US00RE49806E

(19) **United States**

(12) **Reissued Patent**

Edsall et al.

(10) **Patent Number: US RE49,806 E**

(45) **Date of Reissued Patent: Jan. 16, 2024**

(54) **TIMESTAMPING PACKETS IN A NETWORK**

(71) Applicant: **Cisco Technology, Inc.**, San Jose, CA (US)

(72) Inventors: **Thomas J. Edsall**, Los Gatos, CA (US); **Wei-Jen Huang**, Burlingame, CA (US); **Chih-Tsung Huang**, Burlingame, CA (US); **Yichou Lin**, San Jose, CA (US)

(73) Assignee: **CISCO TECHNOLOGY, INC.**, San Jose, CA (US)

(21) Appl. No.: **16/400,117**

(22) Filed: **May 1, 2019**

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **9,641,409**
Issued: **May 2, 2017**
Appl. No.: **14/701,882**
Filed: **May 1, 2015**

U.S. Applications:

(63) Continuation of application No. 13/708,347, filed on Dec. 7, 2012, now Pat. No. 9,054,967.

(Continued)

(51) **Int. Cl.**

H04J 3/14	(2006.01)
H04J 3/24	(2006.01)
G06F 15/16	(2006.01)
H04L 29/08	(2006.01)
H04L 12/26	(2006.01)
H04B 7/212	(2006.01)

(Continued)

(52) **U.S. Cl.**
CPC ***H04L 43/0852*** (2013.01); ***H04L 43/106*** (2013.01); ***H04L 69/321*** (2013.01)

(58) **Field of Classification Search**
CPC .. H04L 69/321; H04L 43/106; H04L 43/0852
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,246,684 B1	6/2001	Chapman et al.
6,690,646 B1	2/2004	Fichou et al.

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2008/097001 A1	8/2008

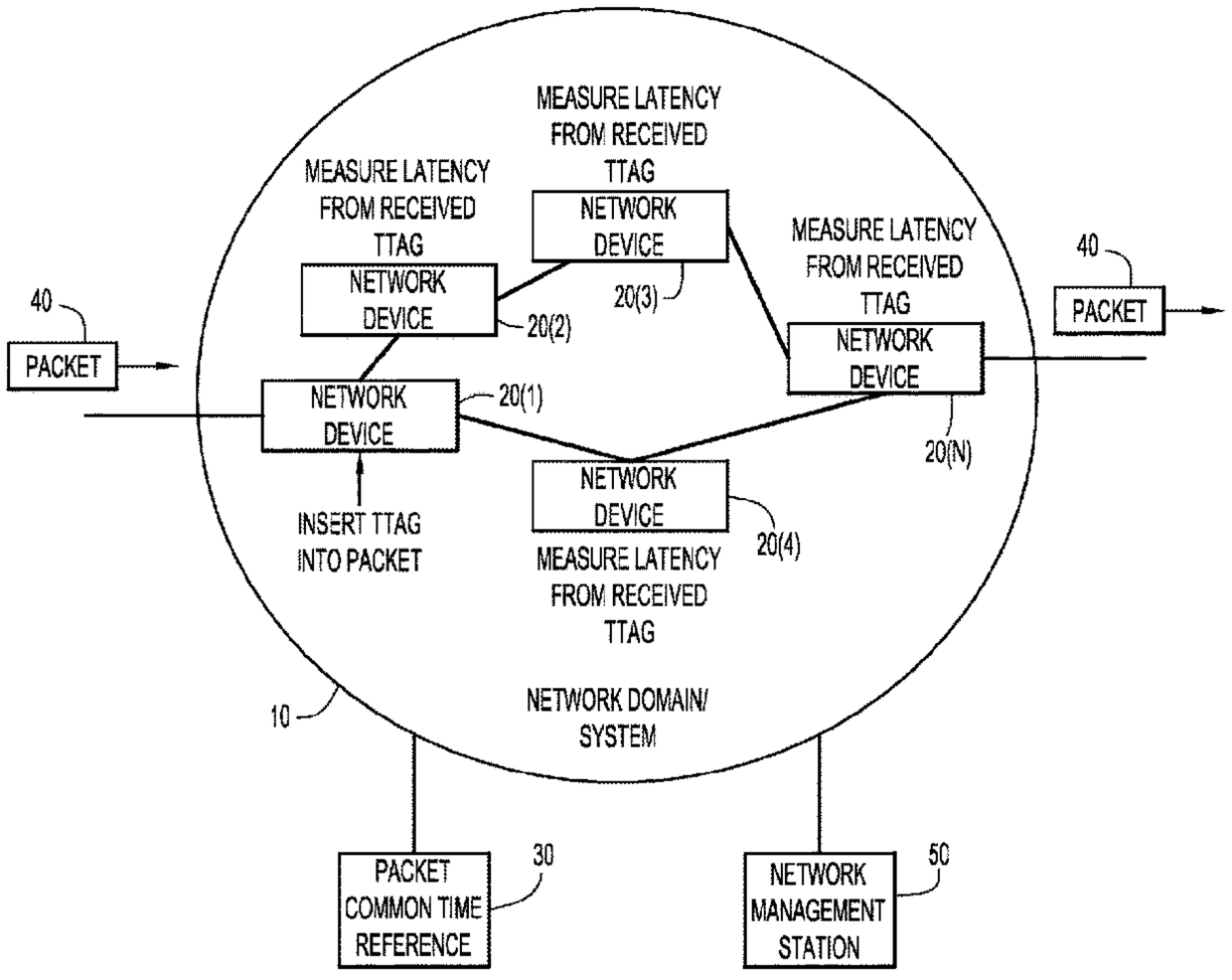
Primary Examiner — Ovidio Escalante

(74) *Attorney, Agent, or Firm* — Edell, Shapiro & Finnan, LLC

(57) **ABSTRACT**

Techniques are presented herein to facilitate latency measurements in a networking environment. A first network device receives a packet for transport within a network domain that comprises a plurality of network devices. The plurality of network devices have a common time reference, that is, they are time synchronized. The first network device generates timestamp information indicating time of arrival of the packet at the first network device. The first network device inserts into the packet a tag that comprises at least a first subfield and a second subfield. The first subfield comprising a type indicator to signify to other network devices in the network domain that the tag includes timestamp information, and the second subfield includes the timestamp information. The first network device sends the packet from to into the network domain to another network device. Other network devices which receive that packet can make latency measurements.

35 Claims, 6 Drawing Sheets



Related U.S. Application Data

(60) Provisional application No. 61/702,323, filed on Sep. 18, 2012.

(51) **Int. Cl.**

H04L 12/28	(2006.01)
H04L 43/0852	(2022.01)
H04L 43/106	(2022.01)
H04L 69/321	(2022.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,853,623	B2	2/2005	Nederveen et al.	
6,892,237	B1	5/2005	Gai et al.	
6,990,202	B2	1/2006	Wee et al.	
7,106,731	B1	9/2006	Lin et al.	
7,395,332	B2	7/2008	Gai et al.	
7,474,666	B2	1/2009	Kloth et al.	
7,539,777	B1 *	5/2009	Aitken	H04L 69/16 370/466
7,656,818	B1	2/2010	Baroudi et al.	
7,792,130	B2	9/2010	Fischer	
7,830,793	B2	11/2010	Gai et al.	
7,899,048	B1	3/2011	Walker et al.	
7,961,621	B2	6/2011	Bergamasco et al.	
7,969,971	B2	6/2011	Gai et al.	
8,116,307	B1	2/2012	Thesayi et al.	
8,166,216	B1 *	4/2012	Kondapalli	G06F 13/385 710/58
8,208,389	B2	6/2012	Alaria et al.	
8,274,905	B2	9/2012	Edwards et al.	
8,605,588	B2	12/2013	Sankaran et al.	
8,640,036	B2	1/2014	Pignataro et al.	
8,681,806	B2	3/2014	Bucknell et al.	
8,718,482	B1 *	5/2014	Roberts	H04L 7/0075 398/161
2003/0231596	A1	12/2003	Hong	
2004/0105392	A1 *	6/2004	Charcranoon	H04L 41/5003 370/252
2006/0007939	A1 *	1/2006	Elangovan	H04L 12/465 370/395.53
2006/0062209	A1	3/2006	Riley	
2006/0253900	A1	11/2006	Paddon et al.	
2006/0268847	A1	11/2006	Halbraich et al.	
2007/0230697	A1 *	10/2007	Wu	H04N 21/242 380/203
2008/0013475	A1 *	1/2008	Bandou	G05B 9/02 370/324
2008/0019282	A1 *	1/2008	Alaria	H04L 43/0864 370/252
2008/0159260	A1 *	7/2008	Vobbilisetty	H04L 69/08 370/351
2008/0279181	A1 *	11/2008	Shake	H04L 45/00 370/389
2009/0034416	A1	2/2009	Baron et al.	
2009/0041011	A1	2/2009	Sheppard	
2009/0100040	A1	4/2009	Sheppard et al.	
2009/0122805	A1 *	5/2009	Epps	H04L 41/5009 370/417
2009/0171474	A1	7/2009	Birze et al.	
2010/0054152	A1	3/2010	Foschiano et al.	
2010/0154033	A1	6/2010	Oulai	
2011/0044173	A1 *	2/2011	Kakadia	H04L 45/02 370/238
2011/0069626	A1 *	3/2011	Sun	H04L 41/5038 370/252
2011/0149998	A1 *	6/2011	Thompson	H04J 3/0697 370/474
2011/0222412	A1 *	9/2011	Kompella	H04L 47/26 370/241.1
2012/0106576	A1 *	5/2012	Hadzic	H04J 3/0697 370/503
2012/0320933	A1 *	12/2012	Magee	H04L 45/70 370/503
2013/0036239	A1 *	2/2013	Spencer	H04L 69/16 709/248

* cited by examiner

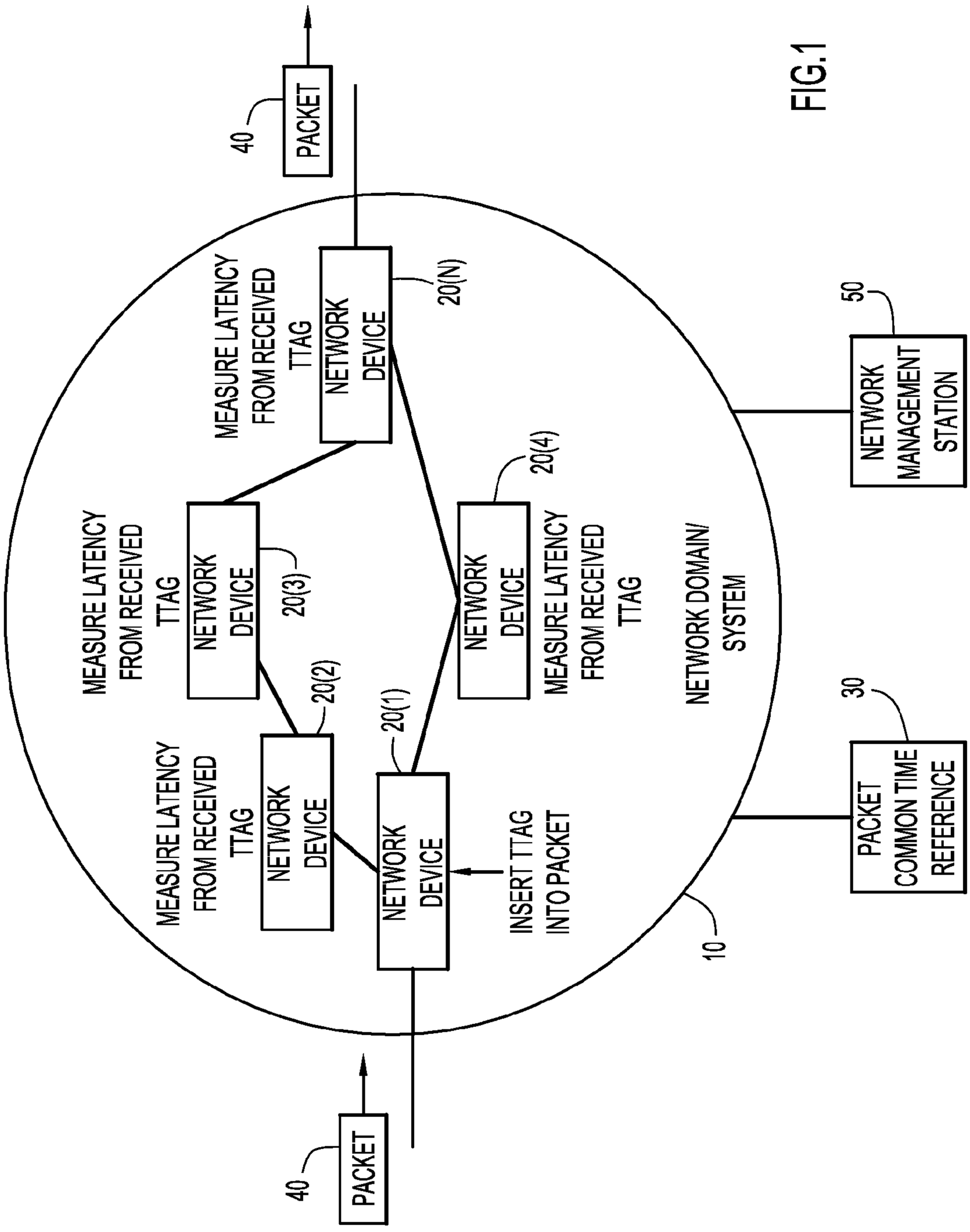

PACKET
40
NETWORK DOMAIN/ SYSTEM
NETWORK DEVICE
20(1)
20(2)
20(3)
20(4)
20(N)
MEASURE LATENCY FROM RECEIVED TTAG
INSERT TTAG INTO PACKET
NETWORK MANAGEMENT STATION
50
PACKET COMMON TIME REFERENCE
30
10

FIG.1

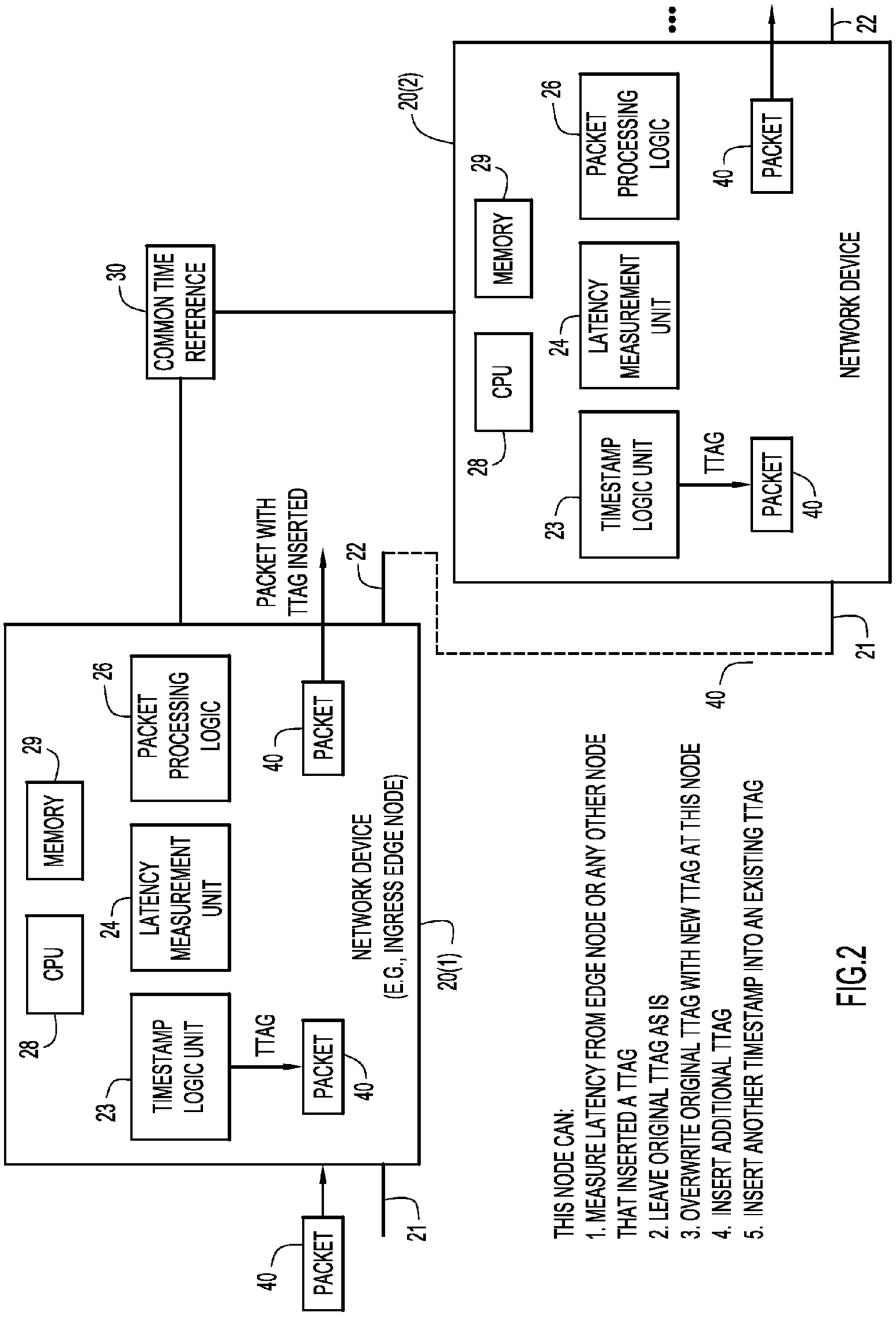

NETWORK DEVICE
(E.G., INGRESS EDGE NODE)
20(1)
20(2)
NETWORK DEVICE
22
21
PACKET WITH TTAG INSERTED
COMMON TIME REFERENCE
30
CPU
28
MEMORY
29
LATENCY MEASUREMENT UNIT
24
PACKET PROCESSING LOGIC
26
TIMESTAMP LOGIC UNIT
23
TTAG
PACKET
40
THIS NODE CAN:
1. MEASURE LATENCY FROM EDGE NODE OR ANY OTHER NODE THAT INSERTED A TTAG
2. LEAVE ORIGINAL TTAG AS IS
3. OVERWRITE ORIGINAL TTAG WITH NEW TTAG AT THIS NODE
4. INSERT ADDITIONAL TTAG
5. INSERT ANOTHER TIMESTAMP INTO AN EXISTING TTAG

FIG.2

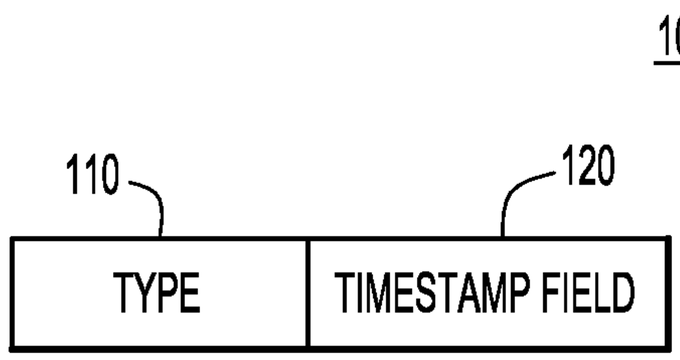

100
110
120
TYPE
TIMESTAMP FIELD

FIG.3A

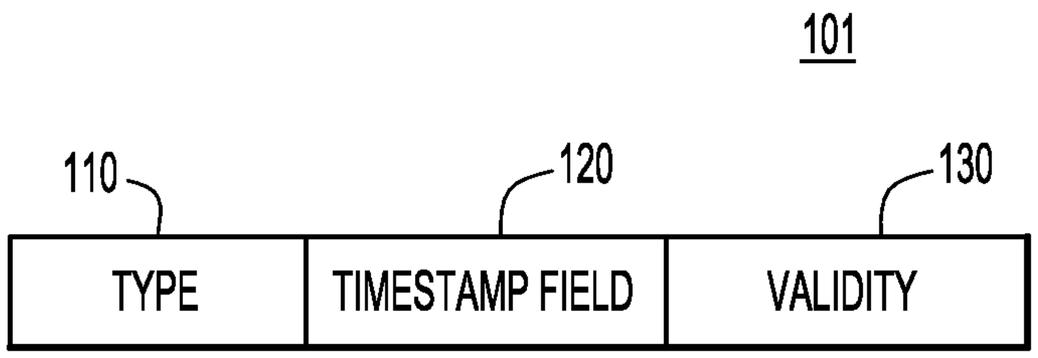

101
110
120
130
TYPE
TIMESTAMP FIELD
VALIDITY

FIG.3B

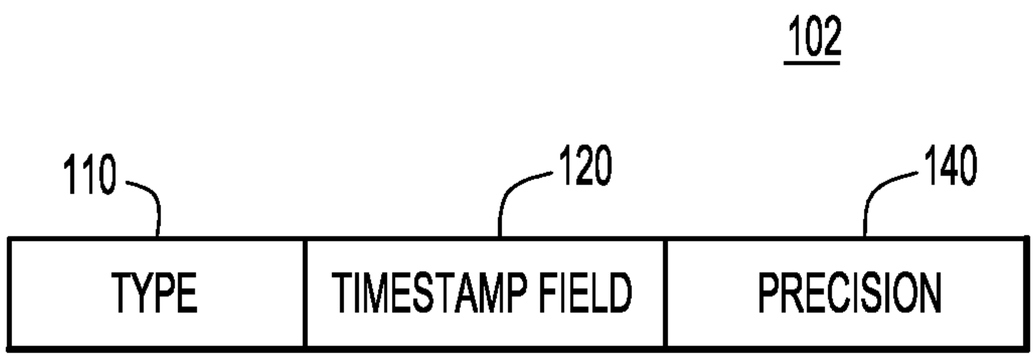

102
110
120
140
TYPE
TIMESTAMP FIELD
PRECISION

FIG.3C

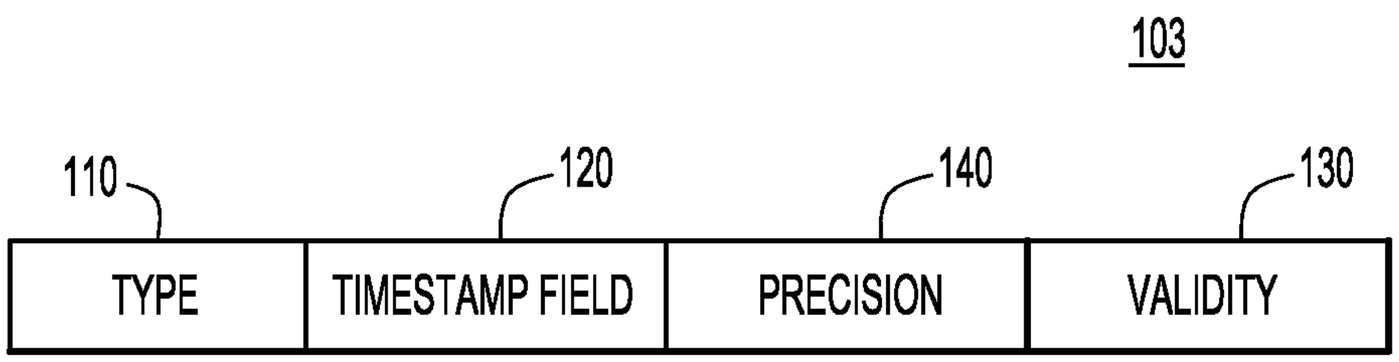

103
110
120
140
130
TYPE
TIMESTAMP FIELD
PRECISION
VALIDITY

FIG.3D

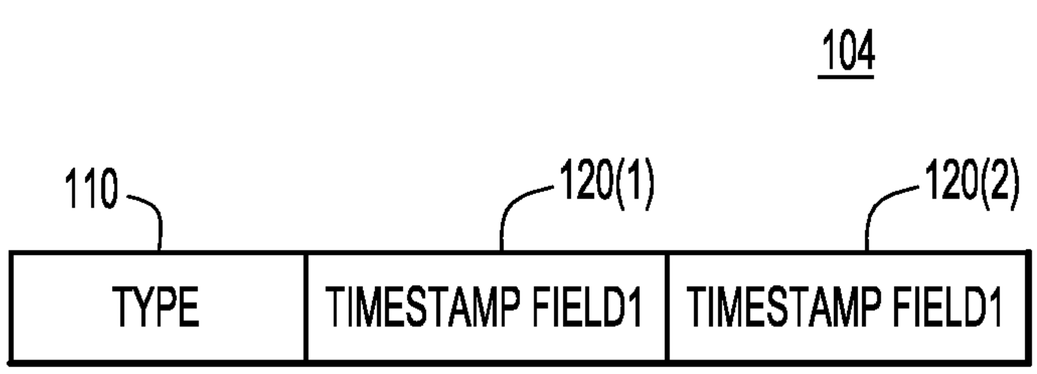

104
110
120(1)
120(2)
TYPE
TIMESTAMP FIELD1
TIMESTAMP FIELD1

FIG.3E

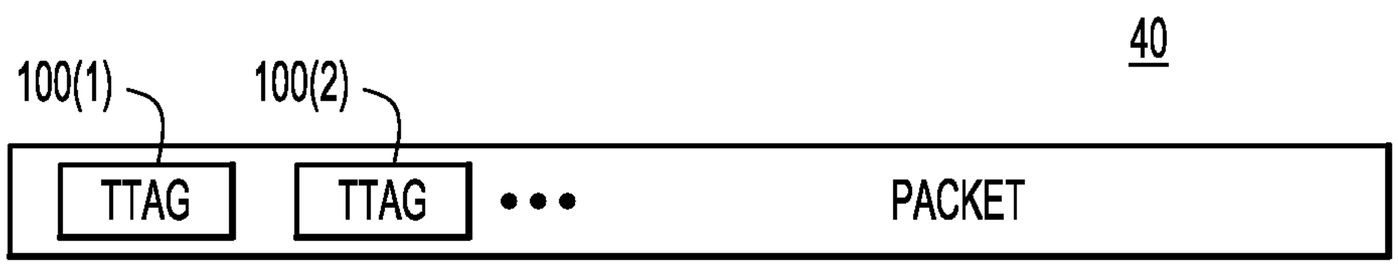

40
100(1)
100(2)
TTAG
TTAG
PACKET

FIG.4

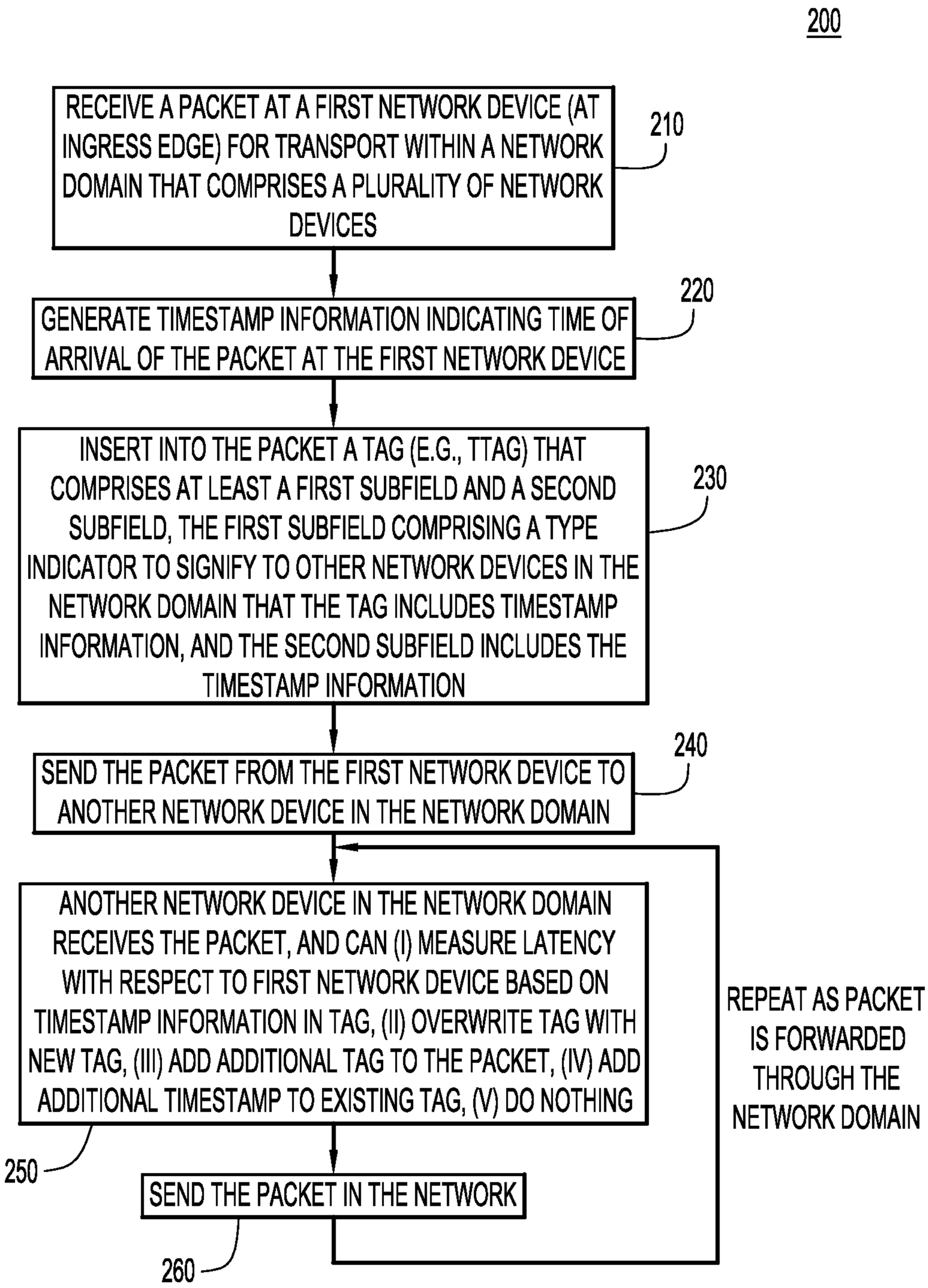

200
RECEIVE A PACKET AT A FIRST NETWORK DEVICE (AT INGRESS EDGE) FOR TRANSPORT WITHIN A NETWORK DOMAIN THAT COMPRISES A PLURALITY OF NETWORK DEVICES
210
GENERATE TIMESTAMP INFORMATION INDICATING TIME OF ARRIVAL OF THE PACKET AT THE FIRST NETWORK DEVICE
220
INSERT INTO THE PACKET A TAG (E.G., TTAG) THAT COMPRISES AT LEAST A FIRST SUBFIELD AND A SECOND SUBFIELD, THE FIRST SUBFIELD COMPRISING A TYPE INDICATOR TO SIGNIFY TO OTHER NETWORK DEVICES IN THE NETWORK DOMAIN THAT THE TAG INCLUDES TIMESTAMP INFORMATION, AND THE SECOND SUBFIELD INCLUDES THE TIMESTAMP INFORMATION
230
SEND THE PACKET FROM THE FIRST NETWORK DEVICE TO ANOTHER NETWORK DEVICE IN THE NETWORK DOMAIN
240
ANOTHER NETWORK DEVICE IN THE NETWORK DOMAIN RECEIVES THE PACKET, AND CAN (I) MEASURE LATENCY WITH RESPECT TO FIRST NETWORK DEVICE BASED ON TIMESTAMP INFORMATION IN TAG, (II) OVERWRITE TAG WITH NEW TAG, (III) ADD ADDITIONAL TAG TO THE PACKET, (IV) ADD ADDITIONAL TIMESTAMP TO EXISTING TAG, (V) DO NOTHING
250
SEND THE PACKET IN THE NETWORK
260
REPEAT AS PACKET IS FORWARDED THROUGH THE NETWORK DOMAIN

FIG.5

TIMESTAMPING PACKETS IN A NETWORK

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue; a claim printed with strikethrough indicates that the claim was canceled, disclaimed, or held invalid by a prior post-patent action or proceeding.

RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 13/708,347, filed Dec. 7, 2012, entitled "Timestamping Packets in a Network," which claims priority to U.S. Provisional Patent Application No. 61/702,323, filed Sep. 18, 2012, also entitled "Timestamping Packets in a Network." Both applications are incorporated herein by reference in their entirety.

TECHNICAL FIELD

The present disclosure relates to networking systems and devices.

BACKGROUND

In a computer network, such as a data center, data is transmitted from a source to a destination in the form of packets that generally pass through one or more networking devices (e.g., switches, routers, firewalls, etc.). During the transmission, packets are generally temporarily stored in one or more network buffers of the networking devices.

Certain data center customers demand network architectures that can provide low latency, high bandwidth, and often massive scalability. However, measuring latency may be difficult and time sensitive applications often do not have the proper visibility into how it has taken for packets to reach a certain destination and when packets were actually sourced at specific locations in the network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates an example network block diagram in which the timestamping of packets is used to track latency at different points in a network domain.

FIG. **2** illustrates in more detail components in each network device in the network domain that enable the timestamping and latency measurements of packets that traverse the network domain.

FIGS. **3**A-**3**E illustrate examples of a timestamp tag (TTAG) that may be inserted into a packet by any of the network devices in the network domain, and which is used for latency measurements.

FIG. **4** illustrates an example of a packet having one or more TTAGs.

FIG. **5** is a flow chart that depicts operations performed by network devices that insert and operate on TTAGs.

DESCRIPTION OF EXAMPLE EMBODIMENTS

Overview

Techniques are presented herein to facilitate latency measurements in a networking environment. A first network device receives a packet for transport within a network domain that comprises a plurality of network devices. The plurality of network devices have a common time reference, that is, they are time synchronized. The first network device generates timestamp information indicating time of arrival of the packet at the first network device. The first network device inserts into the packet a tag that comprises at least a first subfield and a second subfield. The first subfield comprises a type indicator to signify to other network devices in the network domain that the tag includes timestamp information, and the second subfield includes the timestamp information. The first network device sends the packet into the network domain to another network device. Other network devices in the network domain which receive that packet can then make latency measurements, insert another tag, overwrite the tag, and perform various other operations described herein.

Example Embodiments

Reference is first made to FIG. **1**. FIG. **1** shows a network domain or system **10** comprising a plurality of network devices **20**(**1**)-**20**(N) that are all synchronized to a common time reference **30**. That is, all of the network devices **20**(**1**)-**20**(N) of interest in the network domain **10** have the same global time reference, determined by IEEE 1588 Precision Time Protocol (PTP) or other methods now known or hereinafter developed. For simplicity, the term "node" is also used herein synonymously with the term "network device".

A packet **40** enters the network domain **10** at some edge node, e.g., network device **20**(**1**) in the example of FIG. **1**, and departs the network domain **10** at another edge node, e.g., network device **20**(N). Presented herein are techniques to determine latency at any point in the network domain for a packet as it traverses through the network domain **10**. One particular latency measure that is of interest is the end-to-end latency, that is, the elapsed time (latency) for a packet to travel between an ingress port of edge network device **20**(**1**) to an egress port of edge network device **20**(N) of network domain **10**.

A timestamp tag (TTAG) is inserted into a packet **40** by the edge network device **20**(**1**) of the network domain **10**. The TTAG includes timestamp information indicating time of arrival at network device **20**(**1**). All of the network devices in the network domain **10** that receive the packet **40** (with the inserted TTAG) can perform measurements based on the timestamp information contained in TTAG inserted into packet **40**, and perform other operations, including adding another TTAG, overwriting an existing TTAG, adding another timestamp value into an existing TTAG, etc., as will described in more detail hereinafter. As indicated in FIG. **1**, any network device in the network domain **10** can measure and report the latency based on the TTAG contained in a packet. However, not all network devices must understand a TTAG. In cases in which a network device does not understand a TTAG contained in a packet, the TTAG can be skipped as part of packet processing or in the case of Ethernet packets, some switches will process the packet up to the TTAG and skip the rest of the packet.

The network devices **20**(**1**)-**20**(N) shown in FIG. **1** can be any network device now known or hereinafter developed, including a switch, router, gateway, a software stack on a host device, virtual network interface cards (VNICs) virtual switches, physical network interface cards (including those that support virtualization).

FIG. **1** further shows a network management station **50** that may take a variety of forms, e.g., server computer, virtualized server, etc., that communicates with each net-

work device **20**(**1**)-**20**(N) for purposes of configuring the network devices to insert TTAGs, make latency measurements, report latency measurements, and to receive latency measurements from the network devices **20**(**1**)-**20**(N).

Turning now to FIG. **2**, a more detailed description is provided for the components of a network device that are configured to perform the TTAG insertion and latency measurement operations presented herein. FIG. **2** shows a simplified diagram of two network devices **20**(**1**) and **20**(**2**), though it should be understood that each network device **20**(**1**)-**20**(N) in a network domain that is to participate in the techniques presented herein is configured in a similar manner as that shown for network devices **20**(**1**) and **20**(**2**) in FIG. **2**. Specifically, each network device **20**(**1**)-**20**(N) includes multiple ports, and for simplicity an ingress port **21** and egress **22** are shown in FIG. **2**. Furthermore, each network device includes a timestamp logic unit **23**, a latency measurement unit **24**, packet processing logic **26**, a central processing unit (CPU) **28** and memory **29**. The packet processing logic **26** is representative of the conventional packet processing components in a network device, such as buffers, switch tables, switch fabric, queues, etc., that operate to determine whether to drop, forward (and via a particular egress port), switch, etc., a particular packet based on the contents of the header of the packet. The details of the packet processing logic **26** are not described herein because they are well known in the art, and do not pertain to the timestamping techniques presented herein.

The timestamp logic unit **23** generates a timestamp upon arrival of the packet at an ingress port **21** of the network device. The timestamp is with respect to the common time reference **30** used by all network devices in the network domain. The timestamp logic unit **23** may insert the TTAG into a packet **40** immediately upon arrival at the ingress port, and then forward the packet to be processed by the packet processing logic **26**, insert the TTAG in parallel with the processing of the packet by the packet processing logic **26**, or after processing of the packet by the packet processing logic **26**. Examples of various formats of a TTAG are presented hereinafter in connection with FIGS. **3**A-**3**D. The TTAG is inserted in any manner that does not interfere with the normal processing of the packet by the network devices.

The timestamp logic unit **23** may also be configured to insert additional information into a TTAG, including one or more bits to indicate a validity of the timestamp value, one or more bits to indicate a timing precision of the timestamp value. In general, precision is system or network domain wide and is pre-negotiated among the network devices with respect to the common time reference **30**. When a new timestamp value is to be inserted into a packet, a network device uses either ingress port timestamp from the common time reference **30** (synchronized clock) or an invalid value of zero. Invalid values are preserved across the network domain, as described further hereinafter.

Since any device can serve as an edge node in a network domain, each network device includes latency measurement **24** which is configured to perform a latency computation (current time minus the timestamp value contained in a TTAG of a received packet). For example, the latency measurement unit **24** in network device **20**(**2**) may compute the latency associated with packet **40** using the timestamp value contained in the TTAG inserted by edge network device **20**(**1**).

The CPU **28** may perform higher level latency analysis and reporting operations based on software instructions contained in memory **29**. The memory **29** may also serve for additional storage of latency measurements. The CPU **28** may send latency measurements to a local or remotely located computing device that is used by a network administrator to monitor performance of network domain **20**. Moreover, the CPU **28** in any given network device may receive commands or instructions from a network management station (FIG. **1**) to control the TTAG-related operations in a network device, latency measurements made by a network device, etc.

Memory **29** may comprise read only memory (ROM), random access memory (RAM), magnetic disk storage media devices, optical storage media devices, flash memory devices, electrical, optical, or other physical/tangible memory storage devices. The CPU is, for example, a microprocessor or microcontroller. Thus, in general, the memory **29** may comprise one or more tangible (non-transitory) computer readable storage media (e.g., a memory device) encoded with software comprising computer executable instructions and when the software is executed (by the CPU **28**) it is operable to perform the operations described herein.

The timestamp logic unit **23** and latency measurement unit **24** may be embodied by digital logic gates configured to perform the operations described herein, or in another form, by software stored in memory **29** and executed by CPU **28** to perform the operations described herein. In another example, the timestamp logic unit **23** and latency measurement unit **24** may be integrated or embedded with the packet processing logic **26**, which itself may be embodied by one or more application specific integrated circuits (ASICs).

As shown in FIG. **2**, when a network device receives from another network device a packet that includes a TTAG, there are several options for operations that may be performed. First, the network device can measure latency from the edge node or any other node that inserted a TTAG in the packet. Second, the network device can do nothing, leave the TTAG as is and process the packet in the ordinary course. Third, the network device can overwrite an existing TTAG in the packet with a new TTAG (and timestamp of arrival) at this network device. Fourth, the network device can insert an additional TTAG into the packet. For example, multiple TTAGs can be inserted such as through tunnels or if negotiated across ports. Fifth, the network device can insert an additional timestamp value (based on time of arrival at this network device) into an existing TTAG of the packet. The CPU **28** in one or more network devices may be configured, through software stored in memory **29**, to insert additional TTAGs into a packet, overwrite an existing TTAG or insert another timestamp value in a TTAG as described further hereinafter. In any case, the network device processes the packet as it normally would if the TTAG were not present. As shown in FIG. **2**, network device **20**(**2**) sends packet **40** on in the network domain with any existing TTAGs, a newly overwritten TTAG, etc., under control of the CPU **28**.

Reference is now made to FIGS. **3**A-**3**E for examples of various formats of TTAGs. FIG. **3**A illustrates a first basic form of a TTAG **100**, including a first Type subfield **110** and a second Timestamp subfield **120**. The Type subfield **110** is used to identify the "type" as a TTAG which allows any network device to recognize the TTAG. The independence of timestamp information contained in a TTAG from any other existing format liberates current network devices or CPUs to determine system-wide time. In one example, the Type subfield **110** is 8 bytes such as that specified by an Ethertype subfield in an Ethernet frame. The Timestamp subfield **120** is a 48 bit number having a format of an unsigned rolling 48 bit binary number value, e.g., having 100 picosecond resolution. When clock time increments to

zeros for all 48 LSBs, the Timestamp subfield uses a value of one instead. A value of one repeats unlike one's complement. Thus, a lower 48 bit clock time of 0 and 1 both map to Timestamp subfield value of 1.

FIG. **3**B shows an example format of a TTAG **101** with an explicit validity bit shown at **130**. The validity bit **130** is configured so that if it takes on a first value, e.g., logic "1", the timestamp value in Timestamp subfield **120** is valid, and if the validity bit takes on a second value, e.g., logic "0", the timestamp value in the Timestamp subfield **120** is invalid. Invalid timestamp values are preserved across the network domain by other network devices that receive a packet with a TTAG indicated to contain an invalid timestamp value.

There is another way to signify an invalid timestamp value in a packet without using the explicit validity bit **130**. A Timestamp subfield value of zero represents an invalid timestamp. Thus, when the value contained in Timestamp subfield **130** is all zeros, a network device construes this as indicating that the timestamp contained in the TTAG is invalid. The subfield can be compatible with timestamp always valid in the network when invalid capability is disabled in the network domain. Thus, a predetermined bit pattern (e.g., all zeros) in the Timestamp subfield **130** indicates that the timestamp information of the Timestamp subfield is not valid.

FIG. **3**C illustrates another format of a TTAG shown at reference numeral **102**. In this example, there is an additional precision subfield **140** that contains a bit pattern configured to indicate precision of the timestamp value contained in the Timestamp subfield **120**. The concept of network-wide pre-negotiated precision was described above.

FIG. **3**D illustrates still another format of a TTAG shown at reference numeral **103**. This example shows that there is both the explicit validity bit **130** and precision subfield **140**.

FIG. **3**E illustrates yet another format of a TTAG shown at reference numeral **104**. TTAG **104** includes multiple Timestamp fields **120**(**1**), **120**(**2**), etc. Each Timestamp subfield can contain a different timestamp value inserted by the same network device or by different network devices.

Turning to FIG. **4**, a general diagram is shown of a packet **40** having one or more TTAGs **100**(**1**), **100**(**2**), etc., therein. In the simple case, a packet will have only one TTAG at any given time. However, there is utility in the capability of multiple TTAGs in a packet. For example, multiple TTAGs can be inserted in situations when packets are encapsulated in tunnels or if negotiated across ports of network devices. Furthermore, each packet that has a TTAG inserted into it does not affect the networking operations performed by any network device that receives the packet. Any network device can obtain information from the TTAGs contained in packets and thereby obtain visibility to latency within the network.

In some implementations of the techniques described herein, the number of TTAGs that can be inserted into a packet is limited in number to, for example, six (6) or some number between one (1) and ten (10). In other implementations, the number of TTAGs that can be inserted into a packet is unlimited, in which any device that receives the packet within the network for passing the packet to a destination from a source can insert a TTAG into the packet. In some implementations, when the maximum number of TTAGs that can be inserted into a packet is reached, downstream network devices cannot insert TTAGs into the packet. In yet other implementations, when the maximum number of TTAGs that can be inserted into a packet is reached, downstream network devices are allowed to overwrite TTAGs on a first-in, first-out basis.

There are numerous possibilities for locating the TTAG information in the packet. The TTAG can be inserted within a Layer 2 portion of the packet. This is in contrast to conventional approaches that perform application-specific packet time measurements at Layer 3. For example, one conventional packet time measurement approach collects runtime measurement of packets based on an application-specific determination of packet arrivals at Layer 3, as opposed to incorporating timestamp tag information directly into all packets at Layer 2 as accomplished using the techniques presented herein.

In some implementations, such as for Internet Protocol Version 4 (IPv4) or IPv6 packets, the TTAG can be provided immediately after the virtual local area network (VLAN) subfield and immediately before the IPv4 or IPv6 field in the packet header portion of the packet, in which the TTAG is meshed in the protocol stack within the header portion of the packet. Other locations for insertions of the TTAG within a packet may be envisioned while remaining within the spirit and scope of the techniques presented herein.

Turning now to FIG. **5**, a flow chart is presented that illustrates an operational flow **200** with respect to network devices that insert and interpret TTAGs in packets as the packets traverse through a network domain. At **210**, a packet at a first network device (e.g., ingress edge node for the packet) of a network domain is received. The packet is for transport through the network domain, and the network domain includes a plurality of network devices, e.g., as depicted in FIG. **1**. At **220**, the first network device generates timestamp information indicating time of arrival of the packet at the first network device. As explained above in connection with FIG. **1**, the timestamp is generated with respect to a time reference that is common across all of the network devices in the network domain. At **230**, the first network device inserts into the packet a tag that comprises at least a first subfield and a second subfield. The first subfield comprising a type indicator to signify to other network devices in the network domain that the tag includes timestamp information, and the second subfield includes or contains the timestamp information.

At **240**, the first network device sends the packet to another network device in the network domain, using the normal packet processing functions for the packet. At **250**, another network device in the network domain receives the packet, and can perform any one or more of: (i) measuring latency with respect to first network device based on timestamp information in tag, (ii) overwriting tag with new tag, (iii) adding an additional tag to the packet, and (iv) adding an additional timestamp to an existing tag, or (v) doing nothing and processing the packet in the normal course without performing any of operations (i)-)(iv).

At **250**, the network device sends the packet on in the network in the ordinary course of packet processing. Operations **240** and **250** are repeated at subsequent network devices in the network domain as the packet travels through the network domain.

As explained above in connection with FIGS. **1** and **2**, a network management station may receive reports as to latency values measured by network devices in the network domain. The network management station may also configure the various network devices to perform more specialized tagging of packets, depending on certain applications supported in the network, tunnels supported in the network, etc. The latency measurements made by network devices at the edge of the network domain and at various points in between allow a network administrator to understand how the net-

work domain is handling traffic and whether there are network congestion issues within a particular portion of the network domain.

Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

The above description is intended by way of example only.

What is claimed is:

1. A method comprising:

receiving a *network traffic* packet at a first network device for transport within a network domain that comprises a plurality of network devices, *the network traffic packet comprising a header with a Layer 2 portion and a Layer 3 portion,* wherein the plurality of network devices have a common time reference *and the network traffic packet does not include time information sourced from the common time reference*, and wherein the first network device is at an ingress edge of the network domain for the *network traffic* packet, wherein the *network traffic* packet is received at the *first* network device from outside of the network domain;

generating, by the first network device, timestamp information indicating time of arrival of the *network traffic* packet within the network domain;

inserting into the *Layer 2 portion of the header of the network traffic* packet a *first* tag that *is distinct from contents of the network traffic packet, wherein the first tag* comprises at least a first *type* subfield and **[**a**]** *multiple* second **[**subfield**]** *subfields*, *wherein* the first *type* subfield comprising **[**a type**]** *an* indicator to signify to other network devices in the network domain that the *first* tag includes timestamp information data, and *wherein at least one of* the *multiple* second **[**subfield**]** *subfields* includes **[**the**]** timestamp information *indicating time of arrival of the network traffic packet with an associated precision*, wherein the timestamp information is configured to be used by each device in the network domain that receives the *network traffic* packet as an indication of when the *network traffic* packet entered the network domain; and

sending the *network traffic* packet from the first network device to another network device in the network domain.

2. The method of claim **1**, further comprising:

receiving at a second network device the *network traffic* packet sent from the first network device;

determining a time of arrival of the *network traffic* packet at the second network device; and

measuring latency of the *network traffic* packet within the network domain based on the time of arrival of the *network traffic* packet and the timestamp information contained in the *first* tag of the *network traffic* packet.

3. The method of claim **1**, further comprising:

at each of the other network devices in the network domain:

receiving the *network traffic* packet sent by the first network device;

determining time of arrival of the *network traffic* packet; and

measuring latency of the *network traffic* packet within the network domain based on the time of arrival of the *network traffic* packet and the timestamp information contained in the *first* tag of the *network traffic* packet.

4. The method of claim **1**, wherein the first *type* subfield of the *first* tag is an Ethertype subfield.

5. The method of claim **1**, further comprising inserting in the *first* tag a validity bit that indicates whether or not the timestamp information is valid.

6. The method of claim **1**, wherein inserting into the *network traffic* packet comprises inserting a predetermined bit pattern in the second subfield to indicate that the timestamp information of the second subfield is not valid.

7. The method of claim **1**, further comprising:

receiving at a second network device the *network traffic* packet sent from the first network device; and

inserting **[**an additional**]** *a second* tag into the *network traffic* packet, the **[**additional**]** *second* tag including timestamp information representing time of arrival of the *network traffic* packet at the second network device, *wherein the second tag is distinct from contents of the network traffic packet and distinct from the first tag*.

8. A method comprising:

receiving a *network traffic* packet at an egress port of a first network device, the *network traffic* packet *not including time information sourced from a common time reference and* having been transported within a network domain that comprises a plurality of network devices, *the network traffic packet comprising a header with a Layer 2 portion and a Layer 3 portion,* wherein the plurality of network devices have a common time reference, and wherein the first network device is at an egress edge of the network domain for the *network traffic* packet;

determining a time of arrival of the *network traffic* packet at the first network device;

extracting, by the first network device, *from the Layer 2 portion of the header of the network traffic packet a tag that is distinct from contents of the network traffic packet, the tag including a type subfield and multiple timestamp subfields, wherein the type subfield comprises an indicator to signify to other network devices in the network domain that the tag includes timestamp information, wherein at least one of the multiple timestamp subfields including* timestamp information indicating time of arrival of the *network traffic* packet within the network domain *with an associated precision*, wherein the *network traffic* packet was received within the network domain at a second network device, wherein the second network device is arranged at an ingress edge of the network domain for the *network traffic* packet; and

determining **[**the**]** *a* latency of the *network traffic* packet within the network domain by calculating an elapsed time between the timestamp information and the arrival of the *network traffic* packet at the egress port of the first network device.

9. The method of claim **8**, wherein extracting the timestamp information comprises extracting at least a first subfield and **[**a**]** *multiple* second **[**subfield**]** *subfields*, the first subfield *being the type subfield* comprising **[**a type**]** *the* indicator signifying that the *multiple* second **[**subfield includes**]** *subfields include* the timestamp information.

10. The method of claim **9**, wherein extracting the first subfield comprises extracting an Ethertype subfield.

11. The method of claim **9**, wherein extracting the timestamp information comprises extracting a predetermined bit

pattern in the second subfield to indicate that the timestamp information of the second subfield is not valid.

12. The method of claim **8**, wherein extracting the timestamp information comprises extracting timestamp information for each of the plurality of network devices that the *network traffic* packet traversed travelling from the second network device to the first network device.

13. The method of claim **8**, wherein extracting the timestamp information comprises extracting a validity bit that indicates whether or not the timestamp information is valid.

[14. The method of claim **8**, wherein extracting the timestamp information comprises extracting the timestamp information from an Open Systems Interconnection model Layer 2 portion of the packet.**]**

15. An apparatus comprising:

a plurality of ports each configured to receive and send *network traffic* packets in a network domain, *the network traffic packets comprising a header with a Layer 2 portion and a Layer 3 portion,* wherein at least one of the plurality of ports is an egress port at an egress edge of the network domain*, and wherein the network traffic packets do not include time information sourced from a common time reference*;

a timestamp logic unit configured to extract*, from the Layer 2 portion of the header of the network traffic packet a tag that is distinct from contents of the network traffic packet, the tag including a type subfield and multiple timestamp subfields, wherein the type subfield comprises an indicator to signify to other network devices in the network domain that the tag includes timestamp information, wherein at least one of the multiple timestamp subfields including* timestamp information indicating time of arrival of **[**the**]** *a network traffic* packet at the network domain *with an associated precision*, wherein the timestamp information is generated at a network device arranged at an ingress edge of the network domain for the *network traffic* packet; and

a latency measurement unit configured to determine a time of arrival of the *network traffic* packet at the egress port and determine an elapsed time between the arrival of the *network traffic* packet at the network domain and the arrival of the *network traffic* packet at the egress port.

16. The apparatus of claim **15**, wherein the timestamp logic unit is configured to extract at least a first subfield and **[**a**]** *multiple* second **[**subfield**]** *subfields*, the first subfield *being the type subfield* comprising **[**a type**]** *the* indicator signifying that the *multiple* second **[**subfield includes**]** *subfields include* the timestamp information.

17. The apparatus of claim **16**, wherein the timestamp logic unit is configured to extract the first subfield as an Ethertype subfield.

18. The apparatus of claim **16**, wherein the timestamp logic unit is configured to extract the timestamp information by extracting a predetermined bit pattern in the second subfield to indicate that the timestamp information of the second subfield is not valid.

19. The apparatus of claim **15**, wherein the latency measurement unit is configured to determine the elapsed time from the time of arrival of the *network traffic* packet at the egress port and the timestamp information.

[20. The apparatus of claim **15**, wherein the timestamp logic unit is configured to timestamp information from an Open Systems Interconnection model Layer 2 portion of the packet.**]**

21. The apparatus of claim **15**, wherein the timestamp logic unit is configured to extract the timestamp information by extracting a validity bit that indicates whether or not the timestamp information is valid.

[22. The apparatus of claim **15**, wherein the timestamp logic unit is configured to extract the timestamp information by extracting the timestamp information from an Open Systems Interconnection model Layer 2 portion of the packet.**]**

23. The method of claim **1**, wherein generating the timestamp information indicating time of arrival of the *network traffic* packet within the network domain comprises determining a time of arrival at the first network device.

[24. The method of claim **1**, wherein inserting into the packet the tag comprises inserting the tag in an Open Systems Interconnection model Layer 2 portion of the packet.**]**

25. A method, comprising:

capturing, at a first network device that is configured to forward network traffic packets, a network traffic packet for transport within a network domain that comprises a plurality of network devices, the network traffic packet comprising a header with a Layer 2 portion and a Layer 3 portion, wherein the first network device is at an ingress edge of the network domain for the network traffic packet, and wherein the network traffic packet does not include time information sourced from a common time reference and is transported to the first network device from outside of the network domain;

generating, by the first network device, timestamp information indicating time of capture of the network traffic packet at the first network device;

including in the Layer 2 portion of the header of the network traffic packet a tag that is distinct from contents of the network traffic packet, wherein the tag comprises at least a first subfield and multiple second subfields, wherein the first subfield comprising a type subfield comprises an indicator to signify to other network devices in the network domain that the tag includes timestamp information, and wherein at least one of the multiple second subfields comprises a timestamp value that indicates time of capture of the network traffic packet at the first network device with an associated precision; and

sending the network traffic packet from the first network device to a second network device.

26. The method of claim 25, wherein the tag further includes a subfield indicating a validity of the network traffic packet.

27. The method of claim 25, wherein the network traffic packet containing the tag is formatted such that the network traffic packet can be processed by a second network device that is not configured to understand the tag.

28. The method of claim 25, further comprising:

capturing the network traffic packet at the second network device;

determining time of capture of the network traffic packet at the second network device; and

measuring latency of the network traffic packet within the network domain based on the time of capture of the network traffic packet at the second network device and the timestamp information included in the tag of the network traffic packet.

29. The method of claim 25, wherein the tag further includes a precision subfield indicating precision of the timestamp value in an associated second subfield.

30. An apparatus comprising:

an ingress port configured to receive a network traffic packet, the network traffic packet comprising a header with a Layer 2 portion and a Layer 3 portion, wherein the network traffic packet does not include time information sourced from a common time reference; and

a timestamp logic unit configured to generate timestamp information based on time of arrival of the network traffic packet at the ingress port and insert into the Layer 2 portion of the header of the network traffic packet, a tag that is distinct from contents of the network traffic packet, wherein the tag comprises subfields including a type subfield and multiple timestamp subfields, wherein the type subfield comprises an indicator to signify to other network devices that the tag includes timestamp information, and wherein at least one of the multiple timestamp subfields having an associated precision subfield to indicate a precision for a timestamp value for time of arrival of the network traffic packet in an associated timestamp subfield.

31. The apparatus of claim 30, wherein the network traffic packet further comprises a validity check subfield.

32. The apparatus of claim 30, further comprising:

a latency measurement unit configured to determine latency of the network traffic packet traversing a network domain from a first ingress port of the network domain to a second port based on at least one of the subfields containing timestamp information.

33. The apparatus of claim 30, wherein the type subfield comprises an Ethertype subfield.

34. An apparatus comprising:

a plurality of ports, each port configured to receive and send network traffic packets in a network domain, the network traffic packets comprising a header with a Layer 2 portion and a Layer 3 portion, wherein at least one of the plurality of ports is an ingress port at an edge of the network domain, wherein the network traffic packets do not include time information sourced from a common time reference;

a timestamp logic unit configured to insert into the Layer 2 portion of a received network traffic packet a first timestamp information indicating time of arrival of the received network traffic packet at the network domain from outside the network domain, wherein the first timestamp information is generated with reference to the ingress port at the edge of the network domain to enable latency measurements based on the first timestamp information, and the first timestamp information is distinct from contents of the received network traffic packet, the first timestamp information having an associated precision; and

wherein the Layer 2 portion of the received network traffic packet further includes a type subfield that comprises an indicator to signify to other network devices received network traffic packet includes timestamp information.

35. The apparatus of claim 34, wherein the type subfield is an Ethertype subfield.

36. A system comprising the apparatus of claim 34, and further comprising another apparatus that receives the network traffic packet that includes the first timestamp information.

37. The method of claim 1, wherein inserting comprises inserting a plurality of additional tags into the network traffic packet, each additional tag including different timestamp information, wherein each additional tag is distinct from the first tag and distinct from other additional tags.

38. The method of claim 8, wherein the network traffic packet includes multiple tags, each tag of the multiple tags including timestamp information indicating time of arrival of the network traffic packet at a different network device, and wherein each tag of the multiple tags is distinct from other tags.

39. The method of claim 38, wherein one or more tags of the multiple tags have been overwritten by a network device when the network traffic packet has a maximum number of tags.

* * * * *