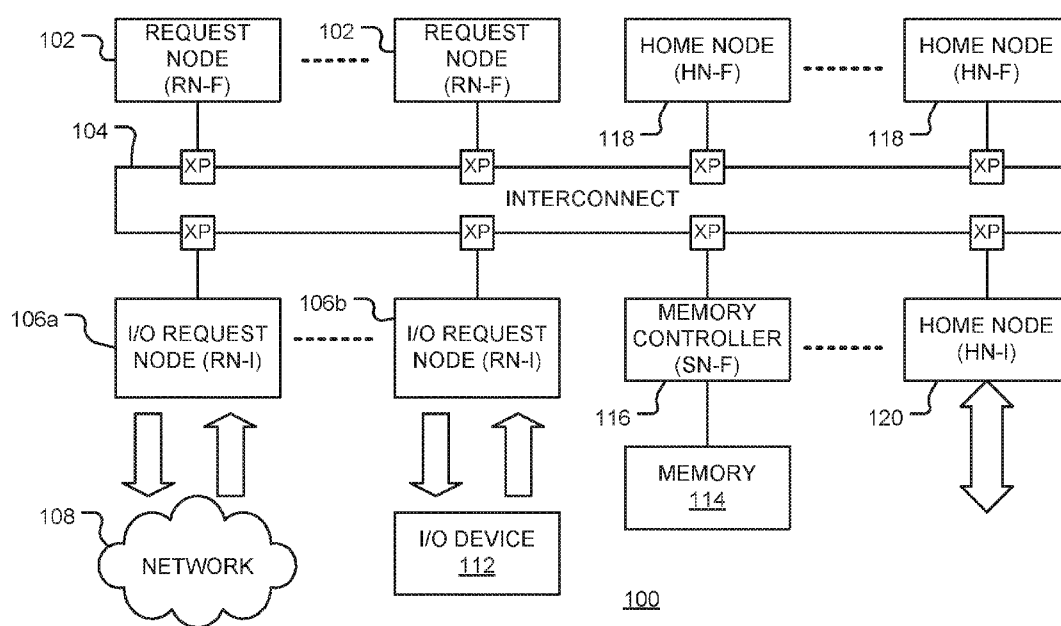


FIG. 1

(57) Abstract: A fault tolerant data processing network includes a number of nodes intercoupled through an interconnect circuit. The micro-architectures of the nodes are configured for sending and receiving messages via the interconnect circuit. In operation, a first Request Node sends a read request to a Home Node. In response, the Home Node initiates transmission of the requested data to the first Request Node. When the first Request Node detects that a fault has occurred, it sends a negative-acknowledgement message to the first Home Node. In response, the Home Node again initiates transmission of the requested data to the first Request Node. The requested data may be transmitted from a local cache of a second Request Node or transmitted by a Slave Node after being retrieved from a memory. The data may be transmitted to the first Request Node via the Home Node or directly via the interconnect.



(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).

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHOD AND APPARATUS FOR COHERENT INTERCONNECT RECOVERY WITH PROTOCOL LAYER RE-TRANSMISSION

BACKGROUND

[0001] A multi-processor data processing system may be arranged as an on-chip network with nodes of various types, such as processors, accelerators, IO, and memory connected via an interconnect fabric.

[0002] Data from a shared data resource, such as a memory, may be accessed by a number of different processors and copies of the data may be stored in a local caches for rapid access. A cache coherence protocol may be used to ensure that all copies are up to date. The protocol may involve the exchange of message between nodes of the network.

[0003] A coherence protocol however, may fail when a request or response is lost. Failure mechanism may involve silent data corruption, inconsistent or incoherent memory operations, dropped request/response messages, deadlock or live-lock, for example.

[0004] Accordingly, there exists a need for a coherence protocol that includes a mechanism for recovering from errors.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The accompanying drawings provide visual representations which will be used to more fully describe various representative embodiments and can be used by those skilled in the art to better understand the representative embodiments disclosed and their inherent advantages. In these drawings, like reference numerals identify corresponding elements.

[0006] **FIG. 1** is a block diagram of a data processing system, in accordance with various representative embodiments.

[0007] **FIG. 2** is a transaction flow diagram for a conventional data access in a data processing network.

[0008] **FIGs 3, 4 and 5** are transaction flow diagrams for data read transactions in a data processing network, in accordance with various representative embodiments.

[0009] **FIG. 6** is a transaction flow diagram for a data write-back transaction in a data processing network, in accordance with various representative embodiments.

[0010] **FIG. 7** is a transaction flow diagram for a data write transaction in a data processing network, in accordance with various representative embodiments.

[0011] **FIG. 8** is a flow chart of a method for coherent interconnect recovery with protocol layer re-transmission, in accordance with certain representative embodiments.

[0012] **FIGs 9-12** are transaction flow diagrams for data read transactions involving transaction replay in a data processing network, in accordance with various representative embodiments.

DETAILED DESCRIPTION

[0013] In accordance with certain representative embodiments of the present disclosure, there is provided a protocol layer re-transmission mechanism for a fault tolerant coherent interconnect architecture of a data processing network. The mechanism extends functional coherent protocols with error resilient response packets at selective points of the transaction flow to provide improved fault tolerance. The mechanism may be implemented by extending the structure of existing protocol-level retry hardware. Snoop and request replay are also introduced to reduce the fault recovery latency.

[0014] While this disclosure is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail specific embodiments, with the understanding that the present disclosure is to be considered as an example of the principles of the disclosure and not intended to limit the disclosure to the specific embodiments shown

and described. In the description below, like reference numerals are used to describe the same, similar or corresponding parts in the several views of the drawings.

[0015] In this document, relational terms such as first and second, top and bottom, and the like may be used solely to distinguish one entity or action from another entity or action without necessarily requiring or implying any actual such relationship or order between such entities or actions. The terms "comprises," "comprising," or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. An element preceded by "comprises ...a" does not, without more constraints, preclude the existence of additional identical elements in the process, method, article, or apparatus that comprises the element.

[0016] Reference throughout this document to "one embodiment", "certain embodiments", "an embodiment" or similar terms means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Thus, the appearances of such phrases or in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments without limitation.

[0017] The term "or" as used herein is to be interpreted as an inclusive or meaning any one or any combination. Therefore, "A, B or C" means "any of the following: A; B; C; A and B; A and C; B and C; A, B and C". An exception to this definition will occur only when a combination of elements, functions, steps or acts are in some way inherently mutually exclusive.

[0018] For simplicity and clarity of illustration, reference numerals may be repeated among the figures to indicate corresponding or analogous elements. Numerous details are set forth to provide an understanding of the embodiments described herein. The embodiments may be practiced without these details. In other instances, well-known methods, procedures, and components have not been described in detail to avoid obscuring the embodiments described. The description is not to be considered as limited to the embodiments described herein.

[0019] In accordance with certain representative embodiments of the present disclosure, there is provided a re-transmission mechanism for a fault tolerant coherent interconnect architecture. The mechanism is implemented in the protocol layer of a data processing network and extends existing functional coherent protocols by introducing error resilient response messages at selective point of the transaction flow to provide improved fault tolerance.

[0020] In a Network-on-a-Chip (NoC), error recovery may be implemented using a Protocol Layer re-transmission mechanism. The present disclosure extends this mechanism for use when the NoC is implemented using a coherent interconnect. In contrast to a standard, non-coherent, a coherent protocol requires Snoop tenure in addition to Request and Data tenure. Herein, 'tenure' refers to specific part of the transaction flow. For example, a coherent transaction flow may be partitioned into request tenure, snoop tenure and response tenure. In order to be able to re-transmit messages, the original message must be retained until it is known that no fault has occurred. When a fault is indicated, through a NACK message for example, the entire transaction flow may be restarted from the beginning of the transaction. For example, if an error occurs during snoop tenure, the transaction is restarted with the request tenure.

[0021] A data processing network, such as a Network-on-a-Chip, may include the following elements:

[0022] Fully-coherent Request Node (RN-F): A fully-coherent master device such as a CPU core or core cluster. A Request Node is a node that generates protocol messages, including reads and writes, to the network interconnect.

[0023] I/O-coherent Request Node (RN-I) Bridge: An input/output-coherent master device acting as a bridge to devices located behind the RN-I bridge.

[0024] Fully-coherent Home Node (HN-F): A device that is a Home Node for a designated region of memory, accepting coherent read and write requests from RN-Fs and generating snoops to all applicable RN-Fs in the system as required to support the coherency protocol. A Home Node receives protocol messages from RNs. Each address in the system has a Home which acts as the Point-of-Coherency (PoC) and Point of Serialization (PoS) for requests to that address. In a typical implementation, Homes for a range of addresses are grouped together as a Home Node. Each of these Home Nodes may include a system level cache and/or a snoop filter to reduce redundant snoops.

[0025] I/O Home Node (HN-I): A device that acts as a home-node for a memory mapped slave I/O subsystem, mainly responsible for ensuring proper ordering of requests sent into the slave I/O subsystem.

[0026] Fully-coherent Slave Node (SN-F): A fully-coherent device, such as a memory controller, that communicates with one or more HN-Fs that is solely a recipient of commands, limited to fulfilling simple read and write commands. A Slave Node receives and completes requests from Home Nodes and can be used for peripheral or main memory.

[0027] Protocol: The nodes exchange data and other information in messages according to a set of rules or procedures referred to as a protocol. Each message is sent as one or more information packets across the network interconnect fabric. The protocol may be

implemented in the micro-architecture of the nodes, for example. An example of a coherent protocol is the Arm[®] AMBA[®] coherent hub interface (CHI) protocol of Arm Limited.

[0028] Transaction: A transaction, such as a read or write transaction, typically involves the exchange of multiple messages as specified in a given protocol.

[0029] **FIG. 1** is a block diagram of a data processing system 100, in accordance with various representative embodiments. A number of processing core clusters 102 (referred to as Request Nodes (RN's)) are coupled to data resources via coherent interconnect 104. Data is received via input/output (I/O) requesting nodes (RN-I). In the example shown, RN-I 106a comprises a network interface controller (NIC) that receives data from network 108 and RN-I 106b receives data from I/O device 112. I/O device 112 may be coupled via a peripheral component interconnect express (PCIe) bus, direct memory access (DMA) unit, or network accelerator, for example. Data may be stored in one or more memory or storage devices 114 that are coupled to coherent interconnect 104 via one or more memory controllers or Slave Nodes 116. Home Nodes (HN) 118 and 120 may include system level caches. Each Home Node (HN) serves as a point of serialization and/or point of coherence for data stored at a given set of system addresses. A Home Node (HN-F), such as 118, may be a home for memory storage resources, while a Home Node (HN-I), such as 120, may provide an interface to memory mapped space or I/O resources. Data requested by a Request Node 102 may be retrieved from a system level cache of the HN or from a memory 114 via a memory controller 116. The memory controllers are referred to as Slave Nodes (SN's).

[0030] To avoid conflicts when multiple RNs try to access the same memory location, the Home Nodes 118 act as point of serialization, processing read requests and other transactions in a serial manner, such as first-come, first-served. Coherent interconnect 104 is used to transfer data over data (DAT) channels between nodes. In addition, a messaging protocol is used to control each access transaction, in which requests and responses are sent over request

(REQ) and response (RSP) channels in the interconnect. Finally, 'snoop' messages are sent over snoop (SNP) channels in the interconnect to ensure data coherence.

[0031] In accordance with an aspect of the disclosure, error information is combined with Data and Snoop responses between Home Nodes, Request Nodes and Slave Nodes of a network. In addition, acknowledgement (ACK) and repudiation or negative-acknowledgement (NACK) response packets are added to the protocol for certain events in the protocol transaction flow. In particular, methods and apparatus are disclosed for fault recovery in an interconnect circuit of a data processing network. In one embodiment, example, the method includes sending, by a first node of the data processing network, a protocol message to a second node of the data processing network via the interconnect circuit, and detecting, by the second node, when a fault has or has not occurred in the protocol message. The second node sends a negative-acknowledgement message to the first node when a fault has occurred in the protocol message and sends an acknowledgement message to the first node when no fault has occurred in the protocol message. Responsive to the negative-acknowledgement message, the first node may re-transmit the protocol message to the second node.

[0032] An acknowledgement or negative-acknowledgement message may be combined with an expected response to the protocol message. Alternatively, when an error occurs, an expected response, such as Data Response (DataResp) and/or Snoop Response (SnpResp) message, may be replaced with simple Error Response message.

[0033] In general, a coherence protocol is augmented to ensure that, at appropriate steps of a transaction flow, a message produces either an implicit response packet or an explicit ACK/NACK response packet before proceeding to next step of the transaction flow.

[0034] In one embodiment, for example, data (such a data provided in response to a snoop request) from a Request Node (RNF) is explicitly acknowledged in order for RNF to determine when to reliably deallocate data or change a cache state of the data. While it is not required to acknowledge data from a Slave Node (SNF) as there is no impact on coherency, it may be desirable to explicitly acknowledge all data for a consistent approach.

[0035] Snoop tenure in a Home Node is always in a closed loop with functional protocol. For example, a loop may be closed by a completion acknowledgement (CompACK) from the Request Node back to the Home Node. Thus, this is no need to send an additional an additional ACK/NACK to the Home Node. The Home Node will deallocate snoop entry if all responses and CompACK messages indicate that no errors have occurred.

[0036] Requests may be acknowledged implicitly, since each request will inherently cause a response. Further, a protocol layer acknowledgment, such as a CompACK message, is explicitly acknowledged in order for the Request Node to determine when to reliably deallocate the request entry.

[0037] **FIG. 2** is a transaction flow diagram for a conventional data access in a data processing network. In **FIG. 2**, vertical bars 202, 204 and 206 show time lines for a Request Node, Home Node and Slave Node, respectively, with time flowing from top to bottom. At time T1, the Request Node issues Read request 208 to the Home Node for data associated with a data address. If the request data is not in the cache of the Home Node, first slave read request 210 is sent to the appropriate Slave Node. The Slave Node sends the requested data 212 to the Request Node. All communications are transmitted via the interconnect fabric. The data may be transmitted to the Request Node in multiple data beats across the interconnect. Once all of the requested data has been received by the Request Node, a completion acknowledgment (CompAck) message 214 is sent from the Request Node to the

Home Node. Thus, the duration of the transaction is T1-T2 for the Request Node, T3-T4 for the Home Node, and T5-T6 for the Slave Node.

[0038] **FIG. 3** is a transaction flow diagram for a data read transaction in a data processing network, in accordance with various representative embodiments. In **FIG. 3**, vertical bars 302, 304 and 306 show time lines for a Request Node, Home Node and Slave Node, respectively, with time flowing from top to bottom. At time T1, the Request Node issues Read request 308 to the Home Node for data associated with a data address. The Home Node sends an acknowledgement (ACK) in response 310 when the Read request is received without error and the Home Node is able to process the request, otherwise a repudiation or negative-acknowledgement (NACK) is sent in response 310. If the requested data is not in the cache of the Home Node, first slave Read request 312 is sent to the appropriate Slave Node. The Slave Node sends an acknowledgement or negative-acknowledgement in ACK/NACK response 314 to the Home Node and sends the requested data to the Request Node in Data response 316. All communications are transmitted via the interconnect fabric. The data may be transmitted to the Request Node in multiple data beats across the interconnect. Once the requested data has been received by the Request Node (possibly in multiple data beats), the Request Node sends an acknowledgement or negative-acknowledgement to the Slave Node in ACK/NACK response 318 and sends a completion acknowledgment (CompAck) message 320 to the Home Node. Finally, on receipt of message 320, the Home Node sends an acknowledgement or negative-acknowledgement in ACK/NACK response 322 to the Request Node. In this manner, an explicit acknowledgement or negative acknowledgement is sent in response to each step of the read transaction. The duration of the transaction is T1-T2 for the Request Node, T3-T4 for the Home Node, and T5-T6 for the Slave Node. While these durations may be in excess of

corresponding durations for a conventional Read transaction, the protocol depicted in **FIG. 3** provides improved fault tolerance.

[0039] If the requested data is present in the cache of the Home Node, the data may be sent directly from the Home Node to the Request node and no messages are exchanged with the Slave Node.

[0040] **FIG. 4** is a further transaction flow diagram for a data read transaction in a data processing network, in accordance with various representative embodiments. In the transaction shown, the Home Node sends a repudiation in NACK response 402, indicating an error in the original Read request or that the Home Node is unable to process the request. In response to NACK response 402, the Request Node re-transmits the Read request in message 404. Optionally, in response to the resent Read request, the Home Node may send ACK response 406 indicating successful receipt of request 404. However, ACK response 406 may be omitted since ACK response 322 indicates success of the read operation to the Request Node. The read operation then continues as in **FIG. 3**, assuming that all subsequent messages are received without error. However, ACK response 408 may be omitted since, again, ACK response 322 indicates success of the complete read operation to the Request Node. When the requested data is not in the cache of the Home Node, first slave Read request 408 is sent to the appropriate Slave Node. The Slave Node sends an acknowledgement in ACK response 410 to the Home Node and sends the requested data to the Request Node in Data response 412. Once the requested data has been received by the Request Node, the Request Node sends an acknowledgement to the Slave Node in ACK response 414 and sends a completion acknowledgment (CompAck) message 416 to the Home Node. Finally, on receipt of message 416, the Home Node sends an acknowledgement in ACK response 418 to the Request Node.

[0041] When the requested data is present in one or more local caches of Request Nodes, the Home Node sends snoop messages to those Request Nodes to maintain data coherency in the network. Faults occurring in these snoop messages, or the responses to them, can result in a loss of data coherence in the network. **FIG. 5** is a further transaction flow diagram for a data read transaction in a data processing network, in accordance with various representative embodiments. In **FIG. 5**, vertical bars 502, 504, 506 and 508 show time lines for a first Request Node (RN 1), Home Node (HN), second Request Node (RN 2) and third Request Node (RN-3), with time flowing from top to bottom. The Request Node RN-1 issues Read request 510 to the Home Node for data associated with a data address. The Home Node sends an acknowledgement (ACK) in response 512 if the Read request is received without error and the Home Node is able to process the request, otherwise a repudiation (NACK) in response 310. The Home Node sends a first Snoop request 514 to the second Request Node and sends a second Snoop request 516 to the third Request Node. The second Request Node responds with response 518 to the Home Node that is a combination of an ACK/NACK response and a Snoop response. The third Request Node responds with response 520 to the Home Node that is an acknowledgement in combination with a Snoop response and the requested data. The Home Node acknowledges receipt of the data in ACK/NACK response 522 to the third Request Node. The Home Node then forwards to the data to the first Request Node in Data message 524 and the first Request Node acknowledges receipt of the data in combined ACK/NACK and completion acknowledgement message 526. More generally, Snoop messages are sent to each Request Node that is indicated to have a copy of the data in its local cache.

[0042] The protocol depicted in **FIG. 5** combines conventional responses with ACK/NACK response in response 518, 520 and 526. This combination reduces the number of additional messages required to provide fault tolerance.

[0043] In general, the protocols shown in **FIGs 3-5** are combined depending upon where local copies of data are stored in the data processing system. In one example, for a simple read operation, the transaction flow is:

- a) RNF sends Read request (ReadReq) to HNF (HNF responds with ACK/NACK)
- b) HNF sends Snoop request (SnpReq) to each RNF
- c) Each RNF responds with Snoop response (SnpResp) to HNF (response combined with ACK/NACK)
- d) HNF sends Read no snoop (ReadNoSnp) to SNF (SNF responds with ACK/NACK)
- e) SNF sends data and notification of completion (CompData) back to RNF (response combined with ACK/NACK)
- f) RNF sends acknowledgment and notification of completion (CompAck) back to HNF (response combined with ACK/NACK).
- g) HNF send ACK back to RNF.

[0044] In another example, for a read operation where a snoop response includes partial data, the transaction flow is:

- a) RNF sends ReadReq (HNF responds with NACK if error, optionally responds with ACK if no error)
- b) HNF sends SnpReq to each RNF
- c) Each RNF responds with SnpResp to HNF (response combined with ACK/NACK)
- d) The Owner RNF can send Snoop response with partial data (SnpRespDataPtl) back to HNF (response combined with ACK/NACK)

- e) HNF sends CompData back to RNF (response combined with ACK/NACK)
- f) RNF sends CompAck back to HNF (response combined with ACK/NACK).
- g) HNF send ACK back to RNF.

[0045] **FIG. 6** is a transaction flow diagram for a data write-back transaction in a data processing network, in accordance with various representative embodiments. In **FIG. 6**, vertical bars 602 and 604 show time lines for a Request Node and a Home Node respectively, with time flowing from top to bottom. The Request Node issues Write-back request 606 to the Home Node indicating a request to write-back data to a specified address in memory. Home Node responds with message 608. When no error occurs, message 608 may be a combination of a Write Data Buffer credit message (CompDBID) and an acknowledgement (ACK) message, acknowledging the Write-back request. The Write Data Buffer credit message (CompDBID) indicates the previous read/write requests and coherence management operations have completed and that resources are available at the Home Node to receive the data to be written back to memory. When an error occurs, message 608 may simply signal the error through a NACK message. When no error occurs, the Request Node sends the data to the Home Node in Data message 610 and the Home Node acknowledges or repudiates receipt of the data in ACK/NACK message 612. Whenever an error occurs it is indicated by a NACK message and the message may be resent. For example, when message 612 is a NACK message, the data message 610 may be resent.

[0046] **FIG. 7** is a transaction flow diagram for a data write transaction in a data processing network, in accordance with various representative embodiments. In **FIG. 7**, vertical bars 702 and 704 show time lines for a Request Node and a Home Node respectively, with time flowing from top to bottom. The Request Node issues Write request 706 to the Home Node indicating a request to write-back data to address A. Home Node responds with message

708. When no error occurs, message 708 may be a combination of a Write Data Buffer credit message (DBID A) (indicating that a write buffer is available at the Home Node to receive the data to be written to memory) and an acknowledgement (ACK) message, acknowledging the Write request. When an error occurs, message 708 may simply signal the error through a NACK message. When no error occurs, the Request Node sends the data to the Home Node in Data message 710 and the Home Node responds with message 712. Message 712 is a combination of a completion message and a message that acknowledges or repudiates receipt of the data. Whenever an error occurs it is indicated by a NACK message and the message may be re-transmitted. For example, when message 708 is a NACK message, Write request 706 may be re-transmitted.

[0047] In accordance with certain embodiments, an additional check-point is introduced in the protocol after each tenure (request, snoop and data) such that an Error Response can be sent to requestor earlier if there is any error in either a Data request and/or a Snoop request/Snoop response handshake. Depending on the response packet information, the Request Node can choose to replay the transaction or not.

[0048] A Request Node may be implemented with a protocol level retry micro-architecture so as to reduce the required data buffer size in the Home Node and to minimize round trip latency between Request Node and Home Node buffer allocation.

[0049] Protocol level retry micro-architecture typically retains the request address, attributes and other data information at the requester. In one embodiment, the protocol-level retry hardware structure is utilized to provide error resiliency and enable implementation of a low cost fault tolerant coherency interconnect.

[0050] A Snoop retry (SnpRetry) message may be introduced for the Home Node (instead of Requestor) at the point of coherency in order to reduce recovery latency further when an error is detected. When errors are detected in either the Snoop request or Snoop response packets,

error information may be included in the Snoop response packet and sent back to the Home Node. Depending on the information in the Snoop response packet information, a Home Node can choose to replay the Snoop tenure again. It is noted that the Home Node possesses at the minimum information (such as address and attribute information) to enable Snoop replay, thereby minimizing the additional cost of replaying the Snoop Requests. It is also noted that, since a Snoop request is fully tagged, it could be sent to different Request Node and combined at the Home Node out of order. However, for each cache line address, there should be no more than one Snoop in the system at any given time. This reduces the number of replays required for error resiliency.

[0051] The technique described above does not affect the functional correctness of a data processing system with Snoop requests included. It is only necessary to replay the transactions which failed Snoop tenure. However, since the additional ACK/NACK response packets in the protocol place additional constraints on when the requestor can release its buffers, there is a trade-off between performance and fault resiliency.

[0052] **FIG. 8** is a flow chart of a method 800 for coherent interconnect recovery with protocol layer re-transmission in accordance with certain embodiments. Following start block 802, a message sent by a first node coupled to the coherent interconnect, is received by a second node coupled to the coherent interconnect at block 804. In an error or fault is detected, as depicted by the positive branch from decision block 806, the message is repudiated by sending a negative-acknowledgement (NACK) response at block 808 back to the first node. Upon receiving the NACK message, the first node determines if the transaction should be retried. If it is determined that the transaction should not be retried, as depicted by the negative branch from decision block 810, the method terminates at block 812. Otherwise, as depicted by the positive branch from decision block 810, the original message is re-transmitted by the first node at block 814 and flow returns to block 804.

[0053] When no error or fault is detected, as depicted by the negative branch from decision block 806, flow continues to decision block 816. When, in the underlying non-fault tolerant protocol, no response is expected from the message received, an acknowledgement is sent by the second node back to the first node at block 818 to acknowledge the message. Conversely, when, in the underlying non-fault tolerant protocol, a response is expected from the message received, an acknowledgement is combined with the expected response at block 820 and the combined response is sent by the second node back to the first node. In a further embodiment, the NACK message sent at block 808 is combined with any expected response. In this manner, any detected fault in a transaction step is signaled to the message sender, enabling the message sender to re-transmit the message if so desired. This provides a mechanism by which faults may be overcome.

[0054] **FIGs 9-12** are transaction flow diagrams for data read transactions involving transaction replay in a data processing network, in accordance with various representative embodiments.

[0055] **FIG. 9** is a transaction flow diagrams for a data read transaction, where the requested data is stored in a cache of a Request Node. In **FIG. 9**, vertical bars 902, 904 and 906 show time lines for a first Request Node (RN 1), Home Node (HN) and second Request Node (RN 2), respectively, with time flowing from top to bottom. The Request Node RN-1 issues Read request 908 to the Home Node for data associated with a data address 'A'. The Home Node sends a Snoop request 910 to the second Request Node. The second Request Node responds with response 912 to the Home Node that is an acknowledgement in combination with a Snoop response and the requested data. The Home Node acknowledges receipt of the data in acknowledgement 914 to the second Request Node. The Home Node then forwards to the data to the first Request Node in Data message 912. In this example, an error occurs and the first Request Node sends a no-acknowledgement in message 918. In response, the Home

Node resends the data in message 920. This time, the data is received without error and the first Request Node responds with combined ACK and completion acknowledgement message 922. Finally, the Home Node acknowledges completion of the transaction associated with message 924. In this manner, the data processing system recovers from the error associated with message 916 and normal operation may continue.

[0056] **FIG. 10** is a further transaction flow diagrams for a data read transaction where the requested data is stored in a memory and accessed via a Slave Node. In **FIG. 10**, vertical bars 1002, 1004 and 1006 show time lines for a Request Node, Home Node and Slave Node, respectively, with time flowing from top to bottom. The Request Node issues a Read request 1008 to the Home Node for data associated with a data address. The Home Node sends a read request 1010 to the Slave Node for the data. The Slave Node retrieves the data from memory and sends the data to the Request Node in message 1012. In this example, an error occurs and the first Request Node sends a no-acknowledgement in message 1014 to the Home Node. In response, the Home Node resends the read request to the Slave Node in message 1016. The data is resent to the Request Node in message 1018. This time, the data is received without error and the first Request Node responds with combined ACK and completion acknowledgement message 1020 to the Home Node. Finally, the Home Node acknowledges completion of the transaction in message 1022. In this manner, the data processing system recovers from the error associated with message 1012 and normal operation may continue.

[0057] **FIG. 11** is a transaction flow diagrams for a data read transaction, where the requested data is stored in a cache of a Request Node and transferred via Direct Cache Transfer (DCT). In **FIG. 11**, vertical bars 1102, 1104 and 1106 show time lines for a first Request Node (RN 1), Home Node (HN) and second Request Node (RN 2), respectively, with time flowing from top to bottom. Request Node RN-1 issues Read request 1108 to the

Home Node for data associated with a data address 'A'. The Home Node sends a Snoop request 1110 to the second Request Node. The second Request Node responds with response 1112 to the Home Node that is an acknowledgement in combination with a Snoop response indicating that the second Request Node has a copy of the requested data in its local cache. The second Request Node sends the requested data directly to the first Request Node in message (1114), bypassing the Home Node. In this example, an error occurs and the first Request Node sends a no-acknowledgement in message 1116 to the Home Node. In response, the Home Node resends the Snoop request in message 1118. Again, the second Request Node sends a snoop response to the Home Node in message 1120 and sends the data to the first Request Node in message 1122. This time, the data is received without error and the first Request Node responds with combined ACK and completion acknowledgement message 1124 to the Home Node. The requested data has been retrieved from a local cache of the second Request Node (RN-2). If the requested data is in a 'dirty' coherence state, that is, the data does not match corresponding data stored in the memory, the Home Node sends data acknowledgement message 1126 (DataAck) to RN-2. This message signals to RN-2 that transaction is complete and that RN-2 may deallocate storage for the requested data. Finally, the Home Node acknowledges completion of the transaction associated with message 1128. In this manner, the data processing system recovers from the error associated with message 1114 and normal operation may continue. In **FIG. 11**, P1 and P2 denote the tenure periods for the second Request Node.

[0058] **FIG. 12** is a further transaction flow diagrams for a data read transaction, where the requested data is stored in a cache of a Request Node and transferred via Direct Cache Transfer (DCT). In **FIG. 12**, vertical bars 1202, 1204 and 1206 show time lines for a first Request Node (RN 1), Home Node (HN) and second Request Node (RN 2), respectively, with time flowing from top to bottom. Messages 1206-1222 follow the same sequence as in

FIG. 11. The Request Node RN-1 issues Read request 1208 to the Home Node for data associated with a data address 'A'. The Home Node sends a Snoop request 1210 to the second Request Node. The second Request Node responds with response 1212 to the Home Node that is an acknowledgement in combination with a Snoop response indicating that the second Request Node has a copy of the requested data in its local cache. The second Request Node sends the requested data directly to the first Request Node in message (1214), bypassing the Home Node. In this example, an error occurs, and the first Request Node sends a no-acknowledgement in message 1216 to the Home Node. In response, the Home Node resends the Snoop request in message 1218. Again, the second Request Node sends a snoop response to the Home Node in message 1220 and resends the data to the first Request Node in message 1222. In this example, the first Request Node determines that the transaction should be cancelled and sends cancellation request 1224 to the Home Node. The Home Node acknowledges the cancellation in message 1226. Subsequently, the first Request Node may reissue the read request in message 1230. Again, the Home Node snoops the second Request Node in message 1232, the second Request Node responds in message 1234 and resends the data directly the first Request Node in message 1236, in a Direct Cache Transfer (DCT). Completion is acknowledged in messages 1238 and 1240. The requested data has been retrieved from a local cache of the second Request Node (RN-2). If the requested data is in a 'dirty' coherence state, that is, the data does not match corresponding data stored in the memory, the Home Node sends data acknowledgement message 1242 (DataAck) to RN-2. This message signals to RN-2 that transaction is complete and that RN-2 may deallocate storage for the requested data. In **FIG. 12**, P1, P2 and P3 denote the tenure periods for the second Request Node. In the manner, a Request Node may cancel an incomplete transaction and thus avoid deadlock type errors.

[0059] The disclosed mechanisms may be implemented using hardware components such as special purpose hardware, dedicated hard wired logic, and/or dedicated processors or their equivalents. In particular the mechanisms may be implemented in the micro-architecture of the data processing network. The mechanisms may be implemented by a state machine that has no stored program instructions, or in one or more application specific integrated circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic. Of course, a combination of these approaches could be used.

[0060] The data processing network disclosed above, or a component thereof, may be described by instructions of a hardware description language. Such instructions may be stored in a non-transient computer readable medium, for example. Such components may be combined with other components to form one or more integrated circuits.

[0061] Similarly, the data processing network disclosed above, or a component thereof, may be described by a netlist of components and connections stored in a non-transient computer readable medium, for example.

[0062] Those skilled in the art will appreciate that the processes described above can be implemented in any number of variations without departing from the present disclosure. For example, the order of certain operations carried out can often be varied, additional operations can be added or operations can be deleted without departing from the disclosure. Such variations are contemplated and considered equivalent.

[0063] Accordingly, some aspects and features of the disclosed embodiments are set out in the following paragraphs. The embodiments described above and summarized below are combinable.

[0064] In one embodiment, a method for fault recovery in an interconnect circuit of a data processing network includes sending, by a first node of the data processing network, a

protocol message to a second node of the data processing network via the interconnect circuit, detecting, by the second node, when a fault has or has not occurred in the protocol message; sending, by the second node, a negative-acknowledgement message to the first node when a fault has occurred in the protocol message; and responsive to the negative-acknowledgement message, the first node re-transmitting the protocol message to the second node.

[0065] In one embodiment, the method further includes sending, by the second node, an acknowledgement message to the first node when no fault has occurred in the protocol message.

[0066] In one embodiment, the method further includes the second node combining the acknowledgement message with an expected response to the protocol message.

[0067] In one embodiment, the method further includes the second node combining the negative-acknowledgement message with an expected response to the protocol message.

[0068] In one embodiment, the method further includes the second node replacing an expected response to the protocol message with the negative-acknowledgement message when a fault has occurred in the protocol message.

[0069] In one embodiment, the first node is a Request Node, the second node is a Home Node and the protocol message is a Read request or a completion acknowledgement response.

[0070] In one embodiment, the first node is a Request Node, the second node is a Home Node and the protocol message is a Write-back request, and the method further includes combining, by the Home Node, the acknowledgement or negative-acknowledgement message with a completion and write data buffer credit message.

[0071] In one embodiment, the first node is a Request Node, the second node is a Home Node and the protocol message is a Write request, and the method further includes combining, by the Home Node, the acknowledgement or negative-acknowledgement message with a write data buffer credit message.

[0072] In one embodiment, the first node is a Home Node, the second node is a Slave Node and the protocol message is a Read request.

[0073] In one embodiment, the first node is a Request Node, the second node is a Home Node and the protocol message is data to be written to a memory, and the method further includes combining, by the Home Node, an acknowledgement message or a negative-acknowledgement message with a completion message.

[0074] In one embodiment, the first node is a Home Node, the second node is a Request Node and the protocol message is a Snoop request, and the method further includes combining, by the Request Node, an acknowledgement message or the negative-acknowledgement message with a Snoop response.

[0075] In one embodiment, the first node is a Home Node, the second node is a Request Node and the protocol message is a Snoop request, and the method further includes combining, by the Request Node, an acknowledgement message or the negative-acknowledgement message with a Snoop response and a Data response.

[0076] In one embodiment, the first node is a Home Node, the second node is a Request Node and the protocol message is a Data response, and the method further includes combining, by the Request Node, an acknowledgement message or the negative-acknowledgement message with a completion and acknowledgement response.

[0077] In one embodiment, the first node is a Slave Node, the second node is a Request Node and the protocol message is a Data response.

[0078] In one embodiment, a fault tolerant data processing network includes an interconnect circuit; and a plurality of nodes intercoupled through the interconnect circuit, the plurality of nodes including a first node, where a micro-architecture of the first node is configured for sending a protocol message to the interconnect circuit; and a second node, where a micro-architecture of the second node is configured for receiving the protocol message from the interconnect circuit, detecting when a fault has or has not occurred in the protocol message, sending a negative-acknowledgement message to the first node when a fault is detected occurred in the protocol message, where the first node is configured for re-transmitting the protocol message to the second node responsive to the negative-acknowledgement message.

[0079] In one embodiment, the micro-architecture of the second node is further configured to send an acknowledgement message to the first node when no fault is detected in the protocol message.

[0080] In one embodiment, the micro-architecture of the first node is further configured for buffering the protocol message at least until a corresponding acknowledgement message or a negative-acknowledge message is received from the second node.

[0081] In one embodiment, the micro-architecture of the second node is further configured for combining the acknowledgement message or a negative-acknowledge message with an expected response to the protocol message.

[0082] In one embodiment, the expected response includes a Snoop response, by a Slave Node, to a Snoop request, a completion acknowledgement response, by a Request Node, to a Data message, a completion and write data buffer credit response, by a Home Node, to a Write-back request, a write data buffer credit response, by a Home Node, to a Write request, or a completion response, by a Home Node, to a Data message,

[0083] In one embodiment, the micro-architecture of the second node is further configured for replacing an expected response to the protocol message combining the negative-acknowledge message when the fault is detected in the protocol message.

[0084] In one embodiment, a non-transient computer readable medium contains instructions of a hardware description language descriptive of the fault tolerant data processing network of the fault tolerant data processing network.

[0085] In one embodiment, a non-transient computer readable medium contains a netlist description of the fault tolerant data processing network of the fault tolerant data processing network.

[0086] In one embodiment, a method for fault recovery in an interconnect circuit of a data processing network includes sending, by a first Request Node of the data processing network, a read request to a Home Node of the data processing network via the interconnect circuit; responsive to the read request, the Home Node initiating transmission of data requested by the read request to the first Request Node; detecting, by the first Request Node, when a fault has or has not occurred in a response to the read request; sending, by the first Request Node, a negative-acknowledgement message to the first Home Node when a fault has occurred in the response to the read request; and responsive to the negative-acknowledgement message, the Home Node again initiating transmission of the data requested by the read request to the first Request Node.

[0087] In one embodiment, the Home Node initiating transmission of data requested by the read request to the first Request Node includes sending, by the Home Node, a snoop request to a second Request Node of the data processing network; retrieving, by the second Request Node, the requested data from a local cache of the second Request Node; sending, by the second Request Node, the requested data to the Home Node; and sending, by the Home Node, the requested data to the first Request Node.

[0088] In one embodiment, the method further includes following successful receipt of the requested data by the first Request Node: when requested data is in a 'dirty' coherence state in the local cache of the second Request Node: sending, by the Home Node, a data acknowledgment message to the second Request Node; and responsive to the data acknowledgment message, the second Request Node deallocating storage for the requested data at the second Request Node.

[0089] In one embodiment, the Home Node initiating transmission of data requested by the read request to the first Request Node includes sending, by the Home Node, a read request to a Slave Node of the data processing network; retrieving, by the Slave Node, the requested data from a memory; and sending, by the Slave Node, the requested data to the first Request Node.

[0090] In one embodiment, the Home Node initiating transmission of data requested by the read request to the first Request Node includes sending, by the Home Node, a snoop request to a second Request Node of the data processing network; and sending, by the second Request Node, the requested data to the first Request Node.

[0091] In one embodiment, the method further includes following successful receipt of the requested data by the first Request Node: sending, by the first Request Node, a completion data acknowledgment message to the Home Node to the second Request Node; and the Home Node deallocating a snoop table entry for the requested data.

[0092] In one embodiment, the method further includes following successful receipt of the requested data by the first Request Node: when requested data is in a 'dirty' coherence state in the local cache of the second Request Node: sending, by the Home Node, a data acknowledgment message to the second Request Node; and responsive to the data acknowledgment message, the second Request Node deallocating storage for the requested data at the second Request Node.

[0093] In one embodiment, the method further includes when the first Request Node has detected a fault in response to one or more read requests: sending, by the first Request Node, a transaction cancellation request to the Home Node; sending, by the Home Node to the first Request Node, an acknowledgment of the cancellation request.

[0094] In one embodiment, the method further includes subsequent to receiving the acknowledgement of the cancellation request, the first resending the read request.

[0095] The various representative embodiments, which have been described in detail herein, have been presented by way of example and not by way of limitation. It will be understood by those skilled in the art that various changes may be made in the form and details of the described embodiments resulting in equivalent embodiments that remain within the scope of the appended claims.

Claims

1. A fault tolerant data processing network, comprising:
an interconnect circuit; and
a plurality of nodes, intercoupled through the interconnect circuit, including:
a first node having a micro-architecture configured for sending a protocol message to the interconnect circuit, and
a second node having a micro-architecture configured for receiving the protocol message from the interconnect circuit, detecting when a fault has or has not occurred in the protocol message, and sending a negative-acknowledgement message to the first node when a fault is detected occurred in the protocol message,
where the first node is configured for re-transmitting the protocol message to the second node responsive to the negative-acknowledgement message.
2. The fault tolerant data processing network of claim 1, where the micro-architecture of the second node is further configured for sending an acknowledgement message to the first node when no fault is detected in the protocol message.
3. The fault tolerant data processing network of claim 1 or claim 2, where the micro-architecture of the first node is further configured for buffering the protocol message at least until a corresponding acknowledgement message or a negative-acknowledge message is received from the second node.
4. The fault tolerant data processing network of any preceding claim, where the micro-architecture of the second node is further configured for combining the acknowledgement message or a negative-acknowledge message with an expected response to the protocol message.

5. The fault tolerant data processing network of claim 4, where the expected response includes:

- a Snoop response, by a Slave Node, to a Snoop request;
- a completion acknowledgement response, by a Request Node, to a Data message;
- a completion and write data buffer credit response, by the Home Node, to a Write-back request;
- a write data buffer credit response, by the Home Node, to a Write request; or
- a completion response, by the Home Node, to a Data message,

6. The fault tolerant data processing network of claim 4 or claim 5, where the micro-architecture of the second node is further configured for replacing an expected response to the protocol message combining the negative-acknowledge message when the fault is detected in the protocol message.

7. A non-transient computer readable medium containing instructions of a hardware description language descriptive of the fault tolerant data processing network of any preceding claim.

8. A non-transient computer readable medium containing a netlist description of the fault tolerant data processing network of any of claims 1 to 6.

9. A method for fault recovery in an interconnect circuit of a data processing network, the method comprising:

- sending, by a first Request Node of the data processing network, a read request to a Home Node of the data processing network via the interconnect circuit;
- responsive to the read request, initiating, by the Home Node, a first transmission of data requested by the read request to the first Request Node;

detecting, by the first Request Node, when a fault has or has not occurred in a response to the read request;

sending, by the first Request Node, a negative-acknowledgement message to the first Home Node when a fault has occurred in the response to the read request; and

responsive to the negative-acknowledgement message, initiating, by the Home Node, a second transmission of the data requested by the read request to the first Request Node.

10. The method of claim 9, where the initiating the first or second transmission of data requested by the read request to the first Request Node includes:

sending, by the Home Node, a snoop request to a second Request Node of the data processing network;

retrieving, by the second Request Node, the requested data from a local cache of the second Request Node;

sending, by the second Request Node, the requested data to the Home Node; and

sending, by the Home Node, the requested data to the first Request Node.

11. The method of claim 10, further comprising, following successful receipt of the requested data by the first Request Node:

when requested data is in a 'dirty' coherence state in the local cache of the second Request Node:

sending, by the Home Node, a data acknowledgment message to the second Request Node; and

responsive to the data acknowledgment message, deallocating, by the second Request Node, storage for the requested data at the second Request Node.

12. The method of any of claims 9 to 11, where the initiating the first or second transmission of data requested by the read request to the first Request Node includes:

sending, by the Home Node, a read request to a Slave Node of the data processing network;

retrieving, by the Slave Node, the requested data from a memory; and

sending, by the Slave Node, the requested data to the first Request Node.

13. The method of any of claims 9 to 12, where the initiating the first or second transmission of data requested by the read request to the first Request Node includes:

sending, by the Home Node, a snoop request to a second Request Node of the data processing network; and

sending, by the second Request Node, the requested data to the first Request Node.

14. The method of claim 13, further comprising, following successful receipt of the requested data by the first Request Node:

sending, by the first Request Node, a completion data acknowledgment message to the Home Node to the second Request Node; and

deallocating, by the Home Node, a snoop table entry for the requested data.

15. The method of claim 14, further comprising, following successful receipt of the requested data by the first Request Node:

when requested data is in a 'dirty' coherence state in the local cache of the second Request Node:

sending, by the Home Node, a data acknowledgment message to the second Request Node; and

responsive to the data acknowledgment message, deallocating, by the second Request Node, storage for the requested data at the second Request Node.

16. The method of any of claims 9 to 15, further comprising, when the first Request Node has detected a fault in response to one or more read requests:

sending, by the first Request Node, a transaction cancellation request to the Home Node; and

sending, by the Home Node to the first Request Node, an acknowledgment of the cancellation request.

17. The method of claim 16, further comprising:

subsequent to receiving the acknowledgement of the cancellation request, resending the read request.

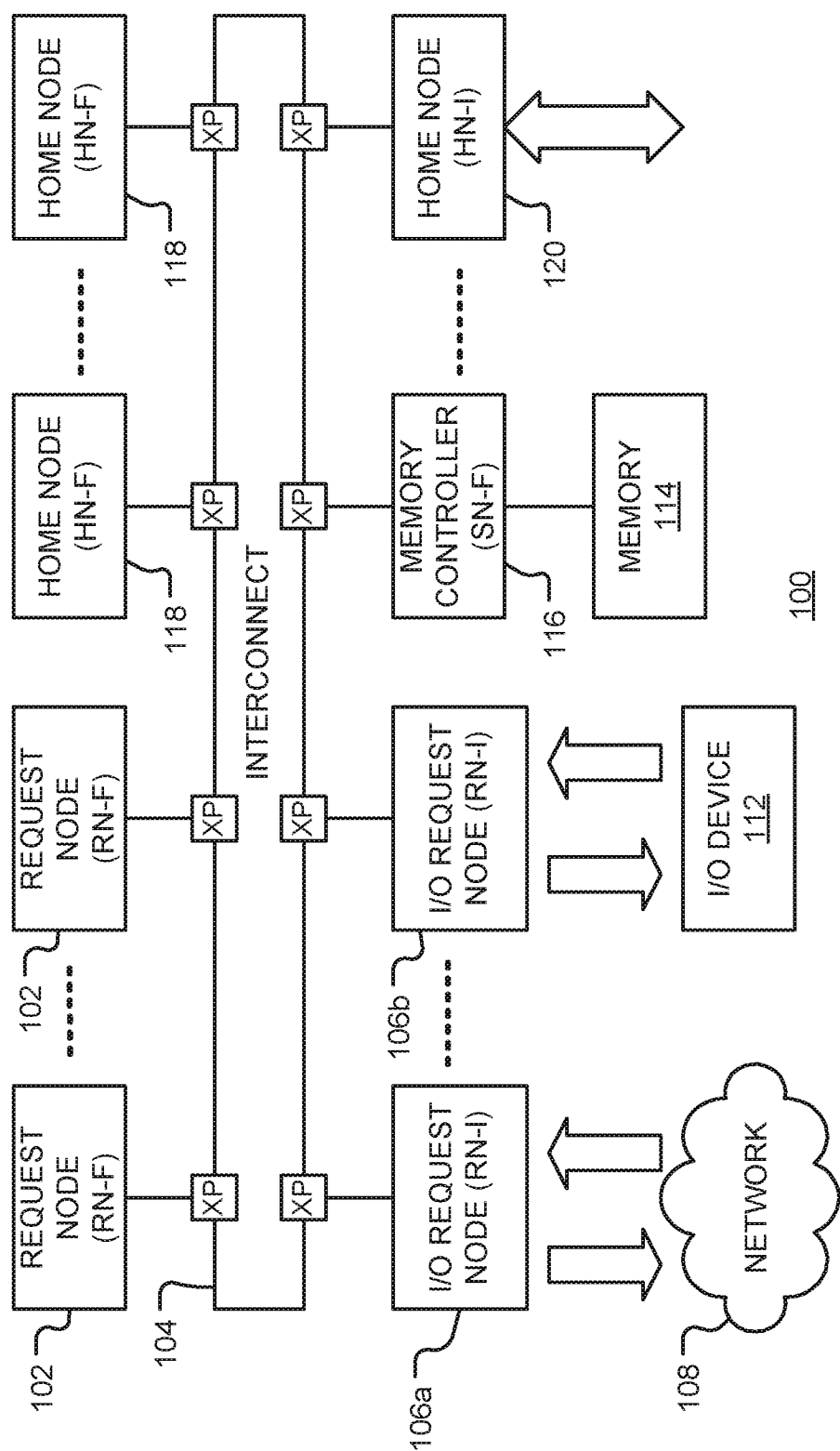
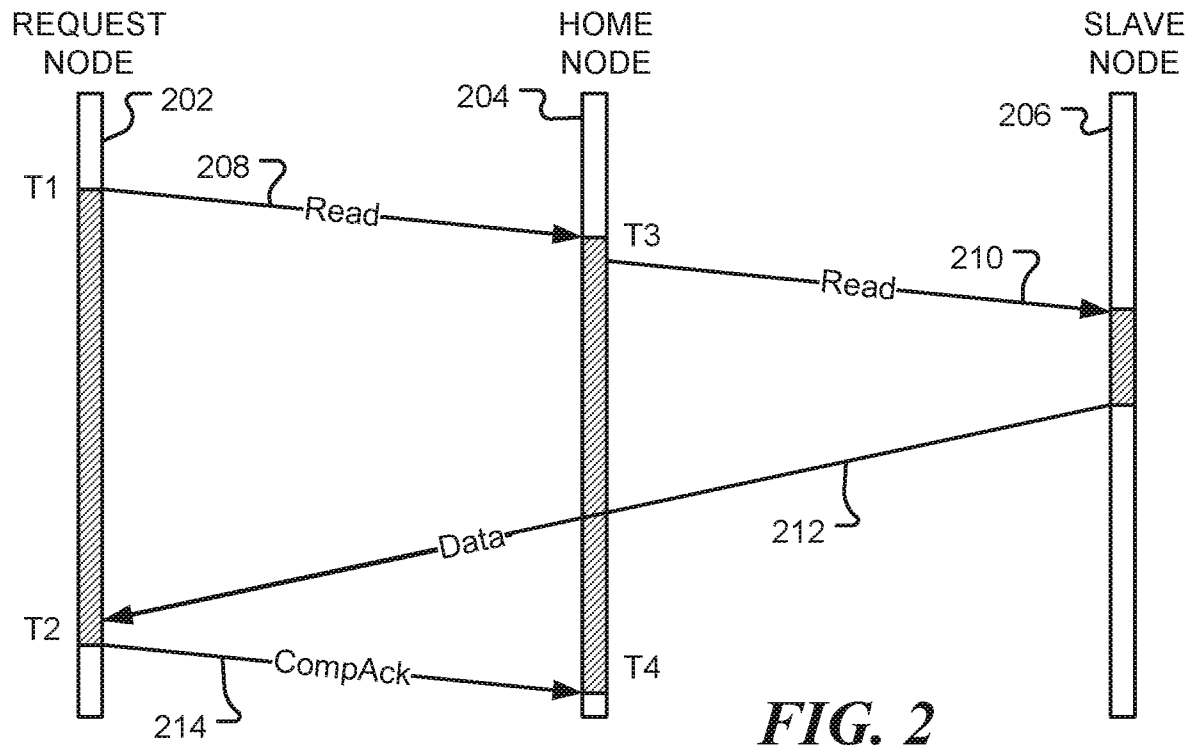
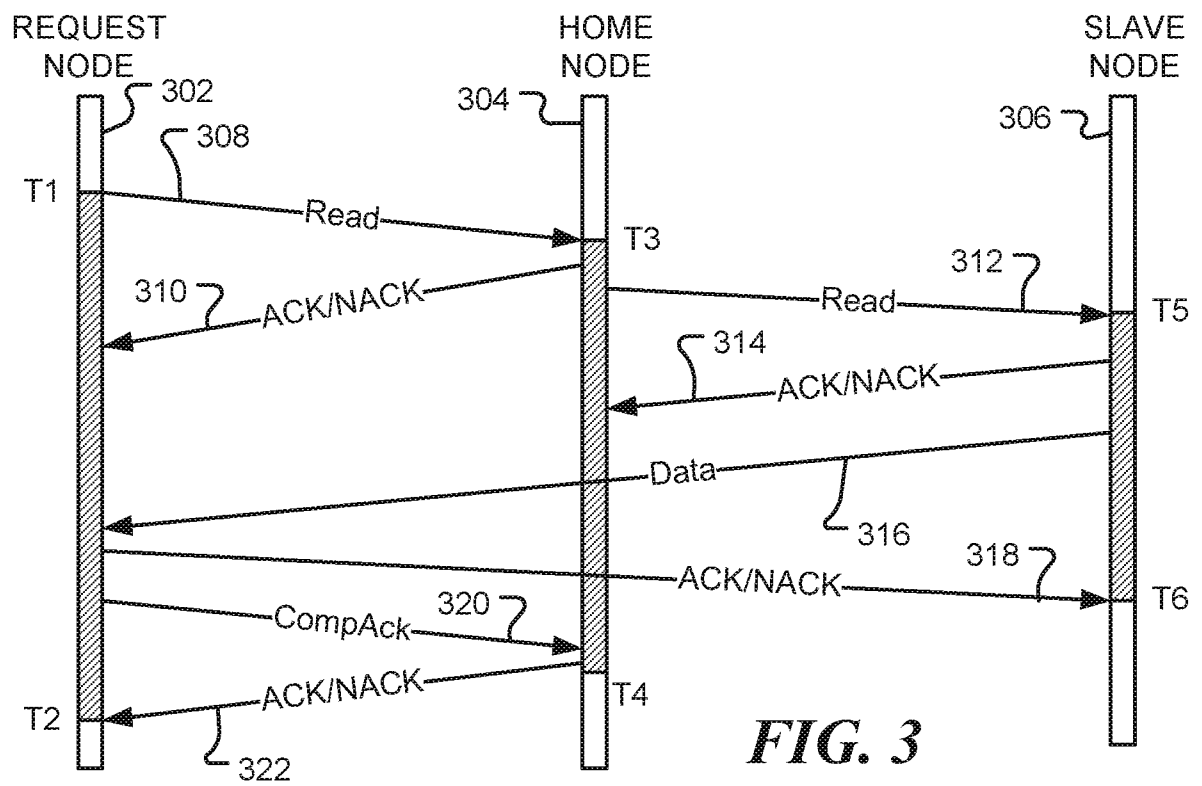
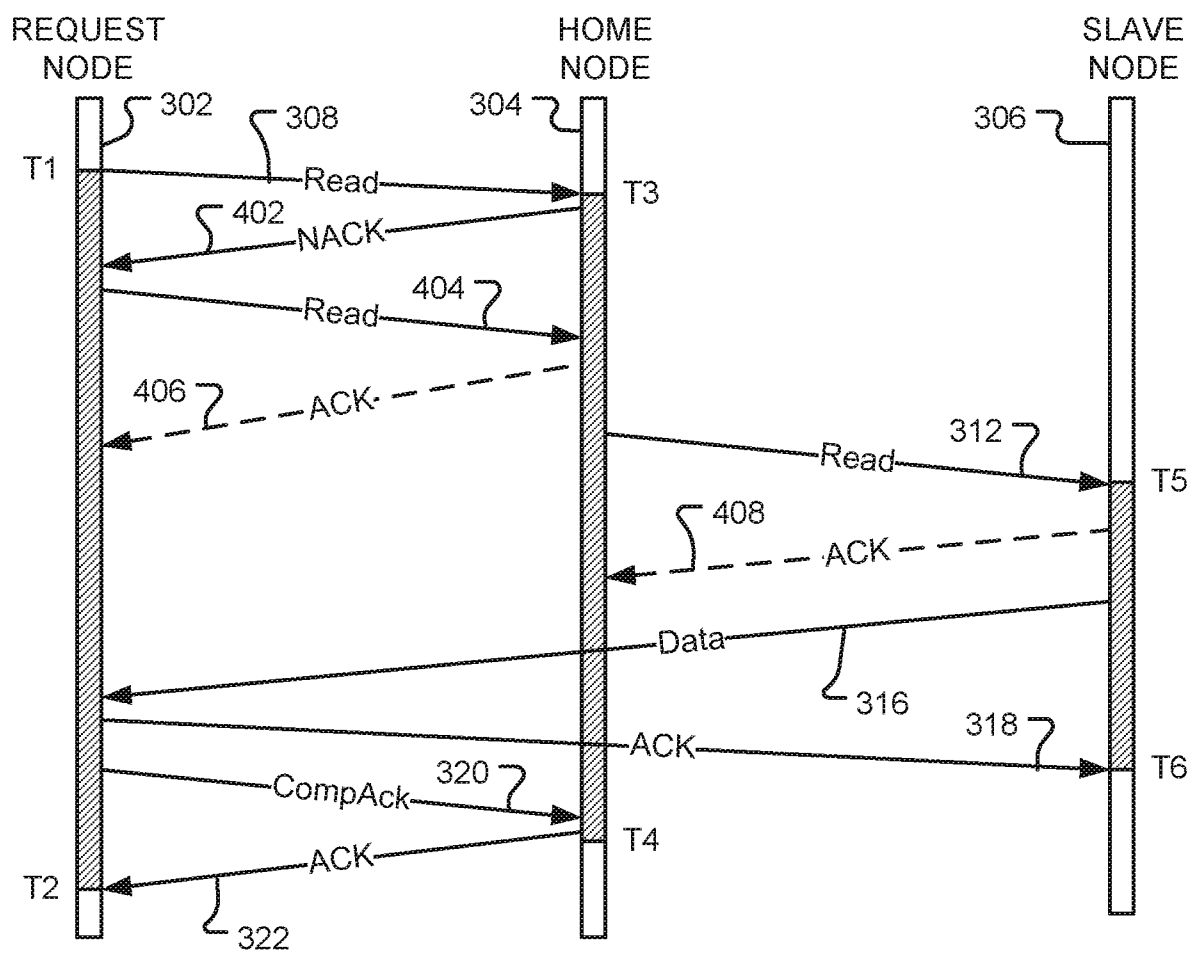


FIG. 1

2/10

**FIG. 2****FIG. 3**

3/10

**FIG. 4**

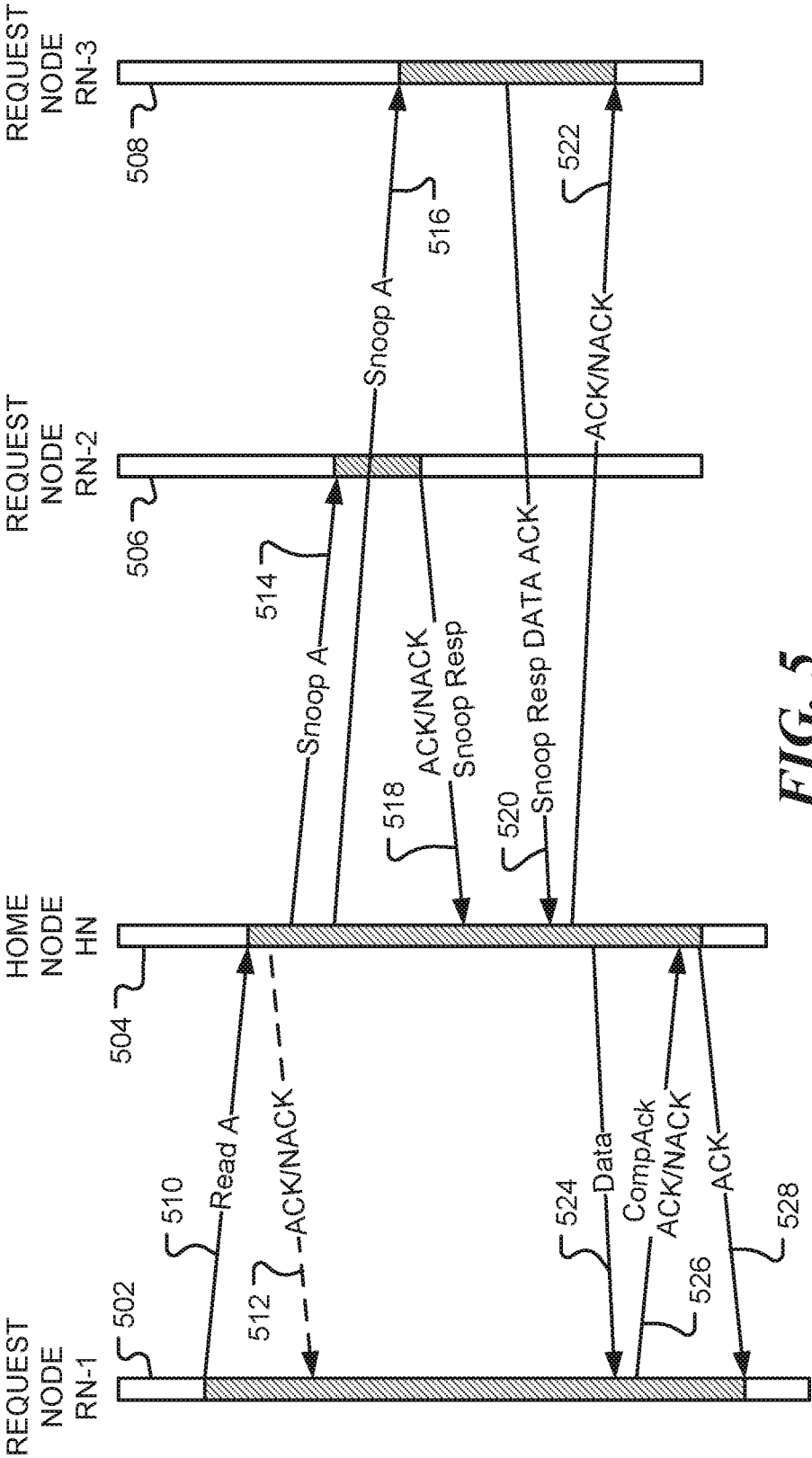
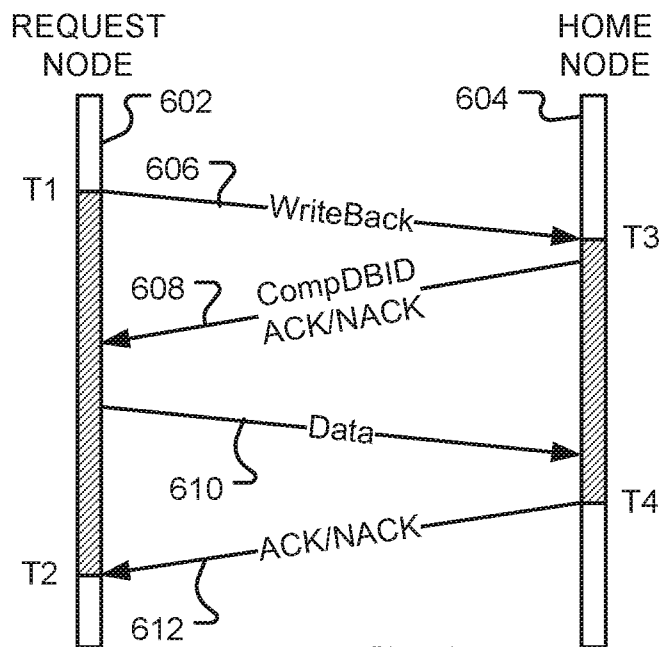
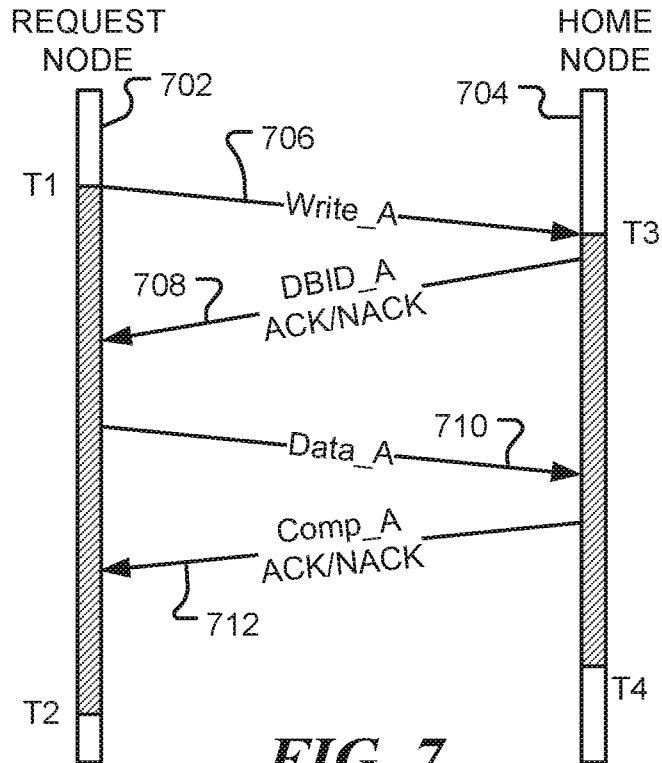
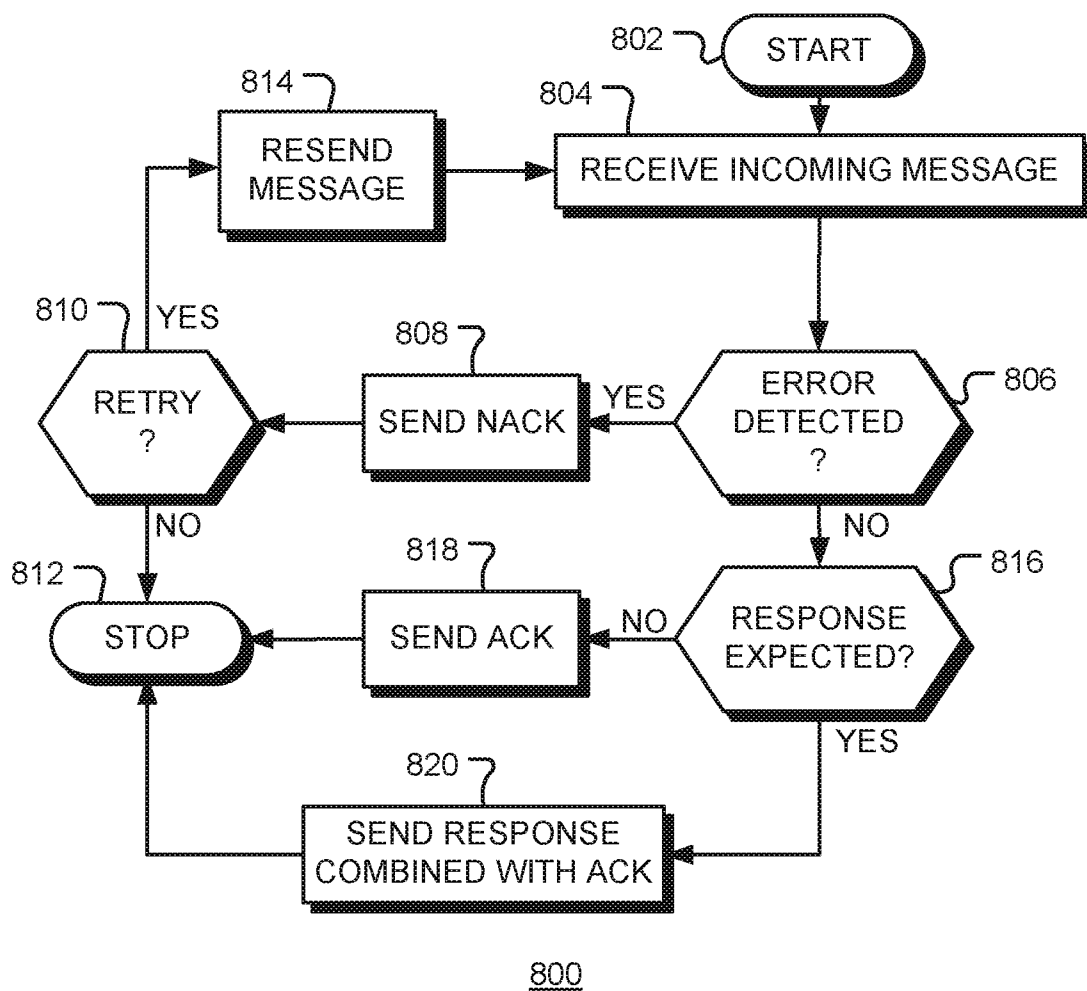


FIG. 5

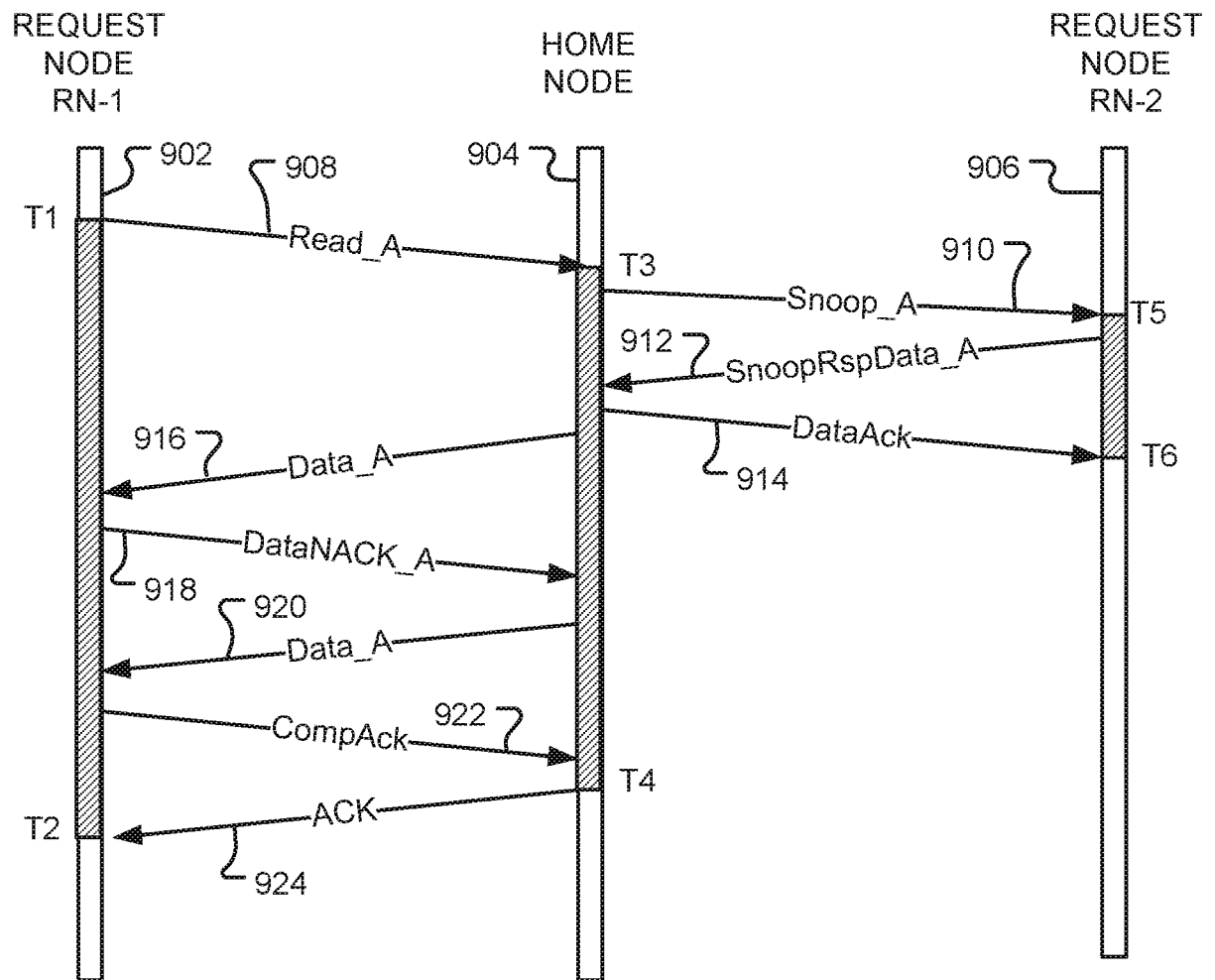
5/10

**FIG. 6****FIG. 7**

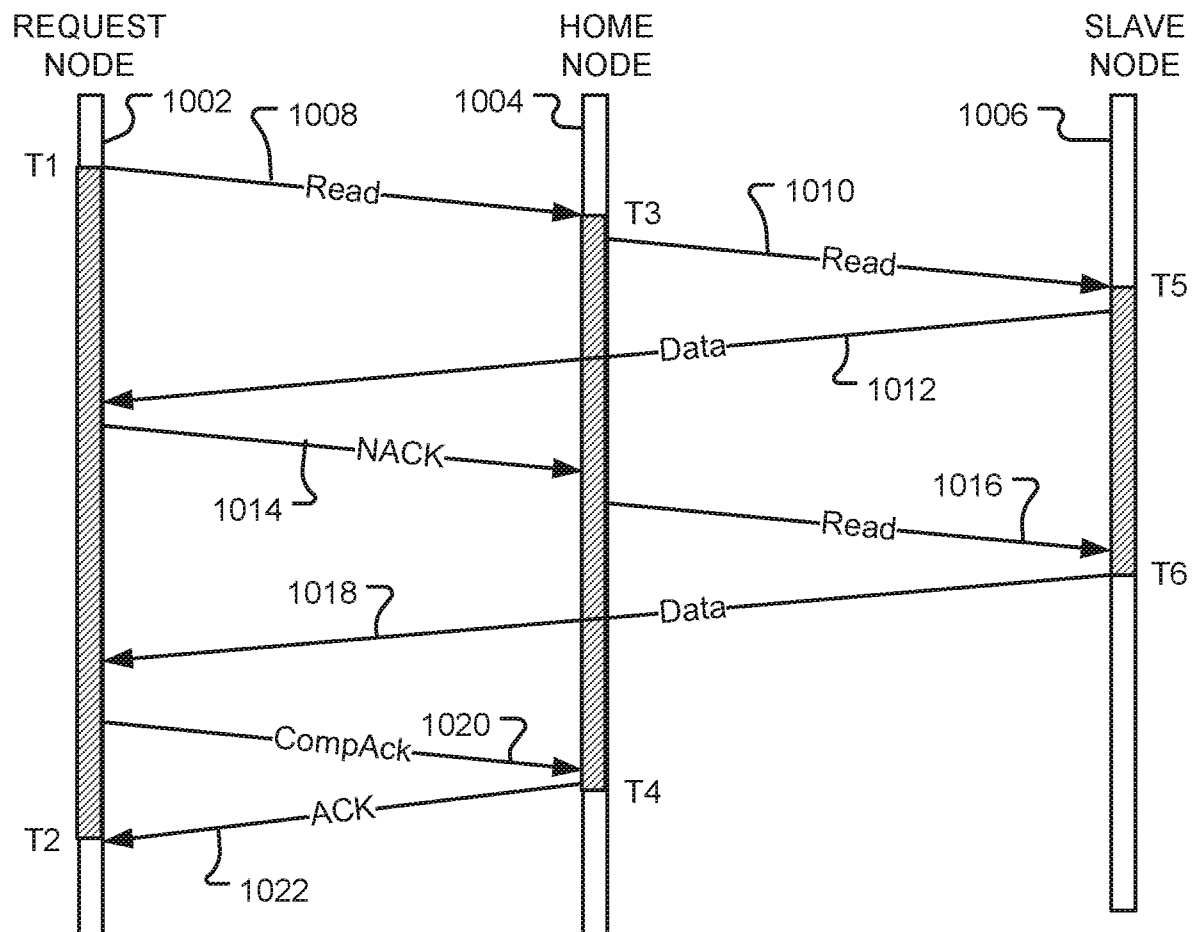
6/10

**FIG. 8**

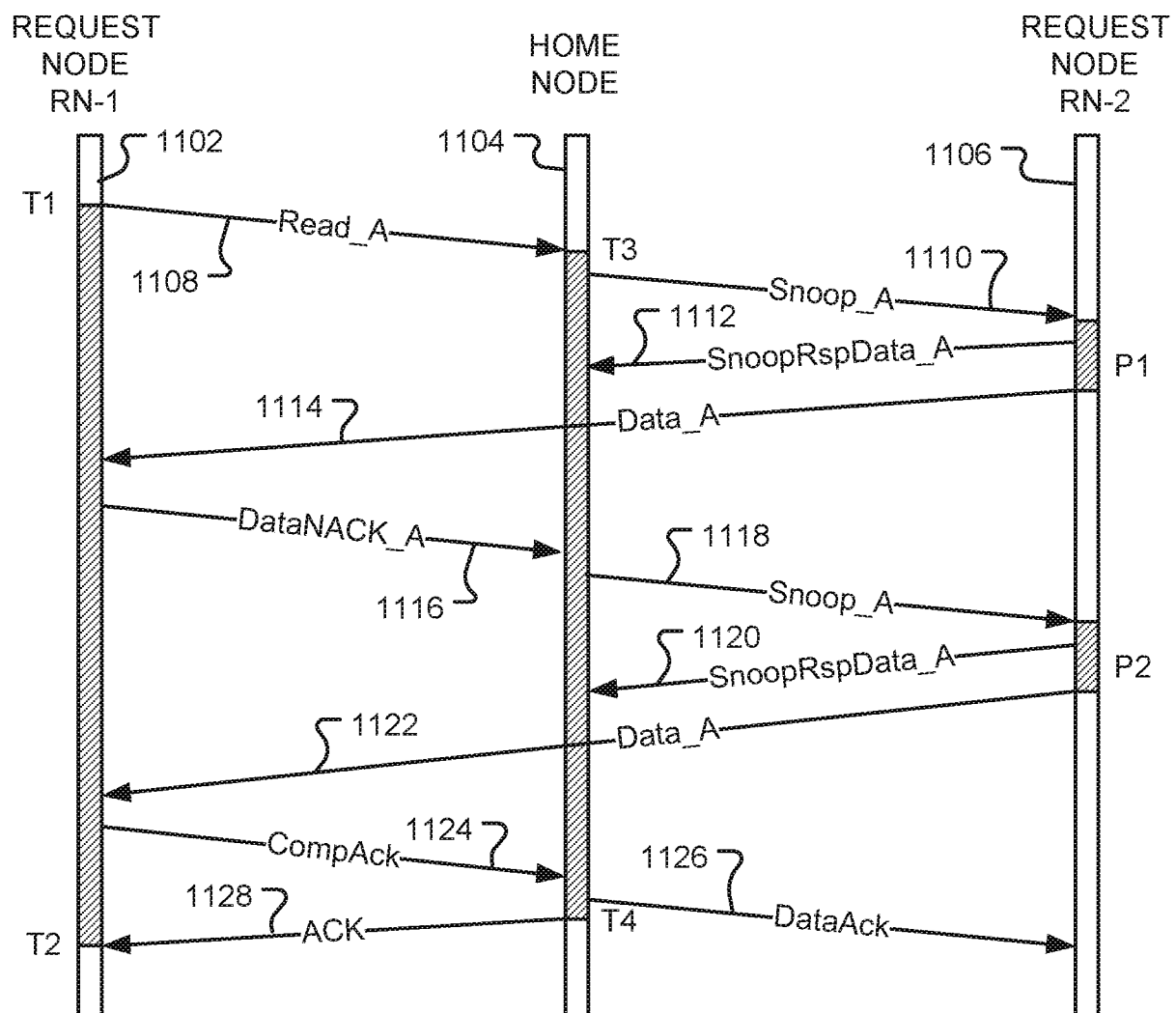
7/10

**FIG. 9**

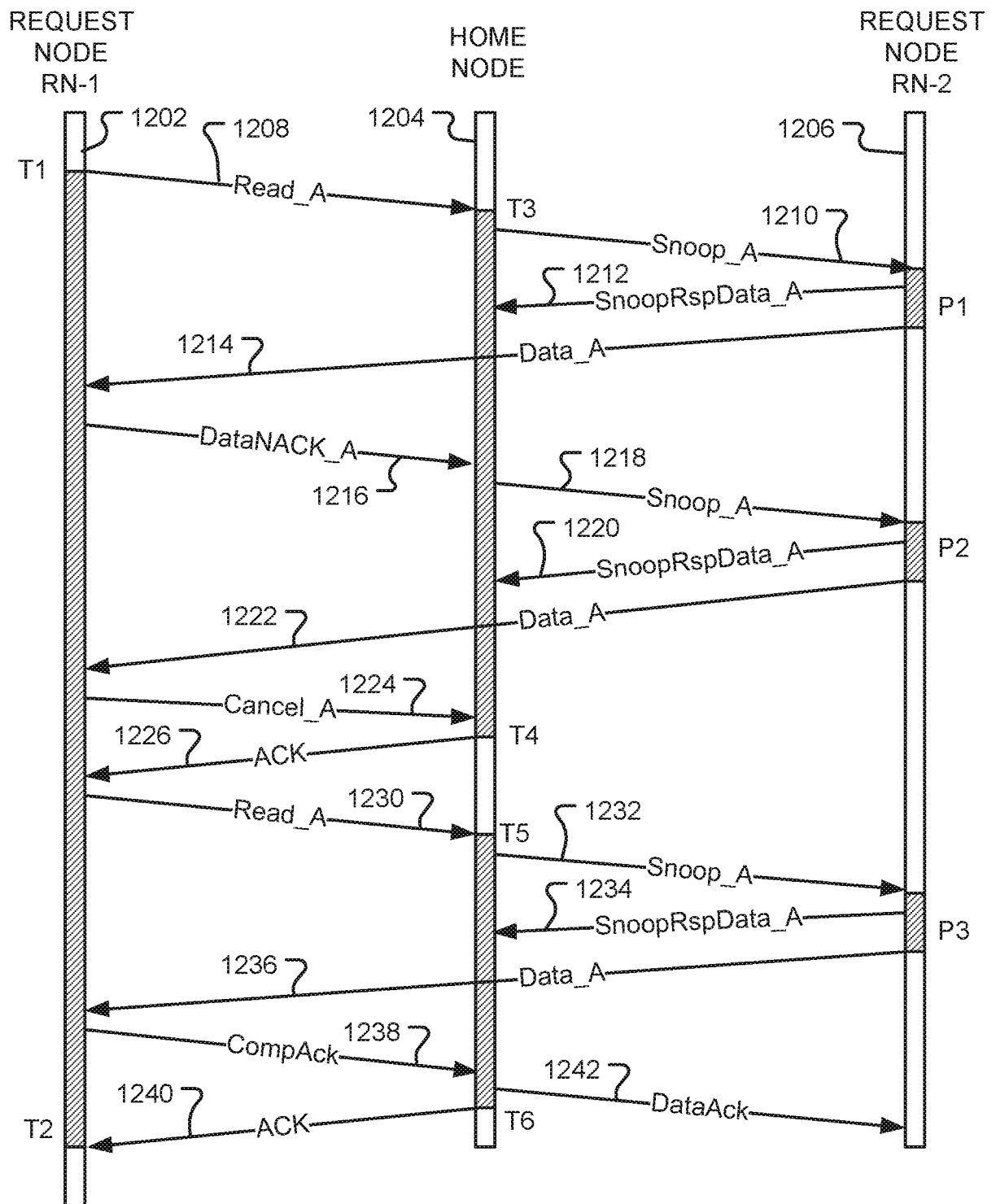
8/10

**FIG. 10**

9/10

**FIG. 11**

10/10

**FIG. 12**

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2019/052400

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L1/16 G06F12/0831 G06F13/36 H04L1/00
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/235138 A1 (CHANG YOUNG-BIN [KR] ET AL) 17 September 2009 (2009-09-17) figure 2 paragraphs [0054] - [0058] -----	1-8
X	US 2013/121296 A1 (JUNG INUK [KR] ET AL) 16 May 2013 (2013-05-16) figure 2A paragraphs [0070] - [0087] -----	1-3,7,8

☐ 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

19 November 2019

Date of mailing of the international search report

31/01/2020

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Marjanovic, Djordje

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2019/052400

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-8

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-8

retransmission of a protocol message via an interconnection circuit

2. claims: 9-17

requesting a data not present in a node from another node

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2019/052400

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009235138 A1	17-09-2009	KR 20090098752 A	17-09-2009
		US 2009235138 A1	17-09-2009

US 2013121296 A1	16-05-2013	KR 20130051473 A	20-05-2013
		US 2013121296 A1	16-05-2013
		WO 2012011786 A2	26-01-2012
