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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 June 2017 (29.06.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/112301 Al

(51) International Patent Classification:
H04L 12/26 (2006.01) *H04L 12/861* (201 3.01)
H04L 12/801 (2013.01)

(21) International Application Number:
PCT/US2016/063592

(22) International Filing Date:
23 November 2016 (23.1 1.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/998,208 26 December 201 5 (26. 12.2015) US

(71) Applicant *(for all designated States except DJ, KW):* **IN¬ TEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **KEPPEL, David;** 101 1 Dale Avenue, Mountain View, California 94040 (US). **DINAN, James;** 75 Reed Road, Hudson, Massachusetts 01749 (US). **ZAK, Robert C ;** 133 Wilder Road, Bolton, Massachusetts 01740 (US).

(74) Agent: **KELLETT, Glen M.;** Barnes & Thornburg LLP, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (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, DJ, 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, 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.

(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 applicant's entitlement to apply for and be granted a patent (Rule 4.1 7(H))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.1 7(in))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: TECHNOLOGIES FOR INLINE NETWORK TRAFFIC PERFORMANCE TRACING

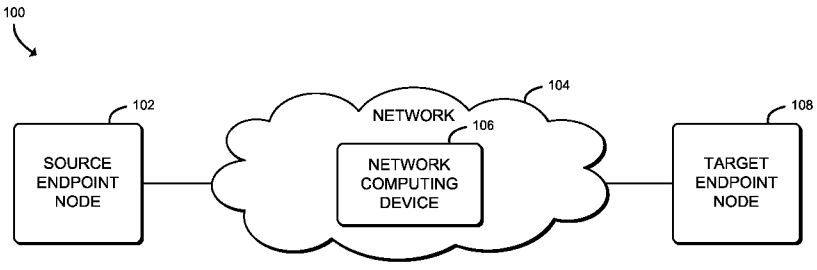

FIG. 1

(57) Abstract: Technologies for tracing network performance in a high performance computing (HPC) network include a network computing device configured to receive a network packet from a source endpoint node and store the header and trace data of the received network packet to a trace buffer of the network computing device. The network computing device is further configured to retrieve updated trace data from the trace buffer and update the trace data portion of the network packet to include the retrieved updated trace data from the trace buffer. Additionally, the network computing device is configured to transmit the updated network packet to a target endpoint node, in which the trace data of the updated network packet is usable by the target endpoint node to determine inline performance of the network relative to a flow of the network packet. Other embodiments are described and claimed herein.

WO 2017/112301 A1

TECHNOLOGIES FOR INLINE NETWORK TRAFFIC PERFORMANCE TRACING

GOVERNMENT RIGHTS CLAUSE

[0001] This invention was made with Government support under contract number H98230-13-D-0124 awarded by the Department of Defense. The Government has certain rights in this invention.

CROSS-REFERENCE TO RELATED APPLICATION

[0002] The present application claims priority to U.S. Utility Patent Application Serial No. 14/998,208, entitled "TECHNOLOGIES FOR INLINE NETWORK TRAFFIC PERFORMANCE TRACING," which was filed on December 26, 2015.

BACKGROUND

[0003] Network operators and communication service providers typically rely on complex, large-scale computing environments, such as high-performance computing (HPC) and cloud computing environments. Due to the complexities associated with such large-scale computing environments, communication performance issues in such HPC systems can be difficult to detect and correct. This problem can be more difficult for performance anomalies (e.g., incast), which can result from the dynamic behavior of an application running on the system or the behavior of the system itself.

[0004] For example, partitioned global address space (PGAS) applications that perform global communication at high message rates can incur network congestion that may be exacerbated by system noise, which may result in performance degradation. Accordingly, HPC systems typically depend on efficient use of the inter-node HPC fabric; however, for many applications, performance is generally limited by HPC fabric performance, as well as by processor, memory, or mass storage performance. Further, due to the complexities associated with HPC system architectures, such HPC fabric performance may be difficult to measure and traffic patterns difficult to understand, making it difficult to identify a root cause of performance problems within the HPC fabric.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The concepts described herein are illustrated by way of example and not by way of limitation in the accompanying figures. For simplicity and clarity of illustration, elements illustrated in the figures are not necessarily drawn to scale. Where considered appropriate,

reference labels have been repeated among the figures to indicate corresponding or analogous elements.

[0006] FIG. 1 is a simplified block diagram of at least one embodiment of a system for inline performance tracing of network traffic through a network computing device of an HPC network that communicatively couples a source endpoint node and a target endpoint node;

[0007] FIG. 2 is a simplified block diagram of at least one embodiment of the source endpoint node of the system of FIG. 1;

[0008] FIG. 3 is a simplified block diagram of at least one embodiment of the network computing device of the system of FIG. 1;

[0009] FIG. 4 is a simplified block diagram of at least one embodiment of the target endpoint node of the system of FIG. 1;

[0010] FIG. 5 is a simplified block diagram of at least one embodiment of an environment that may be established by the source endpoint node of FIG. 2;

[0011] FIG. 6 is a simplified block diagram of at least one embodiment of an environment that may be established by the network computing device of FIG. 3;

[0012] FIG. 7 is a simplified block diagram of at least one embodiment of an environment that may be established by the target endpoint node of FIG. 4;

[0013] FIG. 8 is a simplified block diagram of at least one embodiment of a network packet flow illustrating trace data being forwarded through the HPC network of the system of FIG. 1;

[0014] FIG. 9 is a simplified communication flow diagram of at least one embodiment of a network traffic performance tracing flow that may be executed between the source endpoint node of FIGS. 2 and 5, the network computing device of FIGS. 3 and 6, and the target endpoint node of FIGS. 4 and 7;

[0015] FIG. 10 is a simplified flow diagram of at least one embodiment of a method for inserting trace data into a network packet that may be executed by the network computing device of FIGS. 3 and 6; and

[0016] FIG. 11 is a simplified flow diagram of at least one embodiment of a method for storing trace data of a network packet that may be executed by the target endpoint node of FIGS. 4 and 7.

DETAILED DESCRIPTION OF THE DRAWINGS

[0017] While the concepts of the present disclosure are susceptible to various modifications and alternative forms, specific embodiments thereof have been shown by way of

example in the drawings and will be described herein in detail. It should be understood, however, that there is no intent to limit the concepts of the present disclosure to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives consistent with the present disclosure and the appended claims.

[0018] References in the specification to "one embodiment," "an embodiment," "an illustrative embodiment," etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may or may not necessarily include that particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described. Additionally, it should be appreciated that items included in a list in the form of "at least one of A, B, and C" can mean (A); (B); (C): (A and B); (A and C); (B and C); or (A, B, and C). Similarly, items listed in the form of "at least one of A, B, or C" can mean (A); (B); (C); (A and B); (A and C); (B and C); or (A, B, and C).

[0019] The disclosed embodiments may be implemented, in some cases, in hardware, firmware, software, or any combination thereof. The disclosed embodiments may also be implemented as instructions carried by or stored on one or more transitory or non-transitory machine-readable (e.g., computer-readable) storage media (e.g., memory, data storage, etc.), which may be read and executed by one or more processors. A machine-readable storage medium may be embodied as any storage device, mechanism, or other physical structure for storing or transmitting information in a form readable by a machine (e.g., a volatile or non-volatile memory, a media disc, or other media device).

[0020] In the drawings, some structural or method features may be shown in specific arrangements and/or orderings. However, it should be appreciated that such specific arrangements and/or orderings may not be required. Rather, in some embodiments, such features may be arranged in a different manner and/or order than shown in the illustrative figures. Additionally, the inclusion of a structural or method feature in a particular figure is not meant to imply that such feature is required in all embodiments and, in some embodiments, may not be included or may be combined with other features.

[0021] Referring now to FIG. 1, in an illustrative embodiment, a system 100 for inline performance tracing of network traffic through a network computing device 106 of a high performance computing (HPC) network 104 that communicatively couples a source endpoint

node 102 to a target endpoint node 108. In use, the network computing device 106 performs various operations (e.g., processing services, analysis, forwarding, etc.) on network traffic (i.e., network packets, messages, etc.) received at the network computing device 106. It should be appreciated that the received network traffic may be forwarded (e.g., to other network computing devices (not shown) of the HPC network 104 communicatively coupled to the network computing device 106) or dropped, depending on results of the operations performed thereon.

[0022] As a network packet flows through the system 100 (see, e.g., the network packet flow of FIG. 8), each of the source endpoint node 102, the network computing device 106, and the target endpoint node 108 are configured to record trace data and optionally fill in fields of the network packet with the trace data. To do so, the source endpoint node 102, the network computing device 106, and/or the target endpoint node 108 may each be configured to utilize a tracing engine, described in further detail below, to track or otherwise monitor performance related data (i.e., trace data) of the network traffic passing through the HPC network 104. In some embodiments, the source endpoint node 102, the network computing device 106, and/or the target endpoint node 108 may each be configured to use trace buffers (see, e.g., the trace buffers 810, 812, and 814 of FIG. 8) to temporarily store the trace data while the network packet is being processed. In an illustrative example, when performance tracing of the network traffic through the HPC network 104 is enabled, the source endpoint node 102 is configured to generate a network packet and mark the network packet to indicate that trace data (i.e., network traffic data) is to be collected. After marking the network packet, the source endpoint node 102 transmits the network packet, including the trace data, to the network computing device 106.

[0023] Upon receiving the network packet, the network computing device 106 stores trace data extracted from the network packet, as well as, in some embodiments, header data (i.e., one or more header fields) of the network packet to a local trace buffer (see, e.g., the trace buffer 812 of the illustrative target endpoint node 108 of FIG. 8). Additionally, the network computing device 106 updates the trace data portion of the network packet with updated trace data, which the network computing device 106 then transmits to the target endpoint node 108. Upon receipt of the network packet at the target endpoint node 108, the target endpoint node 108 is configured to retrieve the trace data from the network packet and transmit the trace data to a local trace buffer of the target endpoint node 108 (see, e.g., the trace buffer 814 of the illustrative target endpoint node 108 of FIG. 8), which may then be further processed by the target endpoint node 108 and/or recorded to a trace memory of the target endpoint node 108 (see, e.g., the trace memory 818 of the illustrative target endpoint node 108 of FIG. 8).

[0024] It should be appreciated that while only a single network computing device 106 is shown in the illustrative system 100, the HPC network 104 may include a plurality of network computing devices 106 forming various paths/interconnects through which the network packet may be forwarded through the HPC network 104. Accordingly, it should be further appreciated that, depending on the architecture of the HPC network 104 and a flow of the network packet, the network packet may be transmitted through one or more network computing devices 106 before being forwarded to the target endpoint node 108.

[0025] The source endpoint node 102 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a portable computing device (e.g., smartphone, tablet, laptop, notebook, wearable, etc.) that includes mobile hardware (e.g., processor, memory, storage, wireless communication circuitry, etc.) and software (e.g., an operating system) to support a mobile architecture and portability, a computer, a server (e.g., stand-alone, rack-mounted, blade, etc.), a network appliance (e.g., physical or virtual), a web appliance, a distributed computing system, a processor-based system, and/or a multiprocessor system.

[0026] As shown in FIG. 2, the illustrative source endpoint node 102 includes a processor 202, an input/output (I/O) subsystem 204, a memory 206, a data storage device 208, communication circuitry 210, a tracing engine 216, a clock 218, and one or more peripheral devices 220. Of course, the source endpoint node 102 may include other or additional components, such as those commonly found in a computing device, in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 206, or portions thereof, may be incorporated in the processor 202 in some embodiments. Further, in some embodiments, one or more of the illustrative components may be omitted from the source endpoint node 102.

[0027] The processor 202 may be embodied as any type of processor capable of performing the functions described herein. For example, the processor 202 may be embodied as a single or multi-core processor(s), digital signal processor, microcontroller, or other processor or processing/controlling circuit. Similarly, the memory 206 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 206 may store various data and software used during operation of the source endpoint node 102, such as operating systems, applications, programs, libraries, and drivers.

[0028] The memory 206 is communicatively coupled to the processor 202 via the I/O subsystem 204, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 202, the memory 206, and other components of the source endpoint node 102. For example, the I/O subsystem 204 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, firmware devices, communication links (i.e., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.) and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 204 may form a portion of a system-on-a-chip (SoC) and be incorporated, along with the processor 202, the memory 206, and other components of the source endpoint node 102, on a single integrated circuit chip.

[0029] The data storage device 208 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid-state drives, or other data storage devices. It should be appreciated that the data storage device 208 and/or the memory 206 (e.g., the computer-readable storage media) may store various data as described herein, including operating systems, applications, programs, libraries, drivers, instructions, etc., capable of being executed by a processor (e.g., the processor 202) of the source endpoint node 102.

[0030] The communication circuitry 210 may be embodied as any communication circuit, device, or collection thereof, capable of enabling communications between the source endpoint node 102 and other computing devices (e.g., the network computing device 106, the target endpoint node 108, etc.) over a network (e.g., the HPC network 104). The communication circuitry 210 may be configured to use any one or more communication technologies (e.g., wireless or wired communication technologies) and associated protocols (e.g., Ethernet, Bluetooth®, Wi-Fi®, WiMAX, LTE, 5G, etc.) to effect such communication.

[0031] The illustrative communication circuitry 210 includes a network interface controller (NIC) 212, also commonly referred to as a host fabric interface (HFI) in such HPC networks 104. The NIC 212 may be embodied as one or more add-in-boards, daughtercards, network interface cards, controller chips, chipsets, or other devices that may be used by the source endpoint node 102. For example, in some embodiments, the NIC 212 may be integrated with the processor 202, embodied as an expansion card coupled to the I/O subsystem 204 over an expansion bus (e.g., PCI Express), part of a SoC that includes one or more processors, or included on a multichip package that also contains one or more processors. Additionally or alternatively, in some embodiments, functionality of the NIC 212 may be integrated into one or

more components of the source endpoint node 102 at the board level, socket level, chip level, and/or other levels.

[0032] The illustrative NIC 212 includes a secure memory 214. The secure memory 214 of the NIC 212 may be embodied as any type of memory that is configured to securely store data local to the NIC 212. It should be appreciated that, in some embodiments, the NIC 212 may further include a local processor (not shown) local to the NIC 212. In such embodiments, the local processor of the NIC 212 may be capable of performing functions (e.g., replication, network packet processing, etc.) that may be offloaded to the NIC 212.

[0033] The tracing engine 216 may be embodied as any hardware, firmware, software, or combination thereof (e.g., limited-function high-speed hardware) capable of being configured by a processor (e.g., the processor 302) of the network computing device 106 and performing the various functions described herein, such as collecting performance-relevant data and making the performance-relevant data available for performance analysis to determine network performance characteristics. To do so, the tracing engine 216 is configured to interact with hardware blocks that generate the performance data (e.g., via performance counter hardware) related to network traffic and otherwise process the network packet. In some embodiments, the tracing engine 216 may form a portion of the processor 202 or otherwise may be established by a processor of the source endpoint node 102 (e.g., a processor (not shown) local to the NIC 212). In other embodiments, the tracing engine 216 may be embodied as an independent circuit or processor (e.g., a specialized co-processor or application specific integrated circuit (ASIC)). The tracing engine 216 is configured to manage trace data at the source endpoint node 102.

[0034] The trace data may include any type of information related to network traffic travelling through the HPC network 104, such as a timestamp recording a time of interest (e.g., a time of ingress, a time at which the network packet was queued, a time of egress, etc.), routing information (e.g., a source identifier, a destination identifier, etc.), delay information (e.g., an amount of time between queuing and egress, a number of cycles between ingress and egress, etc.), decision-making information (e.g., why the network packet was forwarded to a particular network computing device 106), information regarding one or more characteristics (e.g., a temperature, an internal buffer usage, a processor usage percentage, a memory usage, etc.) of one or more components of the device that generated, processed, forwarded, and/or received the network packet, etc.

[0035] To manage the trace data, the tracing engine 216 is configured to collect trace data related to a network packet generated at the source endpoint node 102 and insert the trace

data into a network packet To do so, in some embodiments, the tracing engine 216 may store trace data in a local trace buffer of the source endpoint node 102 (see, e.g., the trace buffer 810 of the illustrative source endpoint node 102 of FIG. 8) and retrieve at least a portion of the trace data from the local trace buffer for insertion into the network packet. Additionally or alternatively, in some embodiments, the tracing engine 216 may generate trace data (e.g., an egress time) and directly insert such trace data into the network packet. In other words, some trace data may be generated on-demand and not stored in the local trace buffer. As such, it should be appreciated that the trace data management may be predicated or otherwise influenced by a state of the communication circuitry or progress of the communication circuitry in processing/forwarding the network packet.

[0036] In some embodiments, the tracing engine 216 may be configured to interpret a setting of the source endpoint node 102 that indicates whether to perform tracing of network traffic (i.e., whether to track trace data of network packets through the HPC network 104). In such embodiments, the tracing engine 216 may be further configured to update one or more bits of a header of network packets being transmitted from the source endpoint node 102 that indicate whether performance tracing of network traffic is enabled. It should be appreciated that additional and/or alternative information may be represented in the one or more bits of the header, such as a type of trace data to be collected, a size of the trace data to be collected (e.g., whether the trace data should be compressed), etc. It should be further appreciated that the information may instead be implemented as readable fields in the trace data.

[0037] In some embodiments, additional and/or alternative procedures may be implemented by the tracing engine 216 to enable network traffic performance tracing. For example, the additional and/or alternative procedures may include booting one or more of the source endpoint node 102, the network computing device 106, and the target endpoint node 108 into a tracing mode, phasing operation to adjust tracing at phase boundaries, and/or increasing or extending available header types that are not used for typical network packet transmission.

[0038] The clock 218 may be embodied as any software, hardware component(s), and/or circuitry from which a timestamp can be generated therefrom and is otherwise capable of performing the functions described herein. For example, in the illustrative embodiment, the clock 218 may be implemented via an on-chip oscillator. In some embodiments, the clock 218 may be shared (e.g., multiple distributed clocks being generally synchronized using a synchronization protocol).

[0039] The peripheral devices 220 may include any number of input/output devices, interface devices, and/or other peripheral devices. For example, in some embodiments, the

peripheral devices 220 may include a display, a touch screen, graphics circuitry, a keyboard, a mouse, a microphone, a speaker, and/or other input/output devices, interface devices, and/or peripheral devices. The particular devices included in the peripheral devices 220 may depend on, for example, the type and/or intended use of the source endpoint node 102. The peripheral devices 220 may additionally or alternatively include one or more ports, such as a USB port, for example, for connecting external peripheral devices to the source endpoint node 102.

[0040] Referring again to FIG. 1, the HPC network 104, also commonly referred to as an HPC fabric, may be embodied as any type of HPC architecture capable of aggregating computing power in a manner such that advanced application programs can be run efficiently, reliably, and quickly. Unlike local area networks (LANs) and a wide area networks (WANs), which tend to have larger messages and packets, burstier operation (significant idle periods and significantly-frequent idle periods), and relatively long distances (where the speed of light limits some time constants which in turn constrain the design of long-haul networks), the HPC network 104 is configured to sustain very high rates of small-message traffic over short distances.

[0041] It should be appreciated that the source endpoint node 102 and/or the target endpoint node 108 may be connected to the HPC network 104 via another network, such as a wireless local area network (WLAN), a wireless personal area network (WPAN), a cellular network (e.g., Global System for Mobile Communications (GSM), Long-Term Evolution (LTE), etc.), a telephony network, a digital subscriber line (DSL) network, a cable network, a local area network (LAN), a wide area network (WAN), a global network (e.g., the Internet), or any combination thereof. In other words, the HPC network 104 may serve as a centralized network and, in some embodiments, may be communicatively coupled to another network (e.g., the Internet). Accordingly, the other network may include a variety of other network computing devices (e.g., virtual and physical routers, switches, network hubs, servers, storage devices, compute devices, etc.), as needed to facilitate communication between the source endpoint node 102 and the HPC network 104 and/or the target endpoint node 108 and the HPC network 104, which are not shown to preserve clarity of the description.

[0042] The network computing device 106 may be embodied as any type of network traffic processing and/or forwarding device capable of performing the functions described herein, such as, without limitation, a switch (e.g., rack-mounted, standalone, fully managed, partially managed, full-duplex, and/or half-duplex communication mode enabled, etc.), a server (e.g., stand-alone, rack-mounted, blade, etc.), a network appliance (e.g., physical or virtual), a router, a web appliance, a distributed computing system, a processor-based system, and/or a

multiprocessor system. It should be appreciated, due to the demands for high speed of the HPC network 104 for which the network computing device 106 provides interconnects, that the network computing device 106 may have limited function, but include high-speed hardware (see, e.g., the tracing engine 316 described below) with which to parse header data and insert trace data.

[0043] The tracing engine 316 may be embodied as any hardware, firmware, software, or combination thereof (e.g., limited-function high-speed hardware) capable of being configured by a processor (e.g., the processor 302) of the network computing device 106 and performing the various functions described herein, such as collecting performance-relevant data and making the performance-relevant data available for performance analysis to determine network performance characteristics. The network performance characteristics may include any performance-related data that identifies a performance characteristic of the HPC network 104, as may be determined from the trace data at the target endpoint node 108, such as how full network computing device 102 queues are, the paths each traced network packet takes through the HPC network 104, how much time each network packet takes to travel to/through each network computing device 102, etc.

[0044] To collect the performance-relevant data, the tracing engine 316 is configured to interact with hardware blocks that generate the performance data (e.g., via performance counter hardware) related to network traffic and otherwise process the network packet. In some embodiments, the tracing engine 316 may form a portion of the processor 302 or otherwise may be established by a processor of the network computing device 102 (e.g., a processor (not shown) local to the NIC 312). In other embodiments, the tracing engine 316 may be embodied as an independent circuit or processor (e.g., a specialized co-processor or ASIC).

[0045] As shown in FIG. 3, similar to the illustrative source endpoint node 102 of FIG. 2, the illustrative network computing device 106 includes a processor 302, an I/O subsystem 304, a memory 306, a data storage device 308, communication circuitry 310 that includes a NIC 312 and a secure memory 314 of the NIC 312, a tracing engine 316, and a clock 318. As such, further descriptions of the like components are not repeated herein with the understanding that the description of the corresponding components provided above in regard to the illustrative source endpoint node 102 of FIG. 2 applies equally to the corresponding components of the illustrative network computing device 106 of FIG. 3.

[0046] Unlike existing software-only driven per-network packet tracing technologies, which do not typically provide visibility into the in-flight behavior of the network traffic through the HPC network 104, the tracing engine 316 is configured to trace individual network

packets through the HPC network 104 at low time and space overhead. The tracing engine 316 of the network computing device 106, similar to the tracing engine 216 of the source endpoint node 102 described previously, is configured to insert trace data into a network packet to be forwarded from the network computing device 106. To do so, the tracing engine 316 is configured to generate trace data that may be stored local to the network computing device 106 for later retrieval and insertion or directly inserted into the trace data portion of the network packet.

[0047] The tracing engine 316 is further configured to extract trace data from a network packet received by the network computing device 106, such as may be received from the source endpoint node 102 or another network computing device 106. Additionally, the tracing engine 316 is configured to manage the local trace buffer of the network computing device 106 (see, e.g., the trace buffer 812 of the illustrative network computing device 106 of FIG. 8).

[0048] The tracing engine 316 is further configured to compress the trace data (i.e., reduce the size of the trace data), in some embodiments. For example, the tracing engine 316 may be configured to save routing information as an egress or ingress port/lane number (e.g., one of thirty-two) rather than a number identifying the network computing device 106. Accordingly, for any given path, the number identifying the network computing device 106 can be reconstructed from a sequence of port/lane numbers. Alternatively, the tracing engine 316 may be configured to save one bit indicating whether minimal or non-minimal routing is set, such as by using a Boolean value. In such embodiments, for a given source endpoint node 102 and a target endpoint node 108, the minimal route may be statically predictable.

[0049] Additionally or alternatively, the tracing engine 316 may be configured to save ingress-to-egress time (i.e., a time delta) rather than absolute time. In some embodiments, the tracing engine 316 may be configured to save time in ranges, or bins. For example, a range, or bin, may represent a number of consecutive cycle delays. In such embodiments, eight ranges may be encoded with 3 bits and can encode an arbitrary delay, albeit with reduced resolution, for example. Alternatively, the tracing engine 316 may be configured to save only data deemed to fall into a certain predetermined category. For example, the tracing engine 316 may be configured to support short and long trace entries (e.g., increment a counter by zero or one, increment the counter by one or two, etc.). In another example, network packets handled in a time less than a predetermined threshold duration may save trace data in short-form, while network packets handled in a time greater than a predetermined threshold duration (i.e., delayed network packets) may save trace data in long-form. In such embodiments, a counter may be saved or a change in one counter may be used to trigger saving of another counter.

[0050] Additionally or alternatively, trace data may be collected variably. For example, if one resource of the network computing device 106 could be a bottleneck or another resource of the network computing device 106 could be a bottleneck, then the tracing engine 316 may examine counters for each of the resources and collect trace data on only one of the resources, as well as a bit to indicate which resource was recorded. It should be appreciated that the network computing device 106 may be configured to collect additional other data in transit, such as various metrics of the network computing device 106, which may be stored by the tracing engine, inserted with the trace data in a network packet, and may be compressed in a similar fashion as the trace data described previously.

[0051] Similar to the source endpoint node 102, the target endpoint node 108 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a portable computing device (e.g., smartphone, tablet, laptop, notebook, wearable, etc.) that includes mobile hardware (e.g., processor, memory, storage, wireless communication circuitry, etc.) and software (e.g., an operating system) to support a mobile architecture and portability, a computer, a server (e.g., stand-alone, rack-mounted, blade, etc.), a network appliance (e.g., physical or virtual), a web appliance, a distributed computing system, a processor-based system, and/or a multiprocessor system.

[0052] As shown in FIG. 4, the illustrative target endpoint node 108, similar to the illustrative source endpoint node 102 of FIG. 2, includes a processor 402, an input/output (I/O) subsystem 404, a memory 406, a data storage device 408, communication circuitry 410 that includes a NIC 412 and a secure memory 414 of the NIC 412, a tracing engine 416, a clock 418, and one or more peripheral devices 420. As such, further descriptions of the like components are not repeated herein with the understanding that the description of the corresponding components provided above in regard to the illustrative source endpoint node 102 of FIG. 2 applies equally to the corresponding components of the illustrative network computing device 106 of FIG. 3.

[0053] The tracing engine 416 of the target endpoint node 108, similar to the tracing engine 216 of the source endpoint node 102 previously described, is configured to retrieve locally stored trace data. The tracing engine 416 is further configured to extract trace data from a network packet received by the target endpoint node 108, such as may be received from the network computing device 106, and store the trace data locally. In some embodiments, the tracing engine 416 may be additionally configured to process the stored trace data and record it in a trace memory of the target endpoint node 108. Additionally or alternatively, in some embodiments, the tracing engine 416 may form a portion of the processor 402 or otherwise may

be established by a processor of the target endpoint node 108 (e.g., a processor (not shown) local to the NIC 412). In other embodiments, the tracing engine 416 may be embodied as an independent circuit or processor (e.g., a specialized co-processor or application specific integrated circuit (ASIC)).

[0054] Referring now to FIG. 5, in an illustrative embodiment, the source endpoint node 102 establishes an environment 500 during operation. The illustrative environment 500 includes a network communication management module 510 and a performance tracing module 520. Each of the modules, logic, and other components of the environment 500 may be embodied as hardware, software, firmware, or a combination thereof. For example, each of the modules, logic, and other components of the environment 500 may form a portion of, or otherwise be established by, the processor 202, the communication circuitry 210 (e.g., the NIC 212), and/or other hardware components of the source endpoint node 102. As such, in some embodiments, one or more of the modules of the environment 500 may be embodied as circuitry or a collection of electrical devices (e.g., network communication management circuitry 510, performance tracing circuitry 520, etc.). It should be appreciated that the source endpoint node 102 may include other components, sub-components, modules, sub-modules, and/or devices commonly found in a computing device, which are not illustrated in FIG. 5 for clarity of the description.

[0055] The illustrative environment 500 of the source endpoint node 102 additionally includes trace data 502, which may be stored in the memory 214 and/or the data storage 216 of the source endpoint node 102, and may be accessed by the various modules and/or sub-modules of the source endpoint node 102. As described previously, the trace data 502 may include any type of information related to the flow of network traffic, such as route information, delay information, queue information, decision-making information, etc., as well as any characteristics of one or more components of the source endpoint node 102 (e.g., temperature, internal buffer usage, processor usage, memory usage, etc.) that processed/forwarded the network packet.

[0056] The network communication management module 510 is configured to facilitate inbound and outbound network communications (e.g., network traffic, network packets, network flows, etc.) to and from the source endpoint node 102. To do so, the network communication management module 510 is configured to receive and process network packets from other computing devices (e.g., the network computing device 106). Additionally, the network communication management module 510 is configured to prepare and transmit network packets to another computing device (e.g., the network computing device 106).

Accordingly, in some embodiments, at least a portion of the functionality of the network communication management module 510 may be performed by the communication circuitry 210, and more specifically by the NIC 212.

[0057] The performance tracing module 520 is configured to manage trace data stored local to the source endpoint node 102 and/or generated by the source endpoint node 102, as well as insert such trace data into a network packet generated by the source endpoint node 102. To do so, in some embodiments, the performance tracing module 520 may include a trace data collection module 522 configured to generate and/or retrieve trace data for insertion into the network packet. In some embodiments, the trace data collection module 522 configured to retrieve the trace data from the trace data 502, such as may be stored in a local trace buffer (see, e.g., the trace buffer 810 of FIG. 8 local to the NIC 212). Additionally or alternatively, in some embodiments, the performance tracing module 520 may include a trace data insertion module 524 configured to insert the retrieved trace data into the network packet, such as a portion of the network packet assigned to store the trace data. It should be appreciated that, in some embodiments, at least a portion of the functions of the performance tracing module 520 described herein may be performed by the tracing engine 216 described above.

[0058] Referring now to FIG. 6, in an illustrative embodiment, the network computing device 106 establishes an environment 600 during operation. The illustrative environment 600 includes a network communication management module 610 and a performance tracing module 620. Each of the modules, logic, and other components of the environment 600 may be embodied as hardware, software, firmware, or a combination thereof. For example, each of the modules, logic, and other components of the environment 600 may form a portion of, or otherwise be established by, the processor 302, the communication circuitry 310 (e.g., the NIC 312), and/or other hardware components of the network computing device 106. As such, in some embodiments, one or more of the modules of the environment 600 may be embodied as circuitry or a collection of electrical devices (e.g., network communication management circuitry 610, performance tracing circuitry 620, etc.). It should be appreciated that the network computing device 106 may include other components, sub-components, modules, sub-modules, and/or devices commonly found in a computing device, which are not illustrated in FIG. 6 for clarity of the description.

[0059] The illustrative environment 600 of the network computing device 106 additionally includes trace data 602 and header data 604, each of which may be stored in the memory 314 and/or the data storage 316, and may be accessed by the various modules and/or sub-modules of the network computing device 106. As described previously, the trace data 602

may include any type of information related to the flow of network traffic, such as route information, delay information, queue information, decision-making information, etc., as well as any characteristics of one or more components of the source endpoint node 102 and/or the network computing device(s) 106 (e.g., temperature, internal buffer usage, processor usage, memory usage, etc.) that processed/forwarded the network packet.

[0060] The network communication management module 610, similar to the network communication management module 510 of FIG. 5, is configured to facilitate inbound and outbound network communications (e.g., network traffic, network packets, network flows, etc.) to and from the network computing device 106. To do so, the network communication management module 610 is configured to receive and process network packets from other computing devices (e.g., the source endpoint node 102, the target endpoint node 108, another network computing device 106, etc.). Additionally, the network communication management module 610 is configured to prepare and transmit network packets to another computing device (e.g., the source endpoint node 102, the target endpoint node 108, another network computing device 106, etc.). Accordingly, in some embodiments, at least a portion of the functionality of the network communication management module 610 may be performed by the communication circuitry 310, and more specifically by the NIC 312.

[0061] The performance tracing module 620 is configured to manage trace data stored local to the network computing device 106 and/or generated by the network computing device 106, as well as insert such trace data into a network packet generated by the source endpoint node 102. To do so, in some embodiments, the performance tracing module 620 may include a trace data extraction module 622 configured to extract trace data from a network packet (e.g., from a designated portion of the network packet) and a trace buffer management module 624 configured to store the extracted trace data (e.g., the trace data 602) at a local trace buffer (see, e.g., the trace buffer 812 of FIG. 8 local to the NIC 312). Additionally, in some embodiments, the performance tracing module 620 may include a trace data collection module 626 configured to generate and/or retrieve trace data for insertion into the network packet. In some embodiments, the trace data collection module 626 may be configured to retrieve trace data from the trace data 602. Additionally or alternatively, in some embodiments, the performance tracing module 620 may include a trace data insertion module 628 configured to insert the retrieved trace data into the network packet, such as in a portion of the network packet designated for storing the trace data (see, e.g., the trace data portion 804 of FIG. 8). It should be appreciated that, in some embodiments, at least a portion of the functions of the performance

tracing module 620 described herein may be performed by the tracing engine 316 of the network computing device 106.

[0062] Referring now to FIG. 7, in an illustrative embodiment, the target endpoint node 108 establishes an environment 700 during operation. The illustrative environment 700 includes a network communication management module 710, a performance tracing module 720, and a payload management module 730. Each of the modules, logic, and other components of the environment 700 may be embodied as hardware, software, firmware, or a combination thereof. For example, each of the modules, logic, and other components of the environment 700 may form a portion of, or otherwise be established by, the processor 402, the communication circuitry 410 (e.g., the NIC 412), and/or other hardware components of the target endpoint node 108. As such, in some embodiments, one or more of the modules of the environment 700 may be embodied as circuitry or a collection of electrical devices (e.g., network communication management circuitry 710, performance tracing circuitry 720, payload management circuitry 730, etc.). It should be appreciated that the target endpoint node 108 may include other components, sub-components, modules, sub-modules, and/or devices commonly found in a computing device, which are not illustrated in FIG. 7 for clarity of the description.

[0063] The illustrative environment 700 of the target endpoint node 108 additionally includes trace data 702 and payload data 704, each of which may be accessed by the various modules and/or sub-modules of the target endpoint node 108. Further, each of the trace data 702 and payload data 704 may be stored in the memory 414 and/or the data storage 416. As described previously, the trace data 702 may include any type of information related to the flow of network traffic, such as route information, delay information, queue information, decision-making information, etc., as well as any characteristics of one or more components of the target endpoint node 108 (e.g., temperature, internal buffer usage, processor usage, memory usage, etc.) of the source endpoint node 102, the network computing device(s) 106, and/or target endpoint node 108 that processed/forwarded the network packet.

[0064] The network communication management module 710, similar to the network communication management modules 510 and 610 of FIGS. 5 and 6, respectively, is configured to facilitate inbound and outbound network communications (e.g., network traffic, network packets, network flows, etc.) to and from the target endpoint node 108. To do so, the network communication management module 710 is configured to receive and process network packets from other computing devices (e.g., the network computing device 106). Additionally, the network communication management module 710 is configured to prepare and transmit network packets to another computing device (e.g., the network computing device 106).

Accordingly, in some embodiments, at least a portion of the functionality of the network communication management module 710 may be performed by the communication circuitry 410, and more specifically by the NIC 412.

[0065] The performance tracing module 720 is configured to manage trace data stored local to the target endpoint node 108 and/or generated by the target endpoint node 108. To do so, in some embodiments, the performance tracing module 620 may include a trace data extraction module 722 configured to extract trace data from a network packet (e.g., from a trace data portion of the network packet), a trace buffer management module 724 configured to store the extracted trace data at a local trace buffer (see, e.g., the trace buffer 814 of FIG. 8 local to the NIC 412), a trace data collection module 726 configured to generate and/or retrieve trace data for insertion into the network packet, and a trace data storage management module 728 configured to store trace data from the local trace buffer into a memory location external to the NIC 412. It should be appreciated that, in some embodiments, at least a portion of the functions of the performance tracing module 720 described herein may be performed by the tracing engine 416 of the target endpoint node 108.

[0066] The payload management module 730 is configured to manage payloads of the network packets received by the target endpoint node 108. To do so, in some embodiments, the payload management module 730 may include a payload extraction module 732 to extract the payloads from the received network packets and a payload data storage management module 734 to manage the storage of the extracted payloads to a memory local to the target endpoint node 108 (see, e.g., the application memory 816 of the illustrative target endpoint node 108 of FIG. 8).

[0067] Referring now to FIG. 8, in use, the source endpoint node 102 is configured to generate a network packet 800 that includes a header portion 802, a trace data portion 804, and a payload portion 806. The source endpoint node 102 is additionally configured to generate trace data and/or retrieve trace data from a trace buffer 810 in the secure memory 214 of the NIC 212 and insert the generated/retrieved trace data into the trace data portion 804. The trace data in the trace data portion 804 of the network packet 800 inserted by the source endpoint node 102 may include one or more times of interest (e.g., when the network packet 800 was queued by software of the source endpoint node 102, a time of egress from the source endpoint node 102, etc.), counters (e.g., a hop counter incremented at every network computing device 106 the network packet 800 is forwarded to), an indicator to indicate where to write the trace data in the network packet 800 (e.g., a trace data location indicator to indicate a location in the trace data portion 804 at which the network computing device 106 is to write its trace data), and

an indicator to indicate a maximum size of the field/location in which the network computing device 106 is to insert its trace data into the trace data portion 804 (e.g., a trace data size indicator).

[0068] In some embodiments, the source endpoint node 102 may be further configured to insert an indicator into the network packet 800 that indicates trace data is included in the network packet 800 and/or that trace data is to be tracked throughout the flow of the network packet 800 through the HPC network 104 (i.e., at each target of the network packet 800 through the HPC network 104). In such embodiments, the indicator may be included in the header portion 802 of the network packet 800.

[0069] It should be appreciated that, in some embodiments, no trace data may be collected at one or more of the source endpoint node 102, the network computing device 106, and the target endpoint node, such as when performance tracing is not enabled for a particular network packet. In such embodiments, the trace data portion 804 of the network packet 800 may be empty, excluded, or otherwise left unchanged.

[0070] It should be further appreciated that, in some embodiments, the header portion 802 and/or payload portion 806 of the network packet 800 may include redundant information usable for error detection (e.g., cyclic redundancy check (CRC), checksum, etc.). In such embodiments, the redundant information is typically not modified while the network packet 800 is in-flight, such that the network packet 800 does not need to be recomputed, which can introduce overhead, latency, etc. Accordingly, in some embodiments, the trace data portion 804 may be excluded from the error detection calculations, such that adding/modifying trace data in the trace data portion 804 does not change the error detection calculations for the header portion 802 and/or payload portion 806 of the network packet 800. However, in some embodiments, the trace data portion 804 may be separately protected by a separate error detection technology may be used, such as using a weaker form of CRC that may provide less error protection but reduced overhead or protecting individual fields within the trace data portion 804 (e.g., using the weaker form of CRC).

[0071] The network computing device 106 is configured to receive the network packet 800 from the source endpoint node 102 or another network computing device 106, generate any applicable trace data, and insert the generated trace data into the trace data portion 804 of the network packet 800. It should be appreciated that the network computing device 106 may be configured to determine whether the network packet 800 includes a trace data portion 804 prior to trace data generated, such as may be indicated via a field in the header portion 802 of the network packet 800, prior to attempting to extract the trace data. Additionally, the network

computing device 106 is configured to extract the trace data from the trace data portion 804 (e.g., based on the write location and/or max size indicators), and store the extracted trace data to a trace buffer 812 in the secure memory 314 local to the NIC 312. Accordingly, in such embodiments, the network computing device 106 is further configured to retrieve trace data from the trace buffer 812 and insert the retrieved trace data into the trace data portion 804 of the network packet 800.

[0072] The trace data inserted by the network computing device 106 may include any type of information related to the network packet 800 travelling through the HPC network 104, such as transit information, queue information, credit availability, error rate counters, other event counters (e.g., low bits of the event counters), routing decision information (e.g., a selected routing path), information relating to a detected cause of delay (e.g., delayed due to credit exhaustion, port availability bottleneck, etc.), and/or characteristic(s) of components of the network computing device 106 (e.g., a temperature, an internal buffer usage information, a processor usage percentage, a memory usage percentage, etc.). The trace data may additionally include a type indicator indicating what type of trace data and/or what format to store the trace data.

[0073] Additionally or alternatively, the trace data may include an identifying indicator, such as may be managed by one or more counters that are incremented at every network computing device 106 the network packet is forwarded to (e.g., a hop counter). In other words, the trace data may be paired with the counter value at a particular network computing device 106 and stored in the trace data portion, such that the target endpoint node 108 can determine a path and associate the trace data with the network computing devices 106 of the path. In such embodiments, the counter value may be paired with a time of interest (e.g., a time of ingress of the network packet 800 at the network computing device 106, a time for which the network packet 800 was queued at the network computing device 106, a time of egress of the network packet 800 from the network computing device 106, etc.) or other information related to the network packet 800 travelling through the HPC network 104 as described previously. The network computing device 106 is further configured to forward the network packet 800 to either another network computing device 106 or the target endpoint node 108, depending on a determined flow of the network packet 800.

[0074] The target endpoint node 108 is configured to receive the network packet 800 from the network computing device 106, extract the trace data from the trace data portion 804 (e.g., based on the write location and/or max size indicators), and store the extracted trace data to a trace buffer 814 in the secure memory 414 local to the NIC 412. Additionally, the target

endpoint node 108 is configured to store the trace data from the trace buffer 814 to a trace memory 818 (i.e., a stable storage location) external to the NIC 412. It should be appreciated that storing the trace data to the trace memory 818 frees at least a portion of the trace memory 818, which may allow for additional trace data to be saved. The target endpoint node 108 is further configured to extract a payload from the payload portion 806 of the network packet and store the extracted payload into an application memory 816 external to the NIC 412.

[0075] Referring now to FIG. 9, an embodiment of a communication flow 900 for network traffic performance tracing includes the source endpoint node 102, the network computing device 106, and the target endpoint node 108. The illustrative communication flow 900 includes a number of data flows, some of which may be executed separately or together, depending on the embodiment. It should be appreciated that at least a portion of the data flows described herein may be performed by the tracing engine of the respective computing device (i.e., the tracing engine 216 of the source endpoint node 102, the tracing engine 316 of the network computing device 106, and the tracing engine 416 of the target endpoint node 108). In data flow 902, the source endpoint node 102 collects trace data. To do so, the source endpoint node 102 may generate the trace data (i.e., to be directly inserted, not stored) and/or retrieve previously generated trace data from a trace buffer local to the NIC 212. In data flow 904, the source endpoint node 102 inserts the retrieved trace data into a trace portion of a network packet generated by the source endpoint node 102. In data flow 906, the source endpoint node 102 transmits the network packet, including the trace data, to the network computing device 106.

[0076] In data flow 908, the network computing device 106 extracts header and trace data from the network packet transmitted to the network computing device 106 in data flow 906. In data flow 910, the network computing device 106 stores the extracted header and trace data into a trace buffer local to the network computing device 106. It should be appreciated that source and destination computing devices are typically encoded into a field of the header of the network packet. As such, rather than encoding such information in the trace data, such information may be extracted from the appropriate header fields and saved alongside the trace data. As a result, additional pertinent data related to the trace data may be saved without increasing the size of the network packet (i.e., without bloating the trace data portion of the network packet). It should be further appreciated that, in some embodiments, the trace data is not extracted from the network packet. As such, in some embodiments, at least a portion of the data flows 908 and 910 may be skipped.

[0077] In data flow 912, the network computing device 106 collects trace data. To do so, as described previously, the network computing device 106 may generate trace data and/or retrieve trace data from the trace buffer. In data flow 914, the network computing device 106 inserts the collected trace data into a trace data portion of the network packet. It should be appreciated that the network computing device 106 may be configured to perform an analysis on the trace data in the trace buffer (e.g., to condense or otherwise compress the trace data) prior to being inserted into the trace data portion of the network packet. Additionally or alternatively, in network architectures designed for high throughput of messages (e.g., HPC fabrics), messages may use cut-through implementations in which a network packet ingress into the network computing device 106 and egress from the network computing device 106 occur near simultaneously (i.e., only a few cycle delay in the network computing device 106). In such implementations, packet inspection and decision processing may be required prior to processing data of the network packet (i.e., serialization), which can introduce overhead, latency, etc.

[0078] To minimize serialization, the network computing device 106 may be configured to reduce or altogether remove overhead associated with serialization, such as by allocating space (i.e., a trace data portion) in the network packet prior to transmission of the network packet to the network computing device 106 from the source endpoint node 102, including a position indicator in a header field of the network packet that identifies where the trace data portion is written to or is to be written to in the network packet, incrementing the position indicator unconditionally into the egressing network packet to minimize delay, and/or including a maximum size indicator in a header field of the network packet that controls a size of the trace data and/or whether any trace data is written into the trace data portion of the network packet.

[0079] In data flow 916, the network computing device 106 transmits the network packet to the target endpoint node 108. It should be appreciated that, in some embodiments, the network computing device 106 may forward the network packet through one or more other network computing devices 106 prior to the network packet being forwarded to the target endpoint node 108. In data flow 918, the target endpoint node 108 extracts the payload and trace data from the network packet that was received from the network computing device 106 in data flow 916. In data flow 920, the target endpoint node 108 stores the extracted trace data into a trace buffer of the target endpoint node 108. It should be appreciated that additional trace data may be generated by the target endpoint node 108 (i.t, not extracted from the trace data portion of the network packet), such as an ingress time. Accordingly, such trace data may also be stored in the trace buffer of the target endpoint node 108. In data flow 922, the target endpoint node 108 stores the trace buffer data to a portion of memory of the target endpoint

node 108 (e.g., the memory 406) allocated for trace data storage. In data flow 924, the target endpoint node 108 stores the extracted payload data to a portion of memory of the target endpoint node 108 (e.g., the memory 406) allocated for storage of application data.

[0080] Referring now to FIG. 10, in use, the network computing device 106 may execute a method 1000 for inserting trace data into a network packet. It should be appreciated that at least a portion of the method 1000 may be executed by the NIC 312 of the network computing device 106 or otherwise performed by the tracing engine 316 of the network computing device 106. It should be further appreciated that, in some embodiments, the method 1000 may be embodied as various instructions stored on a computer-readable media, which may be executed by the processor 302, the NIC 312, and/or other components of the network computing device 106 to cause the network computing device 106 to perform the method 1000. The computer-readable media may be embodied as any type of media capable of being read by the network computing device 106 including, but not limited to, the memory 306, the data storage device 308, the secure memory 314 of the NIC 312, other memory or data storage devices of the network computing device 106, portable media readable by a peripheral device of the network computing device 106, and/or other media.

[0081] The method 1000 begins with block 1002, in which the network computing device 106 determines whether a network packet has been received, such as may be received from the source endpoint node 102 or another network computing device 106. If so, the method 1000 advances to block 1004, wherein the network computing device 106 extracts one or more header fields from the network packet received in block 1002. In block 1006, the network computing device 106 determines whether performance tracing is enabled, such as may be determined from a trace data indicator located in one or more of the header fields extracted in block 1004 (e.g., a trace bit). As described previously, the trace data indicator may indicate whether performance tracing is enabled, a type of trace data to collect, a size of the trace data to collect, etc. If the trace data indicator was not detected, or the trace data indicator otherwise indicates not to collect trace data (i.e., performance tracing is disabled), the method 1000 branches to block 1022, wherein the network computing device 106 transmits the network packet to a target (e.g., another network computing device 106 or the target endpoint node 108).

[0082] Otherwise, if performance tracing is enabled, the method 1000 advances from block 1006 to block 1008. In some embodiments, in block 1008 the network computing device 106 may extract trace data from the received network packet. In such embodiments, in block 1010, the network computing device 106 may store the extracted trace data into a local trace buffer (e.g., the trace buffer 812 of FIG. 8). In some embodiments, all of the extracted trace

data may be saved, while in other embodiments, only a portion of the trace data may be saved. In some embodiments, in block 1012, the network computing device 106 additionally stores one or more of the header fields extracted in block 1004, such as to provide context for the stored trace data. Additionally, in some embodiments, the network computing device 106 may store at least a portion of the header field(s) extracted in block 1004.

[0083] As described previously, the trace data may include any type of information related to network traffic travelling through the HPC network 104, such as routing information, delay information, reception/transmission queue information, decision-making information, information regarding one or more characteristics of one or more components of the network computing device 106 (e.g., temperature, internal buffer usage, processor usage, memory usage, etc.) that processed/forwarded the network packet, etc. As also described previously, the trace data may additionally include a location of the trace data (i.e., the trace data portion of the network packet) and/or a maximum size indicator may be stored in a header field of the network packet, such that the indicators may be used to extract the trace data from the trace data portion of the network packet, as well as one or more counters, such as a hop counter that is incremented at every network computing device 106 the network packet is forwarded to.

[0084] In block 1014, the network computing device 106 collects trace data. To do so, in block 1016, the network computing device 106 retrieves trace data from the local trace buffer. It should be appreciated that the trace data retrieved from the local trace buffer may include additional trace data and/or updated trace data. It should be further appreciated that the trace data retrieved from the local trace buffer may be adjusted on a per-network packet bases. For example, an application, a performance tool, a selective mechanism, etc., at the source endpoint node 102 may indicate to the network computing device 106 that some network packets (e.g., based on a flow, a source indicator, a destination indicator, etc.) are to be traced at a higher level, while other network packets are to be traced at a reduced level or not at all (e.g., acknowledgement network packets, network packets generated to handle errors in other network packets, etc.). Accordingly, the trace data retrieve for one network packet may be different than trace data retrieved for another network packet. It should be appreciated that the trace data retrieved for one network packet may be based on a previous or other network packet. As described previously, in some embodiments, trace data may be generated for direct insertion into the trace data portion of the network packet (i.e., not stored in the local trace buffer). In such embodiments, in block 1018, the network computing device 106 generates the trace data to be collected.

[0085] In block 1020, the network computing device 106 inserts the trace data collected in block 1014 into a portion of the network packet dedicated to the trace data (i.e., a trace data portion of the network packet, such as may be determined from the location/size indicator in the trace data portion of the network packet). In block 1022, the network computing device 106 transmits the network packet to a target (e.g., another network computing device 106 or the target endpoint node 108) before the method 1000 returns to block 1002 to determine whether another network packet has been received.

[0086] Referring now to FIG. 11, in use, the target endpoint node 108 may execute a method 1100 for storing trace data of a network packet. It should be appreciated that at least a portion of the method 1100 may be executed by the NIC 412 of the target endpoint node 108 or otherwise performed by the tracing engine 416 of the target endpoint node 108. It should be further appreciated that, in some embodiments, the method 1100 may be embodied as various instructions stored on a computer-readable media, which may be executed by the processor 402, the NIC 412, and/or other components of the target endpoint node 108 to cause the target endpoint node 108 to perform the method 1100. The computer-readable media may be embodied as any type of media capable of being read by the target endpoint node 108 including, but not limited to, the memory 406, the data storage device 408, the secure memory 414 of the NIC 412, other memory or data storage devices of the target endpoint node 108, portable media readable by a peripheral device of the target endpoint node 108, and/or other media.

[0087] The method 1100 begins with block 1102, in which the target endpoint node 108 determines whether a network packet has been received, such as may be received from the network computing device 106. If so, the method 1100 advances to block 1104, wherein the target endpoint node 108 extracts one or more header fields from the network packet received in block 1102. In block 1106, the target endpoint node 108 extracts the payload from a payload portion of the received network packet. In block 1108, the target endpoint node 108 stores the payload of the network packet extracted in block 1106 to a portion of memory (e.g., the application memory 816 of FIG. 8) allocated for application data storage before the method returns to block 1102 to determine whether another network packet has been received.

[0088] In block 1110, the target endpoint node 108 determines whether performance tracing is enabled, such as may be determined from a trace data indicator located in one or more of the header fields extracted in block 1104. As described previously, the trace data indicator may indicate whether performance tracing is enabled, a type of trace data to collect, a size of the trace data to collect, etc. If the trace data indicator was not detected, or the trace data

indicator otherwise indicates not to collect trace data (i.e., performance tracing is disabled), the method 1100 returns to block 1102, wherein the target endpoint node 108 determines whether another network packet has been received.

[0089] Otherwise, if performance tracing is enabled, the method 1100 advances from block 1110 to block 1112, in which the target endpoint node 108 extracts trace data from a trace data portion of the received network packet. As described previously, the trace data may include any type of information related to network traffic travelling through the HPC network 104, such as routing information, delay information, reception/transmission queue information, decision-making information, information regarding one or more characteristics of one or more components of the network computing device 106 (e.g., temperature, internal buffer usage, processor usage, memory usage, etc.) that processed/forwarded the network packet, etc.

[0090] As also described previously, the trace data may include a location indicator that indicates a location in the trace data portion of the network packet at which the network computing device 106 is to store its trace data. Additionally, as also described previously, the trace data may include a maximum size indicator that indicates a maximum size of the trace data to be inserted by the network computing device 106, the start of which may be based on the location indicator. The trace data may additionally include a type indicator indicating what type of trace data and/or what format to store the trace data, and/or an identifying indicator, such as may be managed by one or more counters that are incremented at every network computing device 106 the network packet is forwarded to (e.g., a hop counter). In other words, the trace data may be paired with the counter value at a particular network computing device 106 and stored in the trace data portion, such that the target endpoint node 108 can determine a path and associate the trace data with the network computing devices 106 of the path.

[0091] In block 114, the target endpoint node 108 stores the extracted trace data into a local trace buffer. In some embodiments, all of the extracted trace data may be saved, while in other embodiments, only a portion of the trace data may be saved. In other words, in some embodiments, the target endpoint node 108 may discard at least a portion of the trace data extracted in block 1106 without writing to memory or compress the trace data such that only a portion of the trace data is saved. In such embodiments, the target endpoint node 108 may determine what trace data to save based on a random control feature, an identifier of the source endpoint node 102, a particular range of timestamps, values contained in the trace data (e.g., a value that indicated to only trace network packets with a delivery time that exceeds a predefined threshold). In some embodiments, the target endpoint node 108 may save the trace data to memory in a circular buffer (i.e., overwriting older trace data that has not been consumed).

Alternatively, in some embodiments, the target endpoint node 108 may save the trace data until a memory buffer is filled, in which case the target endpoint node 108 stops saving trace data to avoid overwriting older trace data contained in the memory buffer.

[0092] In some embodiments, in block 1116, the target endpoint node 108 additionally stores one or more of the header fields extracted in block 1104 with the stored trace data, such as to provide context for the stored trace data. As described previously, in some embodiments, the target endpoint node 108 may generate additional trace data not included in the network packet, such as an ingress time. In such embodiments, in block 1118, the target endpoint node 108 may additionally store any trace data generated by the target endpoint node 108 subsequent to having received the network packet at the target endpoint node 108 into the local trace buffer.

[0093] In block 1120, the target endpoint node 108 transfers the trace data from the local trace buffer into a portion of memory (e.g., the trace memory 818) that has been allocated for trace data storage external to the trace buffer and the NIC 412. It should be appreciated that the trace data retrieved from the local trace buffer may include additional trace data and/or updated trace data. For example, the target endpoint node 108 may determine a trip time between the source endpoint node 102 and the target endpoint node 108 based on a timestamp generated at the time of ingress at the target endpoint node 108 and another timestamp generated at the time of egress from the source endpoint node 102 that is stored in the trace data.

[0094] In block 1122, the target endpoint node 108 analyzes the trace data to determine one or more network performance characteristics of the HPC network 104. As described previously, the network performance characteristics may include delay information, paths of each of the traced network packets, queue fullness at each network computing device 102 in the path, durations of time each of the traced network packets spend at each network computing device 102 in the path, and/or any other data related to the performance of the network traffic through the HPC network 104. As such, based on the network performance characteristics, various issues of the HPC network 104 may be detected and the network computing device(s) 106 responsible for the detected issues may be identified.

EXAMPLES

[0095] Illustrative examples of the technologies disclosed herein are provided below. An embodiment of the technologies may include any one or more, and any combination of, the examples described below.

[0096] Example 1 includes a network computing device for tracing network performance, the network computing device comprising one or more processors; and one or more data storage devices having stored therein a plurality of instructions that, when executed by the one or more processors, cause the network computing device to receive a network packet from a source endpoint node, wherein the network packet includes a header and a trace data portion; generate trace data corresponding to the received network packet; update the trace data portion of the network packet to include the generated trace data from the trace buffer of the network computing device; and transmit the updated network packet to a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

[0097] Example 2 includes the subject matter of Example 1, and wherein the trace data portion includes trace data of the source endpoint node.

[0098] Example 3 includes the subject matter of any of Examples 1 and 2, and wherein the plurality of instructions further cause the network computing device to extract at least a portion of the trace data from the trace data portion of the network packet; and store the extracted portion of trace data to a trace buffer of the network computing device.

[0099] Example 4 includes the subject matter of any of Examples 1-3, and wherein the plurality of instructions further cause the network computing device to retrieve the stored portion of the trace data from the trace buffer, wherein to update the trace data portion of the network packet comprise to update the trace data portion of the network packet with the retrieved portion of the trace data.

[00100] Example 5 includes the subject matter of any of Examples 1-4, and wherein the plurality of instructions further cause the network computing device to extract at least a portion of the trace data from the trace data portion of the network packet; and store the extracted portion of trace data to a trace buffer of the network computing device.

[00101] Example 6 includes the subject matter of any of Examples 1-5, and wherein the plurality of instructions further cause the network computing device to extract at least a portion of the header of the network packet; and store the extracted portion of the header to a trace buffer of the network computing device.

[00102] Example 7 includes the subject matter of any of Examples 1-6, and wherein the plurality of instructions further cause the network computing device to parse the header of the network packet to retrieve an indicator inserted by the source endpoint node; and determine whether the indicator indicates to collect trace data, wherein to store the trace data of the

network packet to the trace buffer of the network computing device comprises to store the trace data subsequent to a determination that the indicator indicates to collect trace data.

[00103] Example 8 includes the subject matter of any of Examples 1-7, and wherein the plurality of instructions further cause the network computing device to identify a type of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the type of the trace data to be collected.

[00104] Example 9 includes the subject matter of any of Examples 1-8, and wherein the plurality of instructions further cause the network computing device to identify a size of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the size of the trace data to be collected.

[00105] Example 10 includes the subject matter of any of Examples 1-9, and wherein the plurality of instructions further cause the network computing device to generate trace data for direct insertion into the trace data portion, and wherein to update the trace data portion of the network packet further comprises to include the generated trace data in the trace data portion.

[00106] Example 11 includes the subject matter of any of Examples 1-10, and wherein to generate the trace data comprises to generate at least one of a time of interest, routing information, delay information, decision-making information, and information corresponding to one or more characteristics of a component the network computing device.

[00107] Example 12 includes the subject matter of any of Examples 1-11, and wherein to store the trace data to the trace buffer of the network computing device comprises to store at least a portion of the trace data of the network packet to the trace buffer.

[00108] Example 13 includes the subject matter of any of Examples 1-12, and wherein to store at least a portion of the trace data of the network packet comprises to (i) discard trace data without storing to a memory of the network computing device, (ii) filter the trace data based on an identifier of the source endpoint node, and (iii) store only the portion of the trace data according to a threshold associated with the trace data.

[00109] Example 14 includes the subject matter of any of Examples 1-13, and wherein to store the trace data of the network packet comprises to store the trace data to the trace buffer until the trace buffer is full.

[00110] Example 15 includes the subject matter of any of Examples 1-14, and wherein the trace buffer is a circular buffer.

[00111] Example 16 includes a method for tracing network performance, the method comprising receiving, by a network computing device, a network packet from a source endpoint node, wherein the network packet includes a header and a trace data portion; generating, by the

network computing device, trace data corresponding to the received network packet; updating, by the network computing device, the trace data portion of the network packet to include the generated trace data from the trace buffer of the network computing device; and transmitting, by the network computing device, the updated network packet to a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

[00112] Example 17 includes the subject matter of Example 16, and wherein the trace data portion includes trace data of the source endpoint node.

[00113] Example 18 includes the subject matter of any of Examples 16 and 17, and further including extracting, by the network computing device, at least a portion of the trace data from the trace data portion of the network packet; and storing, by the network computing device, the extracted portion of trace data to a trace buffer of the network computing device.

[00114] Example 19 includes the subject matter of any of Examples 16-18, and further including retrieving, by the network computing device, the stored portion of the trace data from the trace buffer, wherein updating the trace data portion of the network packet comprise updating the trace data portion of the network packet with the retrieved portion of the trace data.

[00115] Example 20 includes the subject matter of any of Examples 16-19, and further including extracting, by the network computing device, at least a portion of the trace data from the trace data portion of the network packet; and storing, by the network computing device, the extracted portion of trace data to a trace buffer of the network computing device.

[00116] Example 21 includes the subject matter of any of Examples 16-20, and further including extracting, by the network computing device, at least a portion of the header of the network packet; and storing, by the network computing device, the extracted portion of the header to a trace buffer of the network computing device.

[00117] Example 22 includes the subject matter of any of Examples 16-21, and further including parsing, by the network computing device, the header of the network packet to retrieve an indicator inserted by the source endpoint node; and determining, by the network computing device, whether the indicator indicates to collect trace data, wherein storing, by the network computing device, the trace data of the network packet to the trace buffer of the network computing device comprises storing the trace data subsequent to a determination that the indicator indicates to collect trace data.

[00118] Example 23 includes the subject matter of any of Examples 16-22, and further including identifying, by the network computing device, a type of the trace data to be collected

based on the indicator, wherein generating the trace data is based on the type of the trace data to be collected.

[00119] Example 24 includes the subject matter of any of Examples 16-23, and further including identifying, by the network computing device, a size of the trace data to be collected based on the indicator, wherein generating the trace data is based on the size of the trace data to be collected.

[00120] Example 25 includes the subject matter of any of Examples 16-24, and further including generating, by the network computing device, trace data for direct insertion into the trace data portion, and wherein updating the trace data portion of the network packet further comprises including the generated trace data in the trace data portion.

[00121] Example 26 includes the subject matter of any of Examples 16-25, and wherein generating the trace data comprises generating at least one of a time of interest, routing information, delay information, decision-making information, and information corresponding to one or more characteristics of a component the network computing device.

[00122] Example 27 includes the subject matter of any of Examples 16-26, and wherein storing the trace data to the trace buffer of the network computing device comprises storing at least a portion of the trace data of the network packet.

[00123] Example 28 includes the subject matter of any of Examples 16-27, and wherein storing at least a portion of the trace data of the network packet comprises (i) discarding trace data without storing to a memory of the network computing device, (ii) filtering the trace data based on an identifier of the source endpoint node, and (iii) storing only the portion of the trace data according to a threshold associated with the trace data.

[00124] Example 29 includes the subject matter of any of Examples 16-28, and wherein storing the trace data of the network packet comprises storing the trace data to the trace buffer until the trace buffer is full.

[00125] Example 30 includes the subject matter of any of Examples 16-29, and wherein storing the trace data of the network packet to the trace buffer comprises saving the trace data to a circular buffer.

[00126] Example 31 includes a network computing device comprising a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the network computing device to perform the method of any of Examples 15-30.

[00127] Example 32 includes one or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a network computing device performing the method of any of Examples 15-30.

[00128] Example 33 includes a network computing device for tracing network performance, the network computing device comprising network communication management circuitry to receive a network packet from a source endpoint node, wherein the network packet includes a header and a trace data portion; and performance tracing circuitry to (i) generate trace data corresponding to the received network packet and (ii) update the trace data portion of the network packet to include the generated trace data from the trace buffer of the network computing device, wherein the network communication management circuitry is further to transmit the updated network packet to a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

[00129] Example 34 includes the subject matter of Example 33, and wherein the trace data portion includes trace data of the source endpoint node.

[00130] Example 35 includes the subject matter of any of Examples 33 and 34, and wherein the performance tracing circuitry is further to (i) extract at least a portion of the trace data from the trace data portion of the network packet and (ii) store the extracted portion of trace data to a trace buffer of the network computing device.

[00131] Example 36 includes the subject matter of any of Examples 33-35, and wherein the performance tracing circuitry is further to retrieve the stored portion of the trace data from the trace buffer, wherein to update the trace data portion of the network packet comprise to update the trace data portion of the network packet with the retrieved portion of the trace data.

[00132] Example 37 includes the subject matter of any of Examples 33-36, and wherein the performance tracing circuitry is further to (i) extract at least a portion of the trace data from the trace data portion of the network packet and (ii) store the extracted portion of trace data to a trace buffer of the network computing device.

[00133] Example 38 includes the subject matter of any of Examples 33-37, and wherein the performance tracing circuitry is further to (i) extract at least a portion of the header of the network packet and (ii) store the extracted portion of the header to a trace buffer of the network computing device.

[00134] Example 39 includes the subject matter of any of Examples 33-38, and wherein the performance tracing circuitry is further to (i) parse the header of the network packet to retrieve an indicator inserted by the source endpoint node and (ii) determine whether the indicator indicates to collect trace data, wherein to store the trace data of the network packet to the trace buffer of the network computing device comprises to store the trace data subsequent to a determination that the indicator indicates to collect trace data.

[00135] Example 40 includes the subject matter of any of Examples 33-39, and wherein the performance tracing circuitry is further to identify a type of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the type of the trace data to be collected.

[00136] Example 41 includes the subject matter of any of Examples 33-40, and wherein the plurality of instructions further cause the network computing device to identify a size of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the size of the trace data to be collected.

[00137] Example 42 includes the subject matter of any of Examples 33-41, and wherein the performance tracing circuitry is further to generate trace data for direct insertion into the trace data portion, and wherein to update the trace data portion of the network packet further comprises to include the generated trace data in the trace data portion.

[00138] Example 43 includes the subject matter of any of Examples 33-42, and wherein to generate the trace data comprises to generate at least one of a time of interest, routing information, delay information, decision-making information, and information corresponding to one or more characteristics of a component the network computing device.

[00139] Example 44 includes the subject matter of any of Examples 33-43, and wherein to store the trace data to the trace buffer of the network computing device comprises to store at least a portion of the trace data of the network packet.

[00140] Example 45 includes the subject matter of any of Examples 33-44, and wherein to store at least a portion of the trace data of the network packet comprises to (i) discard trace data without storing to a memory of the network computing device, (ii) filter the trace data based on an identifier of the source endpoint node, and (iii) store only the portion of the trace data according to a threshold associated with the trace data.

[00141] Example 46 includes the subject matter of any of Examples 33-45, and wherein to store the trace data of the network packet comprises to store the trace data to the trace buffer until the trace buffer is full.

[00142] Example 47 includes the subject matter of any of Examples 33-46, and wherein the trace buffer is a circular buffer.

[00143] Example 48 includes a network computing device for tracing network performance, the network computing device comprising network communication management circuitry to receive a network packet from a source endpoint node, wherein the network packet includes a header and a trace data portion; and means for generating trace data corresponding to the received network packet; means for updating the trace data portion of the network packet to

include the generated trace data from the trace buffer of the network computing device; and wherein the network communication management circuitry is further to transmit the updated network packet to a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

[00144] Example 49 includes the subject matter of Example 48, and wherein the trace data portion includes trace data of the source endpoint node.

[00145] Example 50 includes the subject matter of any of Examples 48 and 49, and further including means for extracting at least a portion of the trace data from the trace data portion of the network packet; and means for storing the extracted portion of trace data to a trace buffer of the network computing device.

[00146] Example 51 includes the subject matter of any of Examples 48-50, and further including means for retrieving the stored portion of the trace data from the trace buffer, wherein the means for updating the trace data portion of the network packet comprise means for updating the trace data portion of the network packet with the retrieved portion of the trace data.

[00147] Example 52 includes the subject matter of any of Examples 48-51, and further including means for extracting at least a portion of the trace data from the trace data portion of the network packet; and means for storing the extracted portion of trace data to a trace buffer of the network computing device.

[00148] Example 53 includes the subject matter of any of Examples 48-52, and further including means for extracting at least a portion of the header of the network packet; and means for storing the extracted portion of the header to a trace buffer of the network computing device.

[00149] Example 54 includes the subject matter of any of Examples 48-53, and further including means for parsing the header of the network packet to retrieve an indicator inserted by the source endpoint node; and means for determining whether the indicator indicates to collect trace data, wherein the means for storing the trace data of the network packet to the trace buffer of the network computing device comprises means for storing the trace data subsequent to a determination that the indicator indicates to collect trace data.

[00150] Example 55 includes the subject matter of any of Examples 48-54, and further including means for identifying a type of the trace data to be collected based on the indicator, wherein generating the trace data is based on the type of the trace data to be collected.

[00151] Example 56 includes the subject matter of any of Examples 48-55, and further including means for identifying a size of the trace data to be collected based on the indicator, wherein generating the trace data is based on the size of the trace data to be collected.

[00152] Example 57 includes the subject matter of any of Examples 48-56, and further including means for generating trace data for direct insertion into the trace data portion, and wherein the means for updating the trace data portion of the network packet further comprises means for including the generated trace data in the trace data portion.

[00153] Example 58 includes the subject matter of any of Examples 48-57, and wherein the means for generating the trace data comprises means for generating at least one of a time of interest, routing information, delay information, decision-making information, and information corresponding to one or more characteristics of a component the network computing device.

[00154] Example 59 includes the subject matter of any of Examples 48-58, and wherein the means for storing the trace data to the trace buffer of the network computing device comprises means for storing at least a portion of the trace data of the network packet.

[00155] Example 60 includes the subject matter of any of Examples 48-59, and wherein the means for storing at least a portion of the trace data of the network packet comprises means for (i) discarding trace data without storing to a memory of the network computing device, (ii) filtering the trace data based on an identifier of the source endpoint node, and (iii) storing only the portion of the trace data according to a threshold associated with the trace data.

[00156] Example 61 includes the subject matter of any of Examples 48-60, and wherein the means for storing the trace data of the network packet comprises means for storing the trace data to the trace buffer until the trace buffer is full.

[00157] Example 62 includes the subject matter of any of Examples 48-61, and wherein the means for storing the trace data of the network packet to the trace buffer comprises means for saving the trace data to a circular buffer.

[00158] Example 63 includes a source endpoint node for tracing network performance, the source endpoint node including one or more processors; and one or more data storage devices having stored therein a plurality of instructions that, when executed by the one or more processors, cause the source endpoint node to generate a network packet that includes a header with a plurality of fields; allocate a trace data portion in the data portion of the network packet; insert a trace data indicator into one of the fields of the header, wherein the trace data indicator indicates whether the network packet includes a trace data portion; and transmit the generated network packet towards a target endpoint node, wherein the generated network packet is usable to determine one or more network performance characteristics.

[00159] Example 64 includes the subject matter of Example 63, and wherein the plurality of instructions further cause the source endpoint node to generate trace data for the network packet and insert the generated trace data into the trace data portion of the allocated trace data portion.

[00160] Example 65 includes the subject matter of any of Examples 63 and 64, and wherein to transmit the generated network packet towards the target endpoint node comprises to transmit the generated network packet to a network computing device, and wherein the trace data indicator is usable by the network computing device to determine whether to generate trace data of the network computing device.

[00161] Example 66 includes the subject matter of any of Examples 63-65, and wherein the plurality of instructions further cause the source endpoint node to insert a location indicator that indicates a location in the trace data portion of the network packet at which the network computing device is to store the trace data of the network computing device.

[00162] Example 67 includes the subject matter of any of Examples 63-66, and wherein the plurality of instructions further cause the source endpoint node to insert a size indicator that indicates a maximum size in the trace data portion of the network packet allocated to the network computing device to store the trace data of the network computing device.

[00163] Example 68 includes the subject matter of any of Examples 63-67, and wherein the plurality of instructions further cause the source endpoint node to insert a type indicator that indicates a type of trace data the network computing device is to generate.

[00164] Example 69 includes a method for tracing network performance, the method including generating, by a source endpoint node, a network packet that includes a header with a plurality of fields; allocating, by the source endpoint node, a trace data portion in the data portion of the network packet; inserting, by the source endpoint node, a trace data indicator into one of the fields of the header, wherein the trace data indicator indicates whether the network packet includes a trace data portion; and transmitting, by the source endpoint node, the generated network packet towards a target endpoint node, wherein the generated network packet is usable to determine one or more network performance characteristics.

[00165] Example 70 includes the subject matter of Example 69, and further comprising generating, by the source endpoint node, trace data for the network packet and insert the generated trace data into the trace data portion of the allocated trace data portion.

[00166] Example 71 includes the subject matter of any of Examples 69 and 70, and wherein transmitting the generated network packet towards the target endpoint node comprises transmitting the generated network packet to a network computing device, and wherein the

trace data indicator is usable by the network computing device to determine whether to generate trace data of the network computing device.

[00167] Example 72 includes the subject matter of any of Examples 69-71, and further comprising inserting, by the source endpoint node, a location indicator that indicates a location in the trace data portion of the network packet at which the network computing device is to store the trace data of the network computing device.

[00168] Example 73 includes the subject matter of any of Examples 69-72, and further comprising inserting, by the source endpoint node, a size indicator that indicates a maximum size in the trace data portion of the network packet allocated to the network computing device to store the trace data of the network computing device.

[00169] Example 74 includes the subject matter of any of Examples 69-73, and further comprising inserting, by the source endpoint node, a type indicator that indicates a type of trace data the network computing device is to generate.

[00170] Example 75 includes a source endpoint node comprising a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the source endpoint node to perform the method of any of claims 69-74.

[00171] Example 76 includes one or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a source endpoint node performing the method of any of claims 69-74.

[00172] Example 77 includes a source endpoint node comprising means for performing the method of any of claims 69-74.

[00173] Example 78 includes a target endpoint node for tracing network performance, the target endpoint node including one or more processors; and one or more data storage devices having stored therein a plurality of instructions that, when executed by the one or more processors, cause the source endpoint node to receive a network packet from a network computing device, wherein the network packet includes a header with a plurality of fields; extract trace data from a trace data portion of the network packet, wherein the trace data includes data generated by a plurality of network computing devices in which the network packet was received at and forwarded from, wherein the plurality of network computing devices includes the network computing device; and store trace data at a trace data buffer of a network interface controller of the target endpoint node.

[00174] Example 79 includes the subject matter of Example 78, and wherein the plurality of instructions further cause the target endpoint node to (i) parse a predetermined one of the fields of the header to extract a trace data indicator and (ii) determine whether the network

packet includes a trace data portion based on the trace data indicator, and wherein to extract the trace data from the trace data portion of the network packet comprises to extract the trace data in response to a determination that the network packet includes the trace data portion.

[00175] Example 80 includes the subject matter of any of Examples 78 and 79, and wherein the plurality of instructions further cause the target endpoint node to move the trace data from the trace data buffer to a portion of memory external to the network interface controller allocated to store the trace data.

[00176] Example 81 includes the subject matter of any of Examples 78-80, and wherein the plurality of instructions further cause the target endpoint node to (i) analyze the trace data stored in the allocated portion of memory external to the network interface controller and (ii) identify one or more network performance characteristics based on the analysis.

[00177] Example 82 includes the subject matter of any of Examples 78-81, and wherein the network packet further includes a payload portion, and wherein the plurality of instructions further cause the target endpoint node to (i) extract the payload portion of the network packet and (ii) store the extracted payload portion into a portion of memory external to the network interface controller allocated to store the payload.

[00178] Example 83 includes a method for tracing network performance, the method including receiving, by a target endpoint node, a network packet from a network computing device, wherein the network packet includes a header with a plurality of fields; extracting, by the target endpoint node, trace data from a trace data portion of the network packet, wherein the trace data includes data generated by a plurality of network computing devices in which the network packet was received at and forwarded from, wherein the plurality of network computing devices includes the network computing device; and storing, by the target endpoint node, trace data at a trace data buffer of a network interface controller of the target endpoint node.

[00179] Example 84 includes the subject matter of Example 83, and further comprising parsing, by the target endpoint node, a predetermined one of the fields of the header to extract a trace data indicator; and determining, by the target endpoint node, whether the network packet includes a trace data portion based on the trace data indicator, wherein extracting the trace data from the trace data portion of the network packet comprises extracting the trace data in response to a determination that the network packet includes the trace data portion.

[00180] Example 85 includes the subject matter of any of Examples 83 and 84, and further comprising moving, by the target endpoint node, the trace data from the trace data

buffer to a portion of memory external to the network interface controller allocated to store the trace data.

[00181] Example 86 includes the subject matter of any of Examples 83-85, and further comprising: analyzing, by the target endpoint node, the trace data stored in the allocated portion of memory external to the network interface controller; and identifying, by the target endpoint node, one or more network performance characteristics based on the analysis.

[00182] Example 87 includes the subject matter of any of Examples 83-86, and further comprising: extracting, by the target endpoint node, a payload portion of the network packet; and storing, by the target endpoint node, the extracted payload portion into a portion of memory external to the network interface controller allocated to store the payload.

[00183] Example 88 includes a target endpoint node including a processor; and a memory having stored therein a plurality of instructions that when executed by the processor cause the target endpoint node to perform the method of any of claims 83-87.

[00184] Example 89 includes one or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a target endpoint node performing the method of any of claims 83-87.

[00185] Example 90 includes a target endpoint node comprising means for performing the method of any of claims 83-87.

WHAT IS CLAIMED IS:

1. A network computing device for tracing network performance, the network computing device comprising:

one or more processors; and

one or more data storage devices having stored therein a plurality of instructions that, when executed by the one or more processors, cause the network computing device to:

receive a network packet generated by a source endpoint node, wherein the network packet includes a header and a trace data portion;

generate trace data corresponding to the received network packet;

update the trace data portion of the network packet to include the generated trace data from the trace buffer of the network computing device; and

transmit the updated network packet towards a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

2. The network computing device of claim 1, wherein the plurality of instructions further cause the network computing device to:

extract at least a portion of the trace data from the trace data portion of the network packet; and

store the extracted portion of trace data to a trace buffer of the network computing device.

3. The network computing device of claim 2, wherein the plurality of instructions further cause the network computing device to retrieve the stored portion of the trace data from the trace buffer, wherein to update the trace data portion of the network packet comprise to update the trace data portion of the network packet with the retrieved portion of the trace data.

4. The network computing device of claim 3, wherein the plurality of instructions further cause the network computing device to:

extract at least a portion of the trace data from the trace data portion of the network packet; and

store the extracted portion of trace data to a trace buffer of the network computing device.

5. The network computing device of claim 4, wherein the plurality of instructions further cause the network computing device to:

extract at least a portion of the header of the network packet; and

store the extracted portion of the header to a trace buffer of the network computing device.

6. The network computing device of claim 1, wherein the plurality of instructions further cause the network computing device to:

parse the header of the network packet to retrieve an indicator inserted by the source endpoint node; and

determine whether the indicator indicates to collect trace data,

wherein to store the trace data of the network packet to the trace buffer of the network computing device comprises to store the trace data subsequent to a determination that the indicator indicates to collect trace data.

7. The network computing device of claim 6, wherein the plurality of instructions further cause the network computing device to identify a type of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the type of the trace data to be collected.

8. The network computing device of claim 6, wherein the plurality of instructions further cause the network computing device to identify a size of the trace data to be collected based on the indicator, wherein to generate the trace data is based on the size of the trace data to be collected.

9. The network computing device of claim 1, wherein the plurality of instructions further cause the network computing device to generate trace data for direct insertion into the trace data portion, and wherein to update the trace data portion of the network packet further comprises to include the generated trace data in the trace data portion.

10. The network computing device of claim 1, wherein to generate the trace data comprises to generate at least one of a time of interest, routing information, delay information, decision-making information, and information corresponding to one or more characteristics of a component the network computing device.

11. The network computing device of claim 1, wherein to store the trace data to the trace buffer of the network computing device comprises to store at least a portion of the trace data of the network packet to the trace buffer.

12. The network computing device of claim 11, wherein to store at least a portion of the trace data of the network packet comprises to (i) discard trace data without storing to a memory of the network computing device, (ii) filter the trace data based on an identifier of the source endpoint node, and (iii) store only the portion of the trace data according to a threshold associated with the trace data.

13. A method for tracing network performance, the method comprising:

receiving, by a network computing device, a network packet generated by a source endpoint node, wherein the network packet includes a header and a trace data portion;

generating, by the network computing device, trace data corresponding to the received network packet;

updating, by the network computing device, the trace data portion of the network packet to include the generated trace data from the trace buffer of the network computing device; and

transmitting, by the network computing device, the updated network packet towards a target endpoint node, wherein the updated network packet is usable to determine one or more network performance characteristics.

14. The method of claim 13, further comprising:

extracting, by the network computing device, at least a portion of the trace data from the trace data portion of the network packet; and

storing, by the network computing device, the extracted portion of trace data to a trace buffer of the network computing device.

15. The method of claim 14, further comprising retrieving, by the network computing device, the stored portion of the trace data from the trace buffer, wherein updating the trace data portion of the network packet comprise updating the trace data portion of the network packet with the retrieved portion of the trace data.

16. The method of claim 15, further comprising:

extracting, by the network computing device, at least a portion of the trace data from the trace data portion of the network packet; and

storing, by the network computing device, the extracted portion of trace data to a trace buffer of the network computing device.

17. The method of claim 16, further comprising:

extracting, by the network computing device, at least a portion of the header of the network packet; and

storing, by the network computing device, the extracted portion of the header to a trace buffer of the network computing device.

18. The method of claim 13, further comprising:

parsing, by the network computing device, the header of the network packet to retrieve an indicator inserted by the source endpoint node; and

determining, by the network computing device, whether the indicator indicates to collect trace data,

wherein storing, by the network computing device, the trace data of the network packet to the trace buffer of the network computing device comprises storing the trace data subsequent to a determination that the indicator indicates to collect trace data.

19. The method of claim 18, further comprising identifying, by the network computing device, a type of the trace data to be collected based on the indicator, wherein generating the trace data is based on the type of the trace data to be collected.

20. The method of claim 18, further comprising identifying, by the network computing device, a size of the trace data to be collected based on the indicator, wherein generating the trace data is based on the size of the trace data to be collected.

21. The method of claim 13, further comprising generating, by the network computing device, trace data for direct insertion into the trace data portion, and wherein updating the trace data portion of the network packet further comprises including the generated trace data in the trace data portion.

22. The method of claim 13, wherein generating the trace data comprises generating at least one of a time of interest, routing information, delay information, decision-making

information, and information corresponding to one or more characteristics of a component the network computing device.

23. The method of claim **13,** wherein storing the trace data to the trace buffer of the network computing device comprises storing at least a portion of the trace data of the network packet, wherein storing the at least a portion of the trace data of the network packet comprises (i) discarding trace data without storing to a memory of the network computing device, (ii) filtering the trace data based on an identifier of the source endpoint node, and (iii) storing only the portion of the trace data according to a threshold associated with the trace data.

24. **A** network computing device comprising:

a processor; and

a memory having stored therein a plurality of instructions that when executed by the processor cause the network computing device to perform the method of any of claims **13-23.**

25. One or more machine readable storage media comprising a plurality of instructions stored thereon that in response to being executed result in a network computing device performing the method of any of claims **13-23.**

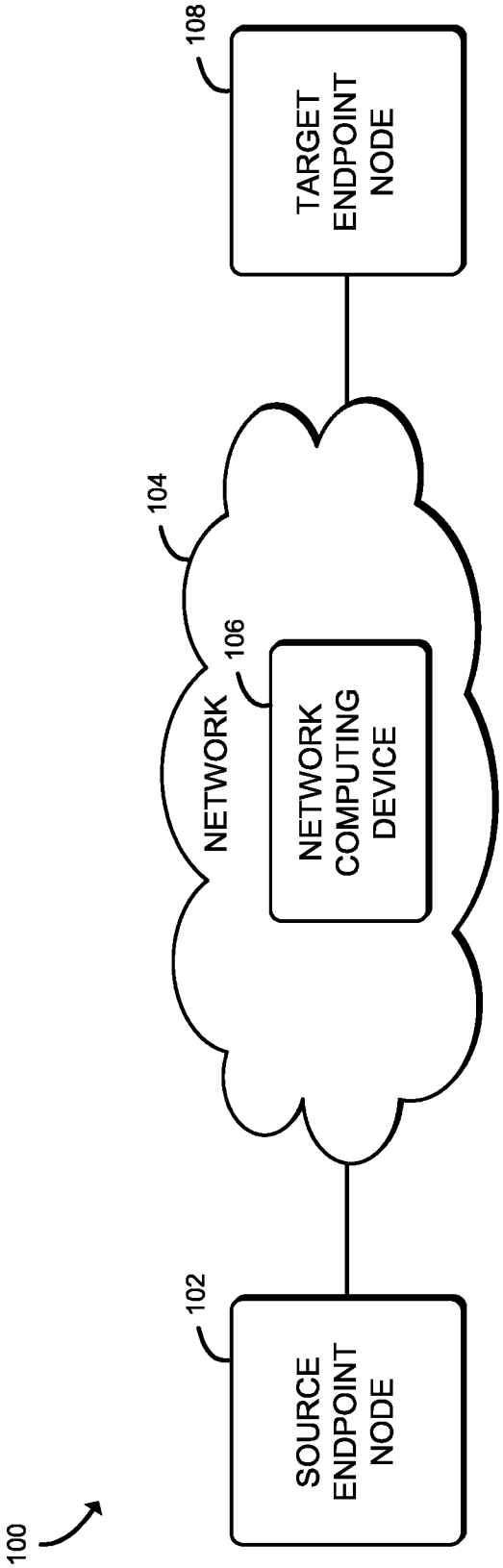
100
SOURCE ENDPOINT NODE
102
NETWORK
104
NETWORK COMPUTING DEVICE
106
TARGET ENDPOINT NODE
108

FIG. 1

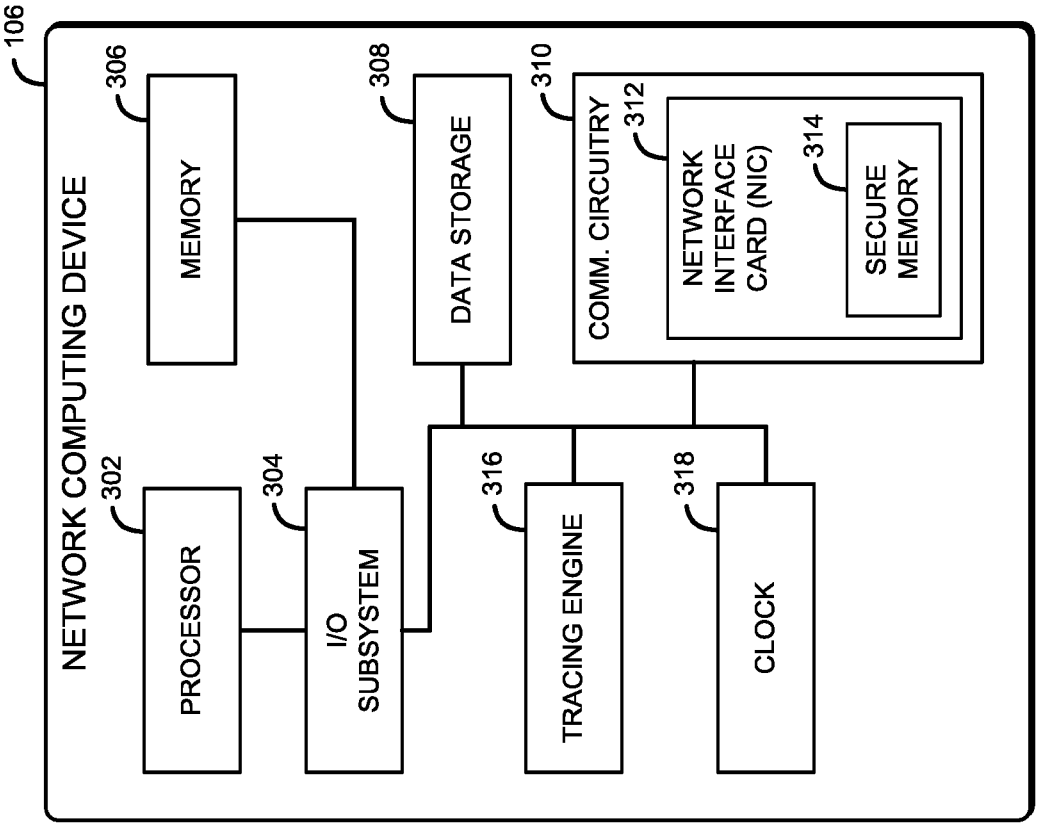
106
NETWORK COMPUTING DEVICE
306
MEMORY
308
DATA STORAGE
310
COMM. CIRCUITRY
312
NETWORK INTERFACE CARD (NIC)
314
SECURE MEMORY
302
PROCESSOR
304
I/O SUBSYSTEM
316
TRACING ENGINE
318
CLOCK

FIG. 3

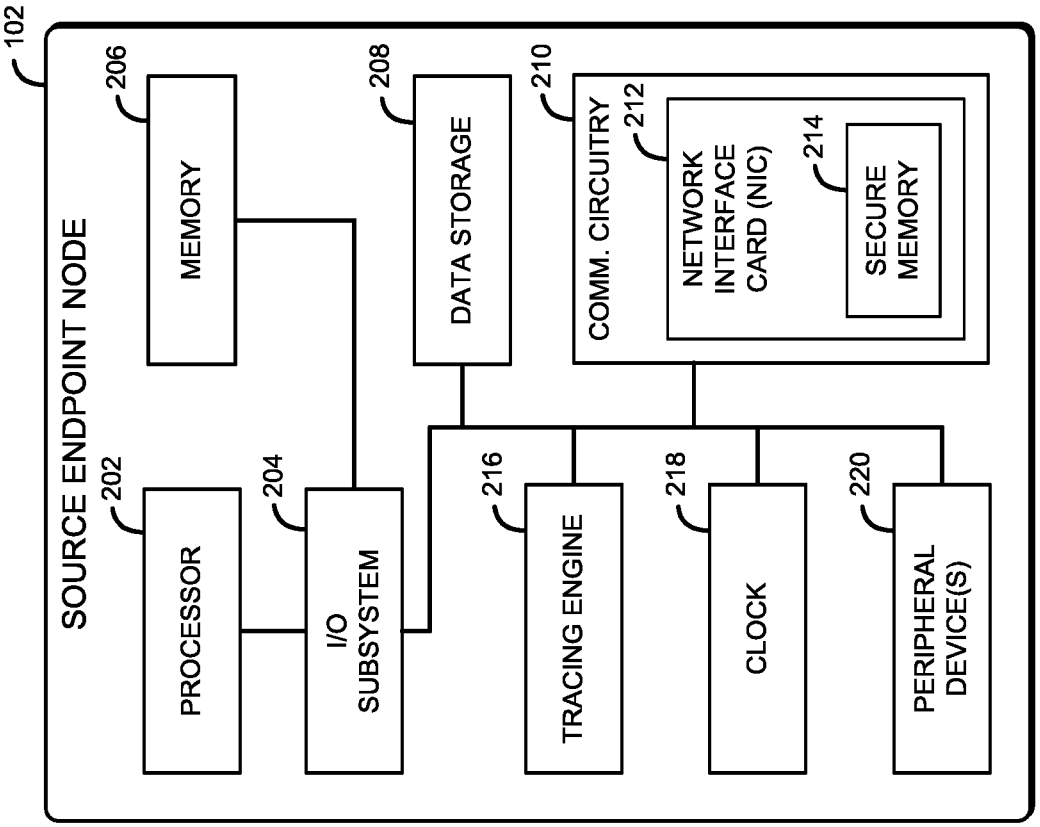
102
SOURCE ENDPOINT NODE
206
MEMORY
208
DATA STORAGE
210
COMM. CIRCUITRY
212
NETWORK INTERFACE CARD (NIC)
214
SECURE MEMORY
202
PROCESSOR
204
I/O SUBSYSTEM
216
TRACING ENGINE
218
CLOCK
220
PERIPHERAL DEVICE(S)

FIG. 2

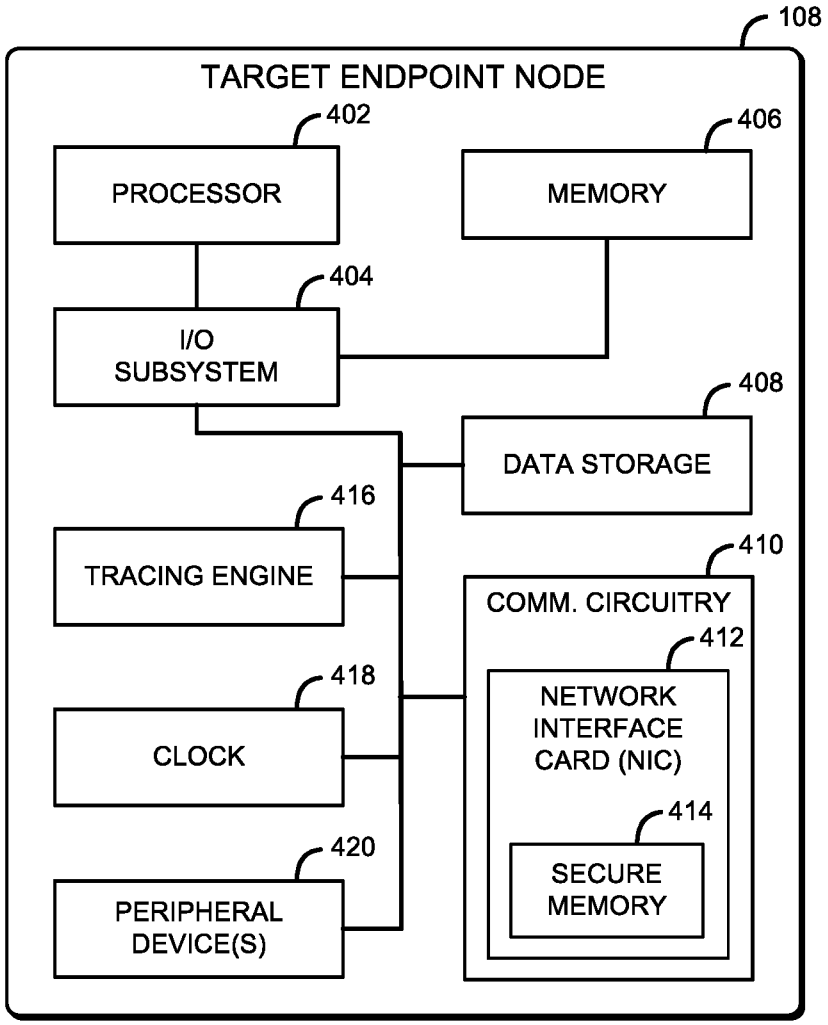
108
TARGET ENDPOINT NODE
402
PROCESSOR
406
MEMORY
404
I/O SUBSYSTEM
408
DATA STORAGE
416
TRACING ENGINE
410
COMM. CIRCUITRY
412
NETWORK INTERFACE CARD (NIC)
418
CLOCK
414
SECURE MEMORY
420
PERIPHERAL DEVICE(S)

FIG. 4

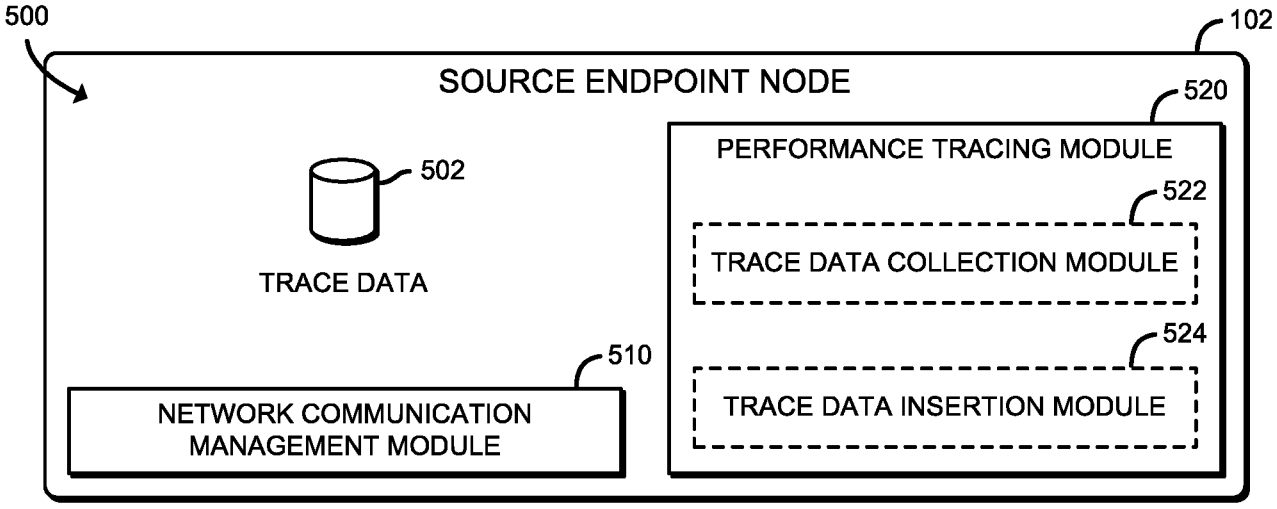
500
102
SOURCE ENDPOINT NODE
520
PERFORMANCE TRACING MODULE
502
TRACE DATA
522
TRACE DATA COLLECTION MODULE
524
TRACE DATA INSERTION MODULE
510
NETWORK COMMUNICATION MANAGEMENT MODULE

FIG. 5

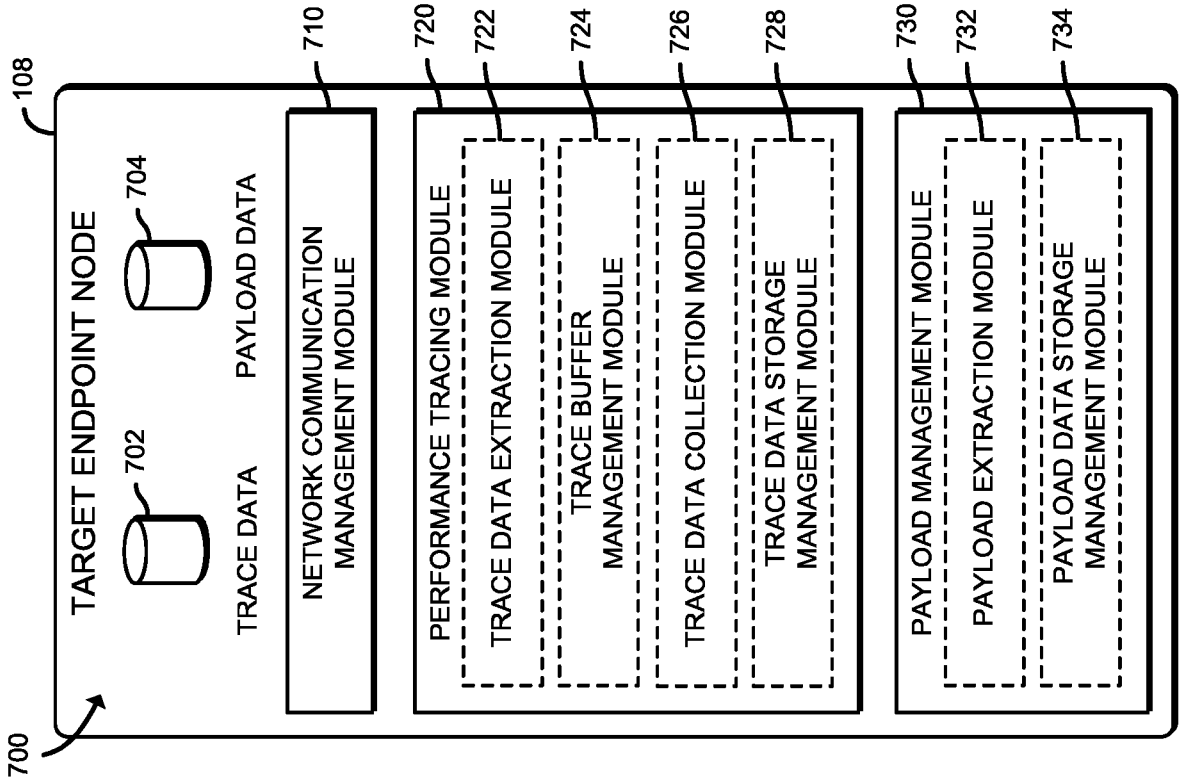
108
TARGET ENDPOINT NODE
700
TRACE DATA
702
PAYLOAD DATA
704
NETWORK COMMUNICATION MANAGEMENT MODULE
710
PERFORMANCE TRACING MODULE
720
TRACE DATA EXTRACTION MODULE
722
TRACE BUFFER MANAGEMENT MODULE
724
TRACE DATA COLLECTION MODULE
726
TRACE DATA STORAGE MANAGEMENT MODULE
728
PAYLOAD MANAGEMENT MODULE
730
PAYLOAD EXTRACTION MODULE
732
PAYLOAD DATA STORAGE MANAGEMENT MODULE
734

FIG. 7

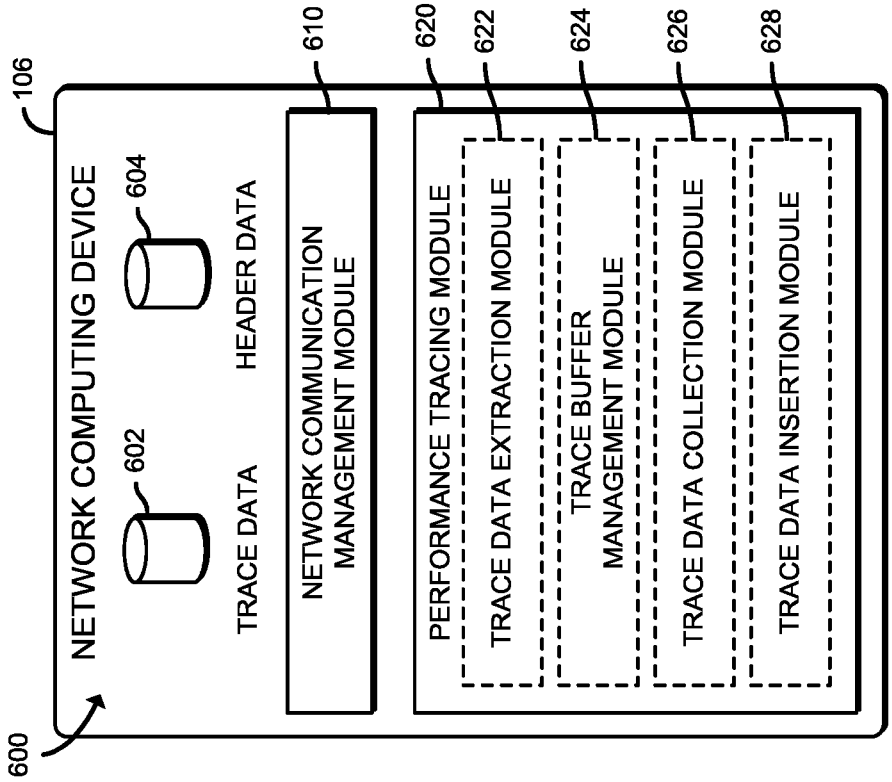
106
NETWORK COMPUTING DEVICE
600
TRACE DATA
602
HEADER DATA
604
NETWORK COMMUNICATION MANAGEMENT MODULE
610
PERFORMANCE TRACING MODULE
620
TRACE DATA EXTRACTION MODULE
622
TRACE BUFFER MANAGEMENT MODULE
624
TRACE DATA COLLECTION MODULE
626
TRACE DATA INSERTION MODULE
628

FIG. 6

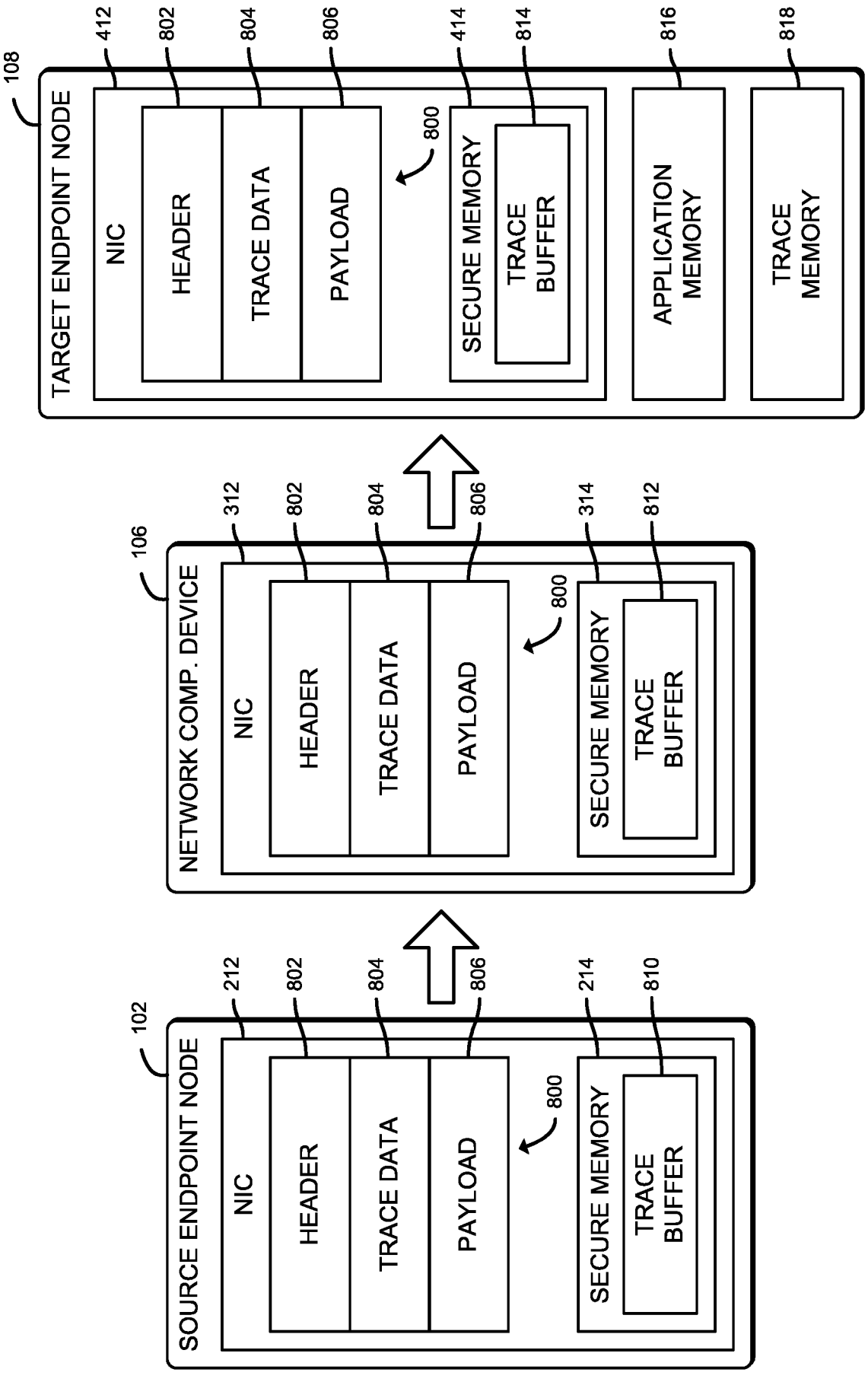
102
SOURCE ENDPOINT NODE
NIC
212
HEADER
802
TRACE DATA
804
PAYLOAD
806
800
SECURE MEMORY
214
TRACE BUFFER
810
106
NETWORK COMP. DEVICE
NIC
312
HEADER
802
TRACE DATA
804
PAYLOAD
806
800
SECURE MEMORY
314
TRACE BUFFER
812
108
TARGET ENDPOINT NODE
NIC
412
HEADER
802
TRACE DATA
804
PAYLOAD
806
800
SECURE MEMORY
414
TRACE BUFFER
814
APPLICATION MEMORY
816
TRACE MEMORY
818

FIG. 8

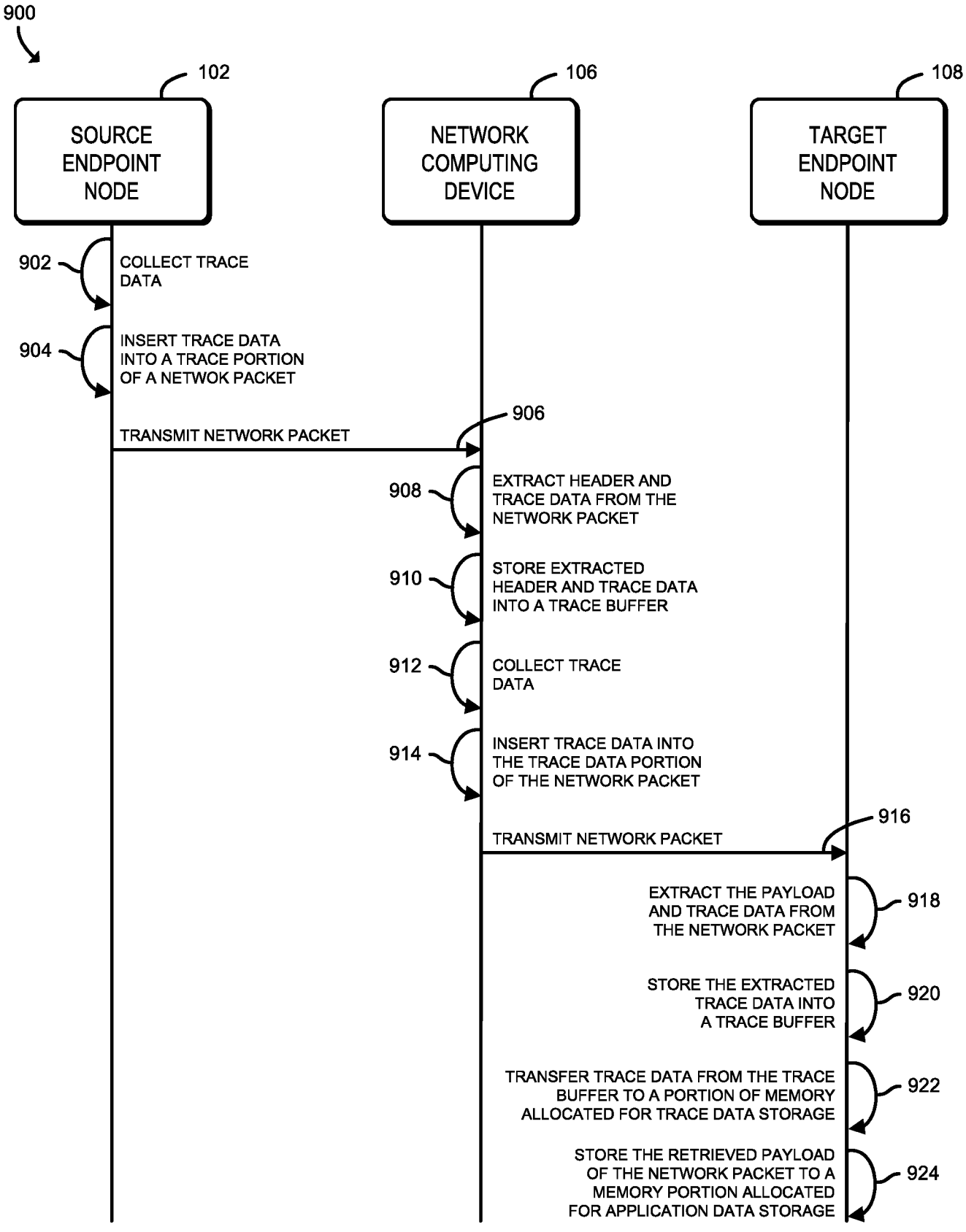
900
102
106
108
SOURCE ENDPOINT NODE
NETWORK COMPUTING DEVICE
TARGET ENDPOINT NODE
902
COLLECT TRACE DATA
904
INSERT TRACE DATA INTO A TRACE PORTION OF A NETWOK PACKET
906
TRANSMIT NETWORK PACKET
908
EXTRACT HEADER AND TRACE DATA FROM THE NETWORK PACKET
910
STORE EXTRACTED HEADER AND TRACE DATA INTO A TRACE BUFFER
912
COLLECT TRACE DATA
914
INSERT TRACE DATA INTO THE TRACE DATA PORTION OF THE NETWORK PACKET
916
TRANSMIT NETWORK PACKET
918
EXTRACT THE PAYLOAD AND TRACE DATA FROM THE NETWORK PACKET
920
STORE THE EXTRACTED TRACE DATA INTO A TRACE BUFFER
922
TRANSFER TRACE DATA FROM THE TRACE BUFFER TO A PORTION OF MEMORY ALLOCATED FOR TRACE DATA STORAGE
924
STORE THE RETRIEVED PAYLOAD OF THE NETWORK PACKET TO A MEMORY PORTION ALLOCATED FOR APPLICATION DATA STORAGE

FIG. 9

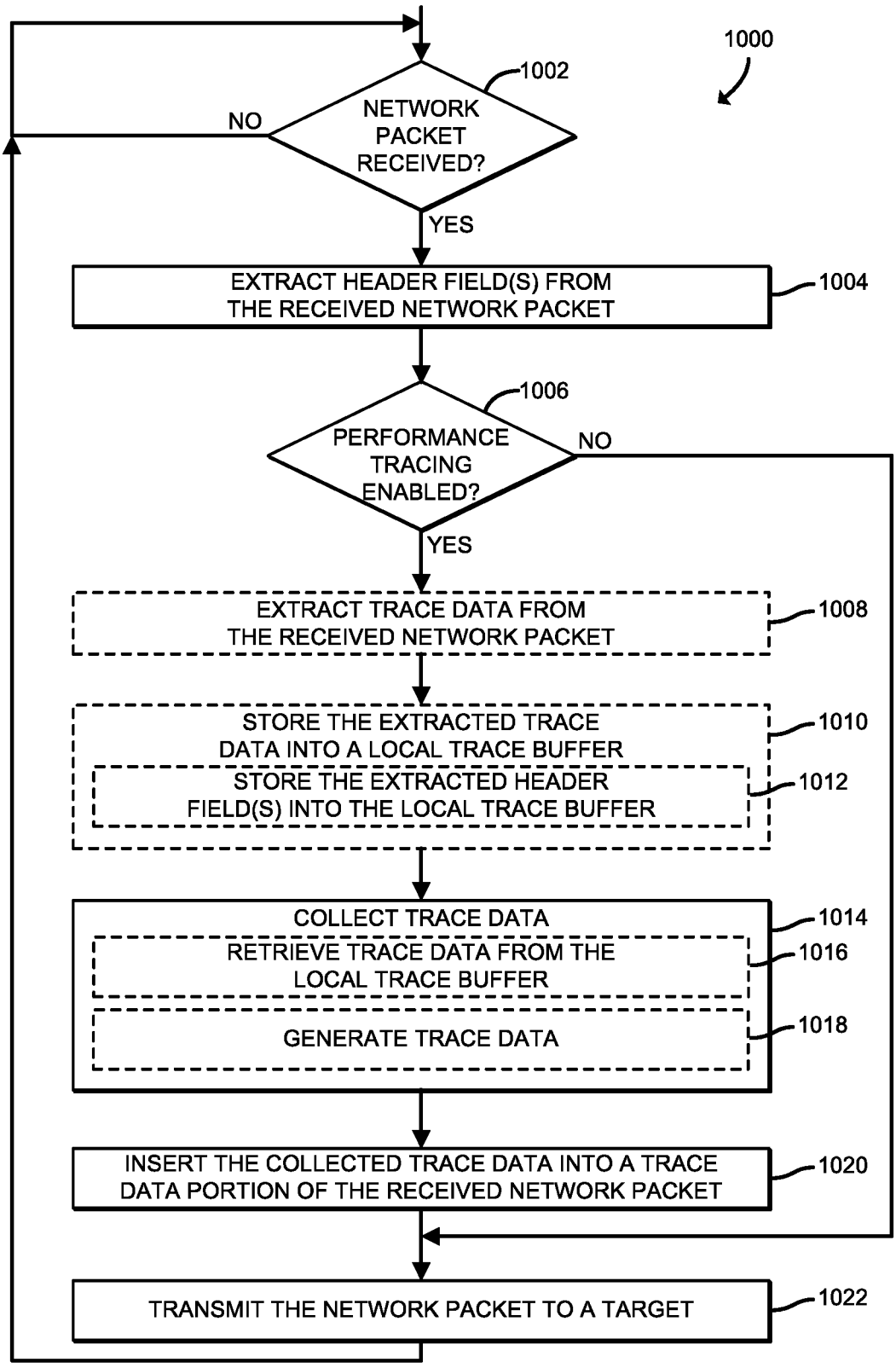
1000
1002
NETWORK PACKET RECEIVED?
NO
YES
EXTRACT HEADER FIELD(S) FROM THE RECEIVED NETWORK PACKET
1004
1006
PERFORMANCE TRACING ENABLED?
NO
YES
EXTRACT TRACE DATA FROM THE RECEIVED NETWORK PACKET
1008
STORE THE EXTRACTED TRACE DATA INTO A LOCAL TRACE BUFFER
1010
STORE THE EXTRACTED HEADER FIELD(S) INTO THE LOCAL TRACE BUFFER
1012
COLLECT TRACE DATA
1014
RETRIEVE TRACE DATA FROM THE LOCAL TRACE BUFFER
1016
GENERATE TRACE DATA
1018
INSERT THE COLLECTED TRACE DATA INTO A TRACE DATA PORTION OF THE RECEIVED NETWORK PACKET
1020
TRANSMIT THE NETWORK PACKET TO A TARGET
1022

FIG. 10

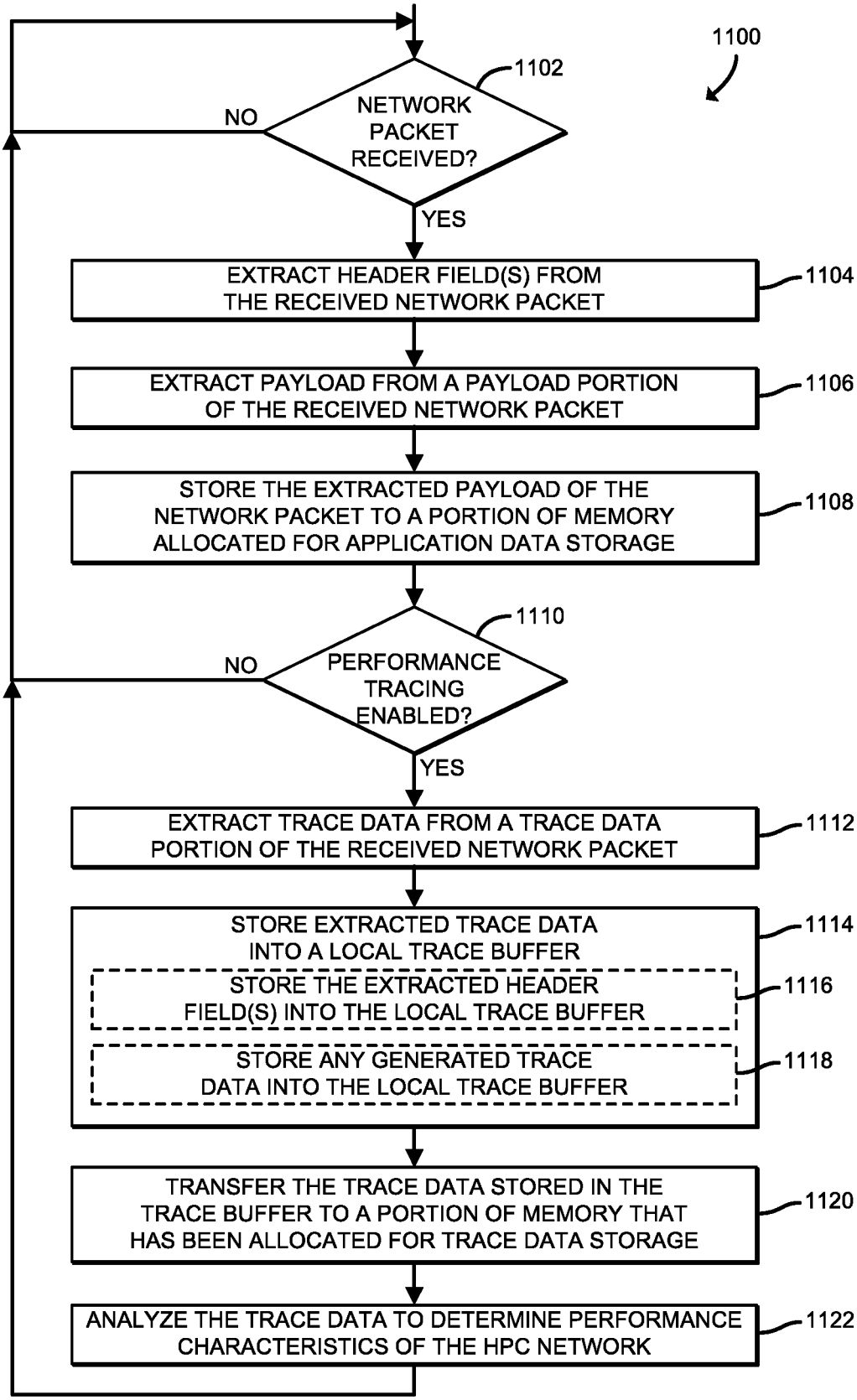
1100
1102
NETWORK PACKET RECEIVED?
NO
YES
EXTRACT HEADER FIELD(S) FROM THE RECEIVED NETWORK PACKET
1104
EXTRACT PAYLOAD FROM A PAYLOAD PORTION OF THE RECEIVED NETWORK PACKET
1106
STORE THE EXTRACTED PAYLOAD OF THE NETWORK PACKET TO A PORTION OF MEMORY ALLOCATED FOR APPLICATION DATA STORAGE
1108
1110
PERFORMANCE TRACING ENABLED?
NO
YES
EXTRACT TRACE DATA FROM A TRACE DATA PORTION OF THE RECEIVED NETWORK PACKET
1112
STORE EXTRACTED TRACE DATA INTO A LOCAL TRACE BUFFER
1114
STORE THE EXTRACTED HEADER FIELD(S) INTO THE LOCAL TRACE BUFFER
1116
STORE ANY GENERATED TRACE DATA INTO THE LOCAL TRACE BUFFER
1118
TRANSFER THE TRACE DATA STORED IN THE TRACE BUFFER TO A PORTION OF MEMORY THAT HAS BEEN ALLOCATED FOR TRACE DATA STORAGE
1120
ANALYZE THE TRACE DATA TO DETERMINE PERFORMANCE CHARACTERISTICS OF THE HPC NETWORK
1122

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/063592

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/26(2006.01)i, H04L 12/801(2013.01)i, H04L 12/861(2013.01)i

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 12/26; G06F 15/173; H04L 12/701; H04L 12/56; H04L 12/801; H04L 12/861

Documentation 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 models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: trace, performance, update, packet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2014-0226658 Al (CELLCO PARTNERSHIP D/B/A VERIZON WIRELESS et al.) 14 August 2014 See paragraphs [0034], [0037] - [0038], [0053H0057], [0059], claims 1, 4 and figures 2, 4A, 5.	1-25
Y	US 2014-0211639 Al (BROADCOM CORPORATION) 31 July 2014 See paragraphs [0035], [0044]- [0045], [0049] and figures 2-3.	1-25
A	US 2014-0126573 Al (BROADCOM CORPORATION) 08 May 2014 See paragraphs [0028]- [0042] and figures 2-3.	1-25
A	US 2008-0159287 Al (RAMESH NAGARAJAN et al.) 03 July 2008 See paragraphs [0024], [0031], [0034]- [0035] and figure 3.	1-25
A	US 7987257 Bl (JOHN W. STEWART et al.) 26 July 2011 See column 11, line 64 - column 13, line 25 and figure 4.	1-25

☐ 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	Date of mailing of the international search report
06 March 2017 (06.03.2017)	**08 March 2017 (08.03.2017)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KIM, Seong Woo
Telephone No. +82-42-481-3348

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/063592

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0226658 Al	14/08/2014	US 9088612 B2	21/07/2015
us 2014-0211639 Al	31/07/2014	**us** 2016-0080240 Al	17/03/2016
		us 9203723 B2	01/12/2015
us 2014-0126573 Al	08/05/2014	**us** 2014-0126396 Al	08/05/2014
		us 9178782 B2	03/11/2015
		us 9286620 B2	15/03/2016
us 2008-0159287 Al	03/07/2008	CN 101569137 A	28/10/2009
		EP 2115942 Al	11/11/2009
		JP 2010-515366 A	06/05/2010
		KR 10-2009-0100377 A	23/09/2009
		WO 2008-085471 Al	17/07/2008
us 7987257 Bl	26/07/2011	**us** 7606887 Bl	20/10/2009