US 20170257314A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2017/0257314 A1**

Wood et al. (43) **Pub. Date: Sep. 7, 2017**

(54) PROTOCOL TO QUERY FOR HISTORICAL NETWORK INFORMATION IN A CONTENT CENTRIC NETWORK

(71) Applicant: **CISCO TECHNOLOGY, INC.**, San Jose, CA (US)

(72) Inventors: **Christopher A. Wood**, San Francisco, CA (US); **Ignacio Solis**, Scotts Valley, CA (US); **Andriana Ioannou**, Thessaloniki (GR); **Glenn C. Scott**, Portola Valley, CA (US)

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

(21) Appl. No.: **15/061,979**

(22) Filed: **Mar. 4, 2016**

Publication Classification

(51) **Int. Cl.**

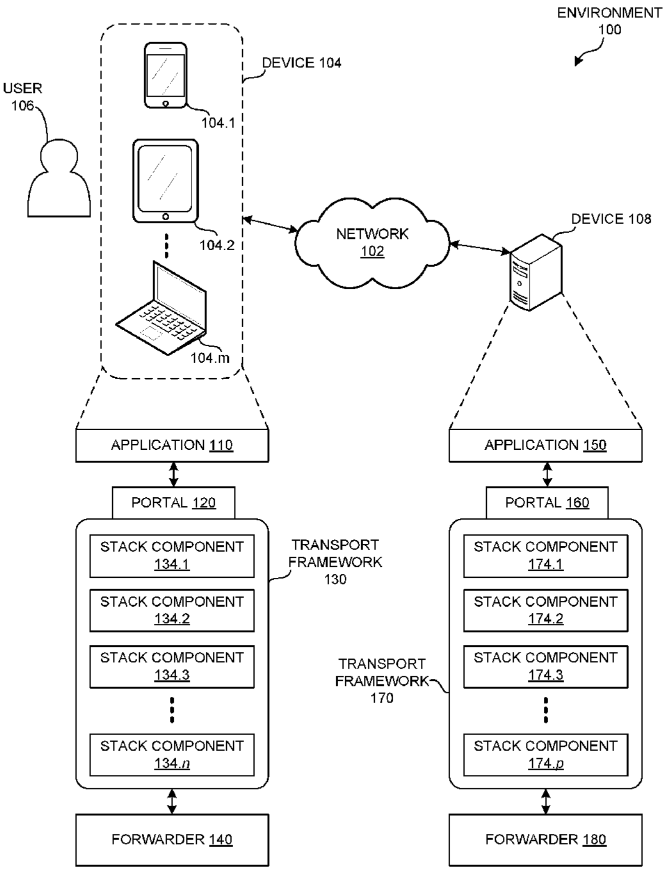
H04L 12/741	(2006.01)
H04L 29/06	(2006.01)
H04L 29/08	(2006.01)
H04L 12/841	(2006.01)
H04L 12/807	(2006.01)
H04L 12/723	(2006.01)
H04L 29/12	(2006.01)
G06F 15/167	(2006.01)
H04L 12/803	(2006.01)

(52) **U.S. Cl.**

CPC ***H04L 45/74*** (2013.01); ***G06F 15/167*** (2013.01); ***H04L 63/0823*** (2013.01); ***H04L 67/2842*** (2013.01); ***H04L 67/1097*** (2013.01); ***H04L 47/125*** (2013.01); ***H04L 47/27*** (2013.01); ***H04L 69/16*** (2013.01); ***H04L 69/02*** (2013.01); ***H04L 45/50*** (2013.01); ***H04L 61/6009*** (2013.01); ***H04L 47/283*** (2013.01)

(57) ABSTRACT

One embodiment provides a system that facilitates querying of historical network information. During operation, the system generates a query for historical information associated with interest and content object packets, wherein a name for an interest is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, wherein the query is based on a name prefix that includes one or more contiguous name components. The system transmits the query to a responding entity. In response to receiving the historical information from the responding entity, the system performs an operation that increases network efficiency based on the historical information, thereby facilitating a protocol for querying the historical information to increase network efficiency.



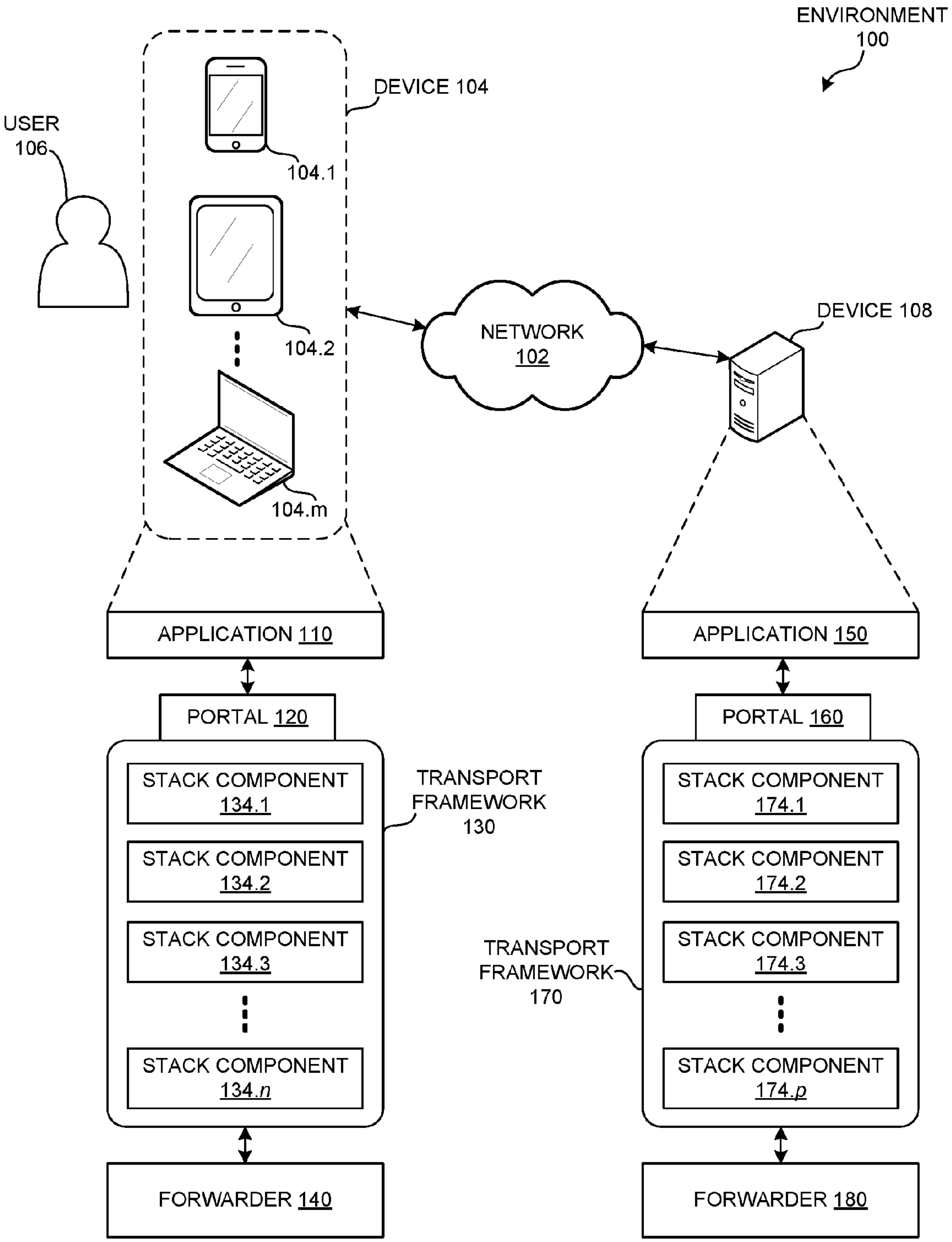

ENVIRONMENT 100
USER 106
DEVICE 104
104.1
104.2
104.m
NETWORK 102
DEVICE 108
APPLICATION 110
PORTAL 120
STACK COMPONENT 134.1
STACK COMPONENT 134.2
STACK COMPONENT 134.3
STACK COMPONENT 134.n
TRANSPORT FRAMEWORK 130
FORWARDER 140
APPLICATION 150
PORTAL 160
STACK COMPONENT 174.1
STACK COMPONENT 174.2
STACK COMPONENT 174.3
STACK COMPONENT 174.p
TRANSPORT FRAMEWORK 170
FORWARDER 180

FIG. 1

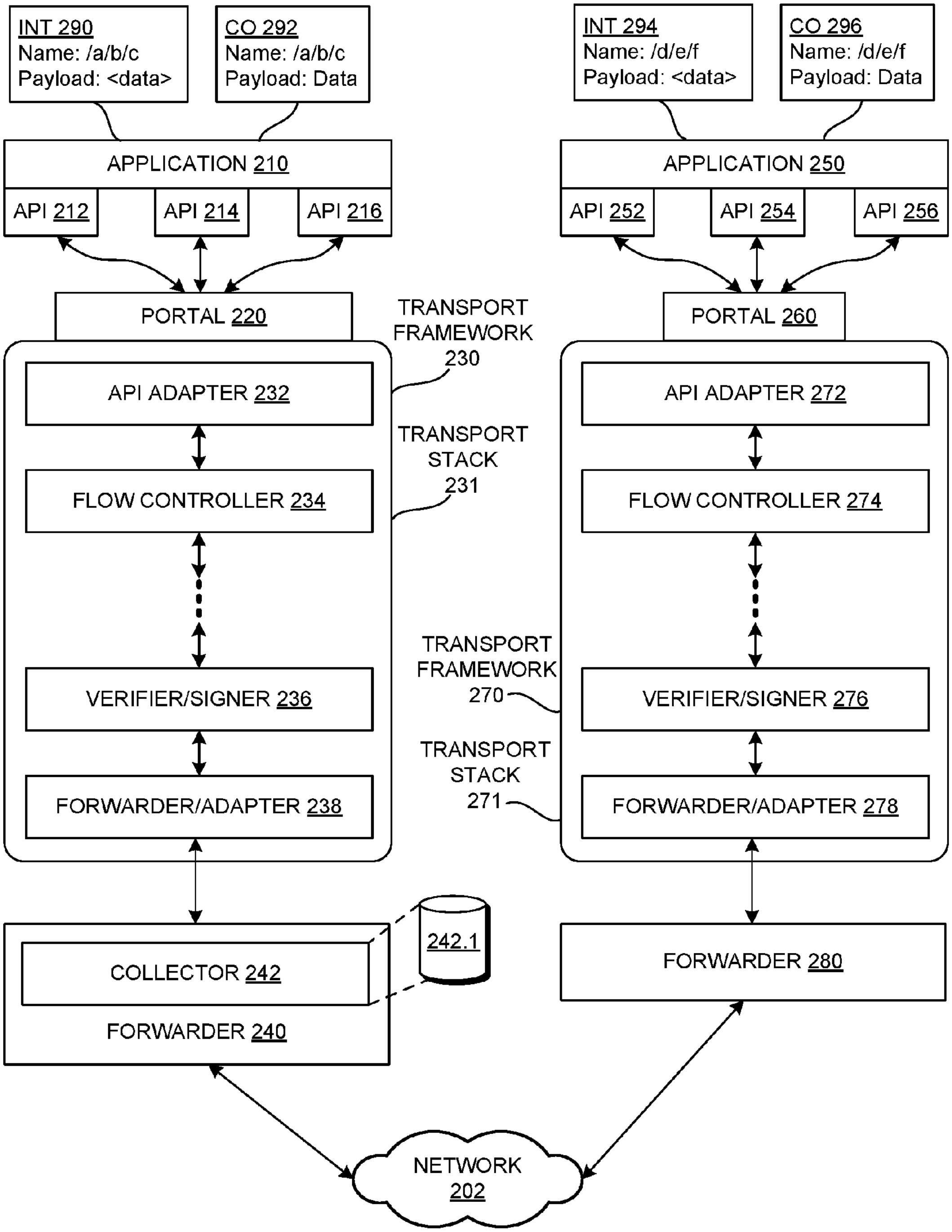

INT 290
Name: /a/b/c
Payload: <data>
CO 292
Name: /a/b/c
Payload: Data
INT 294
Name: /d/e/f
Payload: <data>
CO 296
Name: /d/e/f
Payload: Data
APPLICATION 210
API 212
API 214
API 216
APPLICATION 250
API 252
API 254
API 256
PORTAL 220
PORTAL 260
TRANSPORT FRAMEWORK 230
TRANSPORT STACK 231
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
TRANSPORT FRAMEWORK 270
TRANSPORT STACK 271
242.1
COLLECTOR 242
FORWARDER 240
FORWARDER 280
NETWORK 202

FIG. 2A

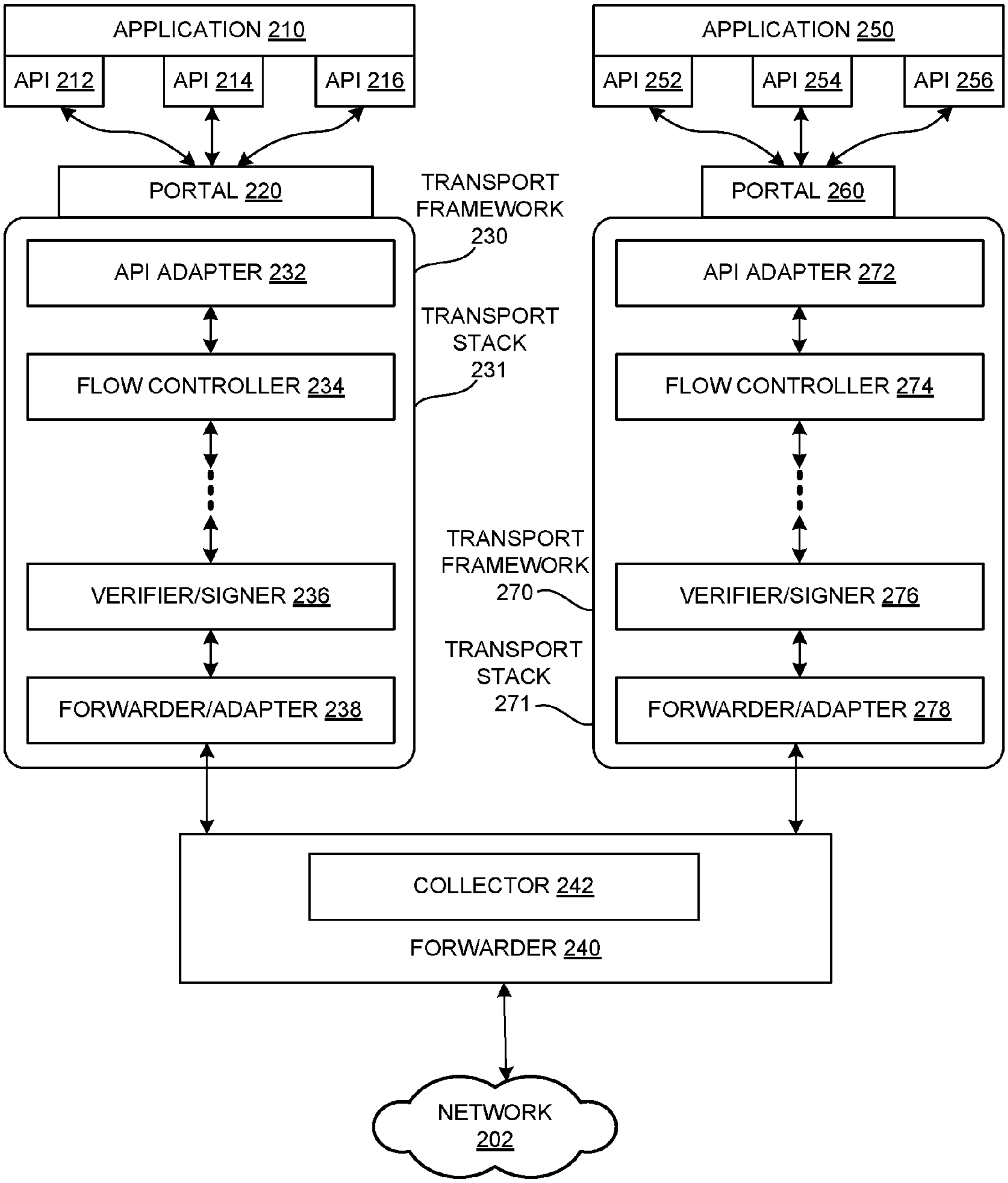

APPLICATION 210
API 212
API 214
API 216
PORTAL 220
TRANSPORT FRAMEWORK 230
API ADAPTER 232
TRANSPORT STACK 231
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
TRANSPORT FRAMEWORK 270
VERIFIER/SIGNER 276
TRANSPORT STACK 271
FORWARDER/ADAPTER 278
COLLECTOR 242
FORWARDER 240
NETWORK 202

FIG. 2B

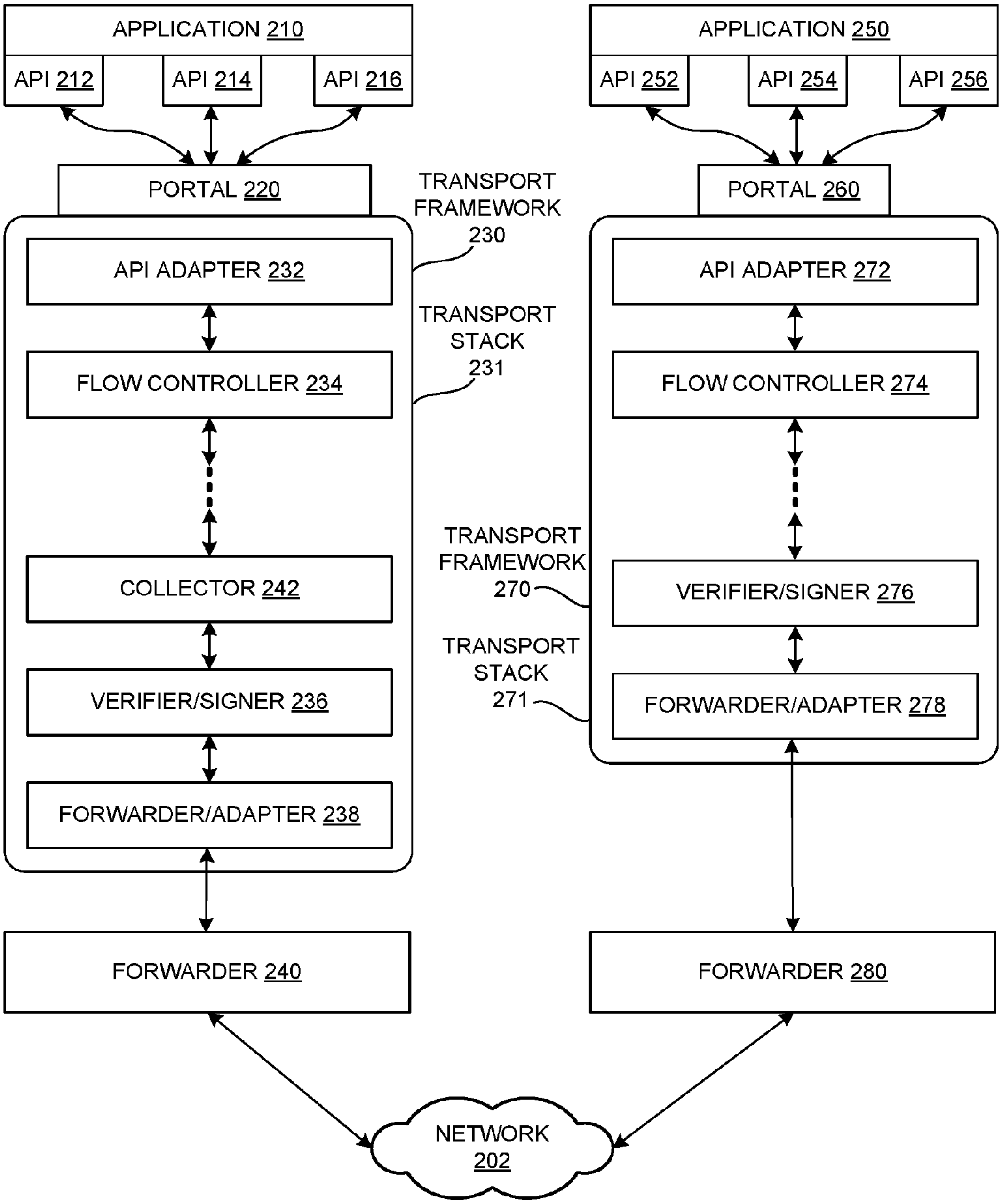

APPLICATION 210
API 212
API 214
API 216
PORTAL 220
TRANSPORT FRAMEWORK 230
API ADAPTER 232
TRANSPORT STACK 231
FLOW CONTROLLER 234
COLLECTOR 242
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
FORWARDER 240
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
TRANSPORT FRAMEWORK 270
VERIFIER/SIGNER 276
TRANSPORT STACK 271
FORWARDER/ADAPTER 278
FORWARDER 280
NETWORK 202

FIG. 2C

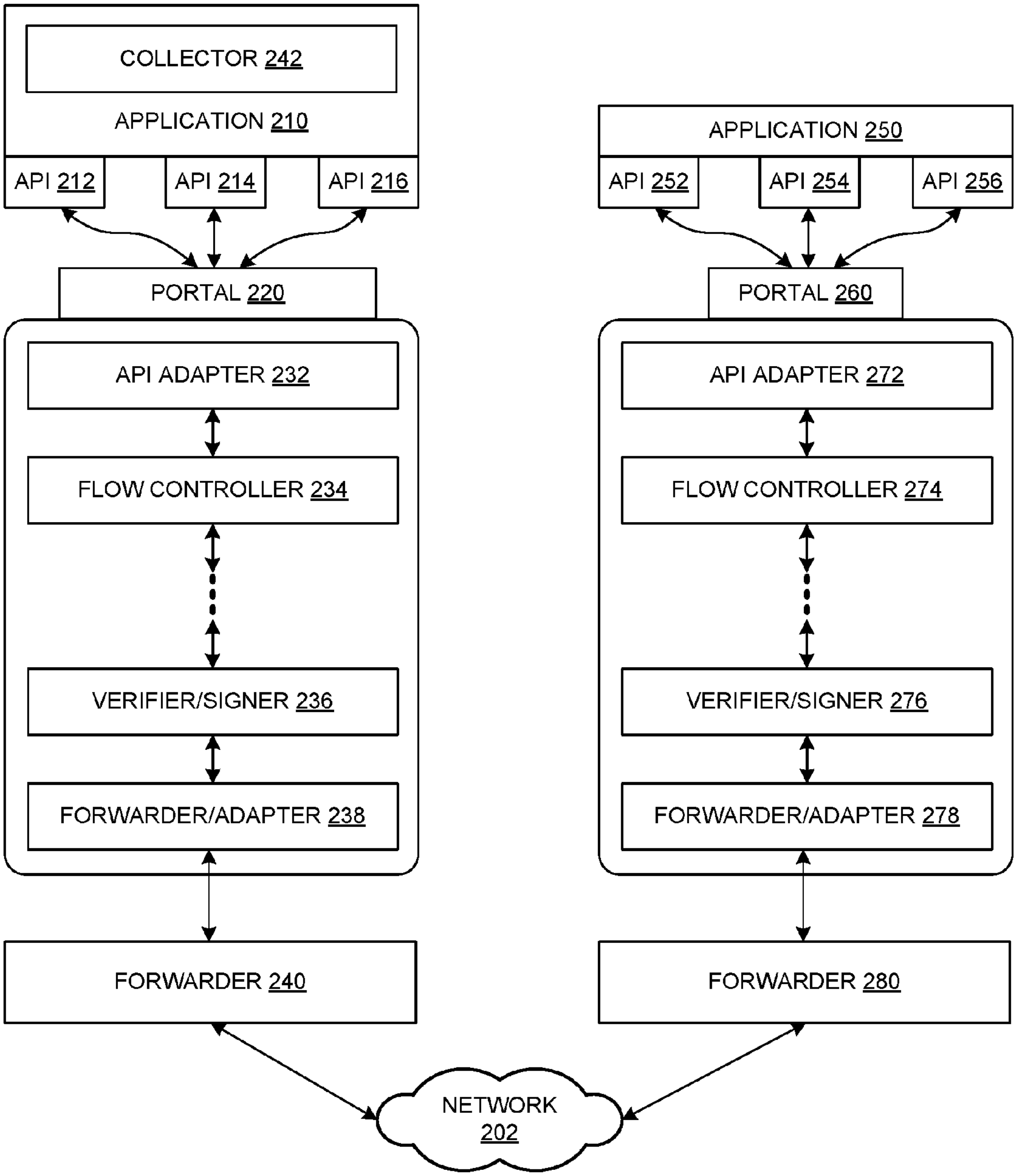

COLLECTOR 242
APPLICATION 210
API 212
API 214
API 216
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
FORWARDER 240
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
FORWARDER 280
NETWORK
202

FIG. 2D

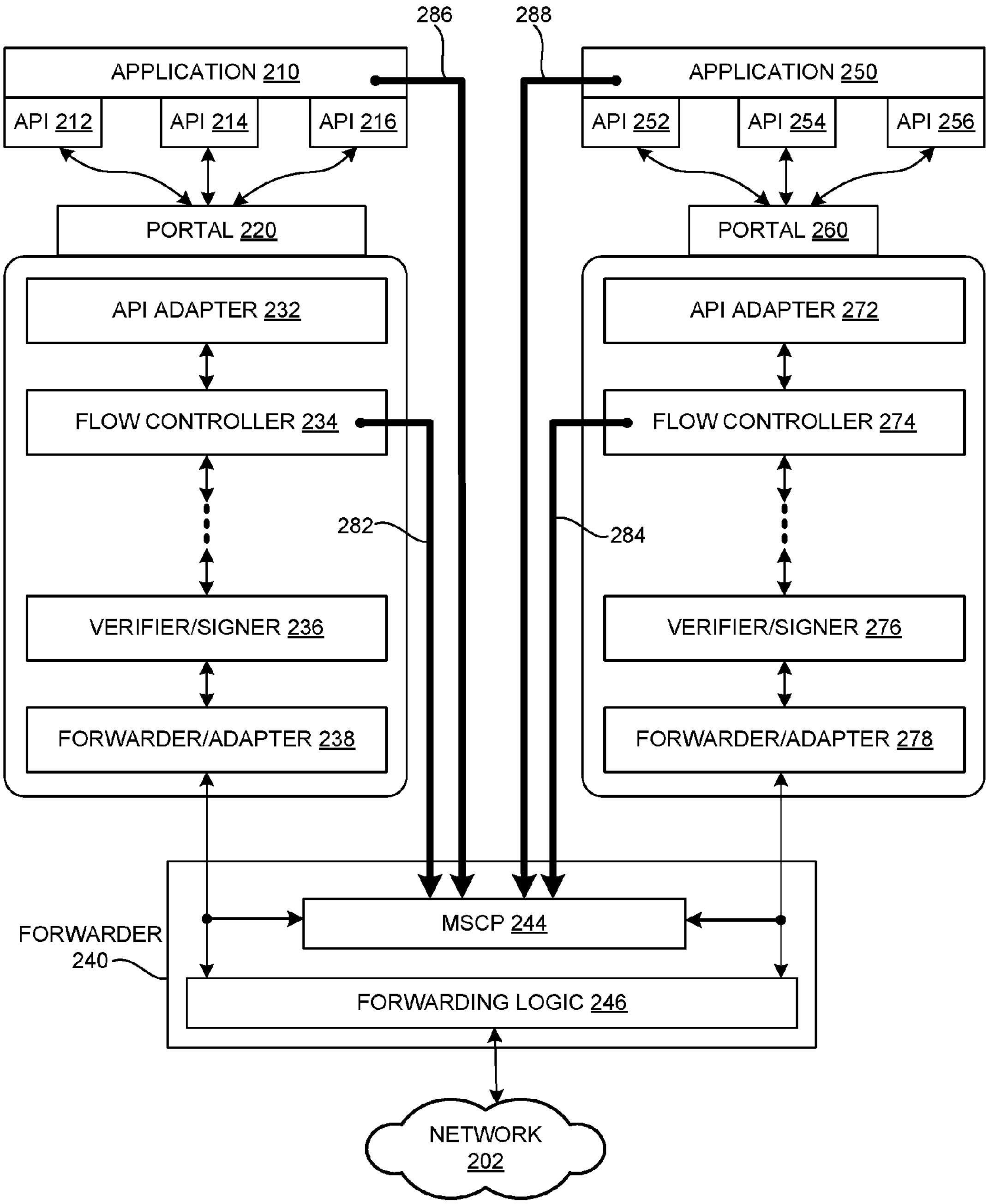

286
288
APPLICATION 210
API 212
API 214
API 216
APPLICATION 250
API 252
API 254
API 256
PORTAL 220
PORTAL 260
API ADAPTER 232
FLOW CONTROLLER 234
282
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
API ADAPTER 272
FLOW CONTROLLER 274
284
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
FORWARDER 240
MSCP 244
FORWARDING LOGIC 246
NETWORK 202

FIG. 2E

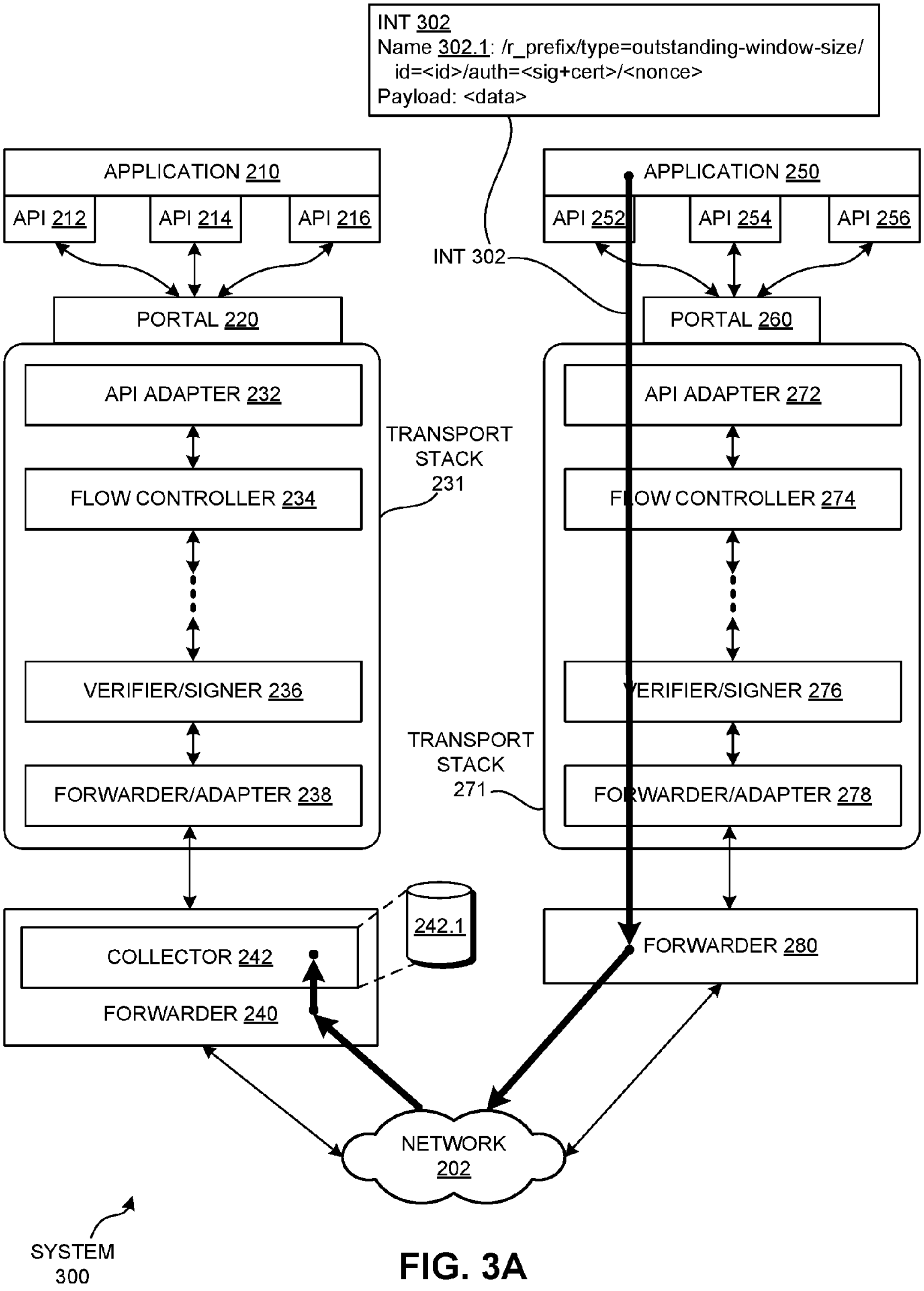

INT 302
Name 302.1: /r_prefix/type=outstanding-window-size/
id=<id>/auth=<sig+cert>/<nonce>
Payload: <data>
APPLICATION 210
API 212
API 214
API 216
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
TRANSPORT STACK 231
APPLICATION 250
API 252
API 254
API 256
INT 302
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
TRANSPORT STACK 271
COLLECTOR 242
242.1
FORWARDER 240
FORWARDER 280
NETWORK 202
SYSTEM 300

FIG. 3A

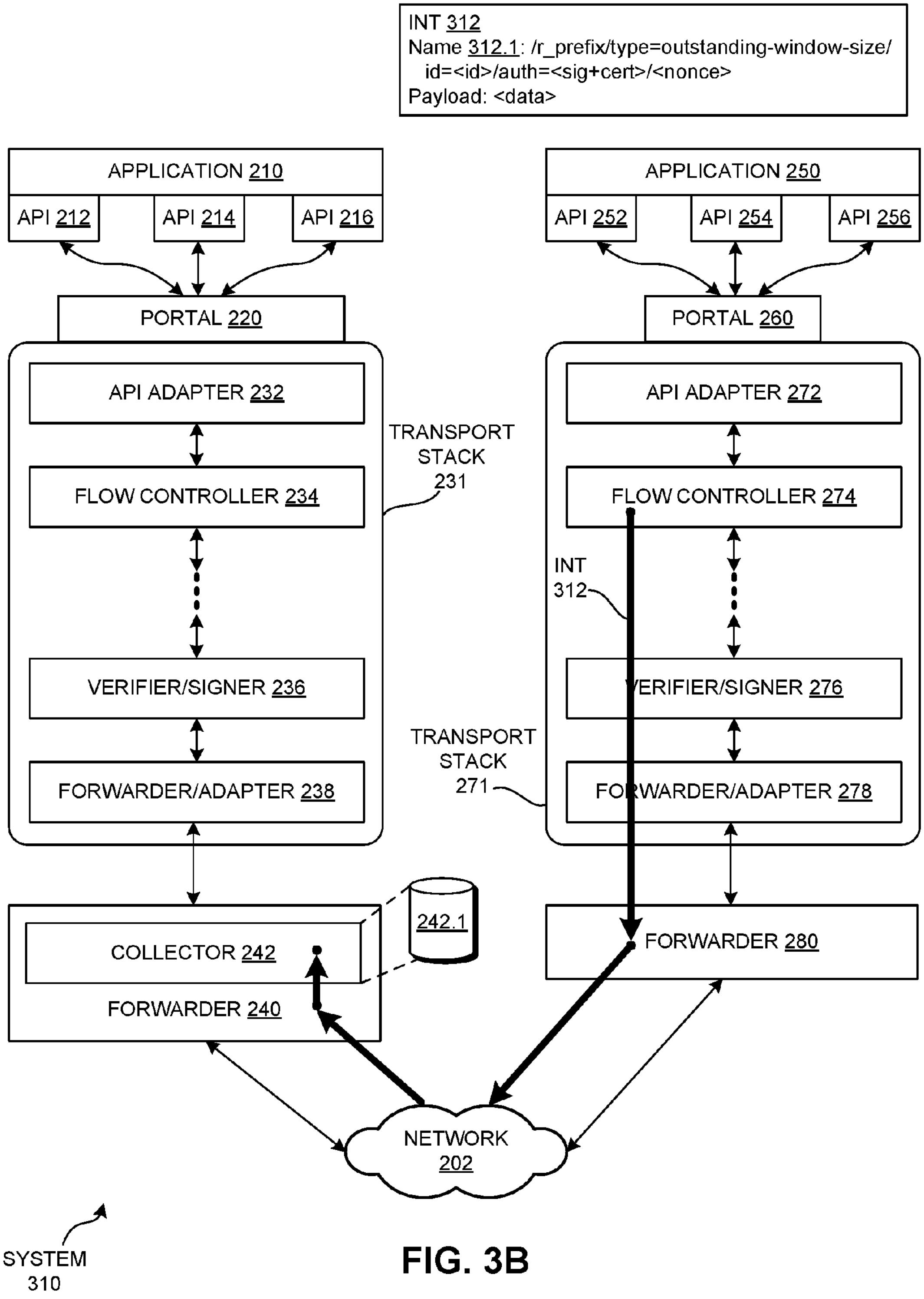

INT 312
Name 312.1: /r_prefix/type=outstanding-window-size/
id=<id>/auth=<sig+cert>/<nonce>
Payload: <data>
APPLICATION 210
API 212
API 214
API 216
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
TRANSPORT STACK 231
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
INT 312
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
TRANSPORT STACK 271
COLLECTOR 242
FORWARDER 240
242.1
FORWARDER 280
NETWORK 202
SYSTEM 310

FIG. 3B

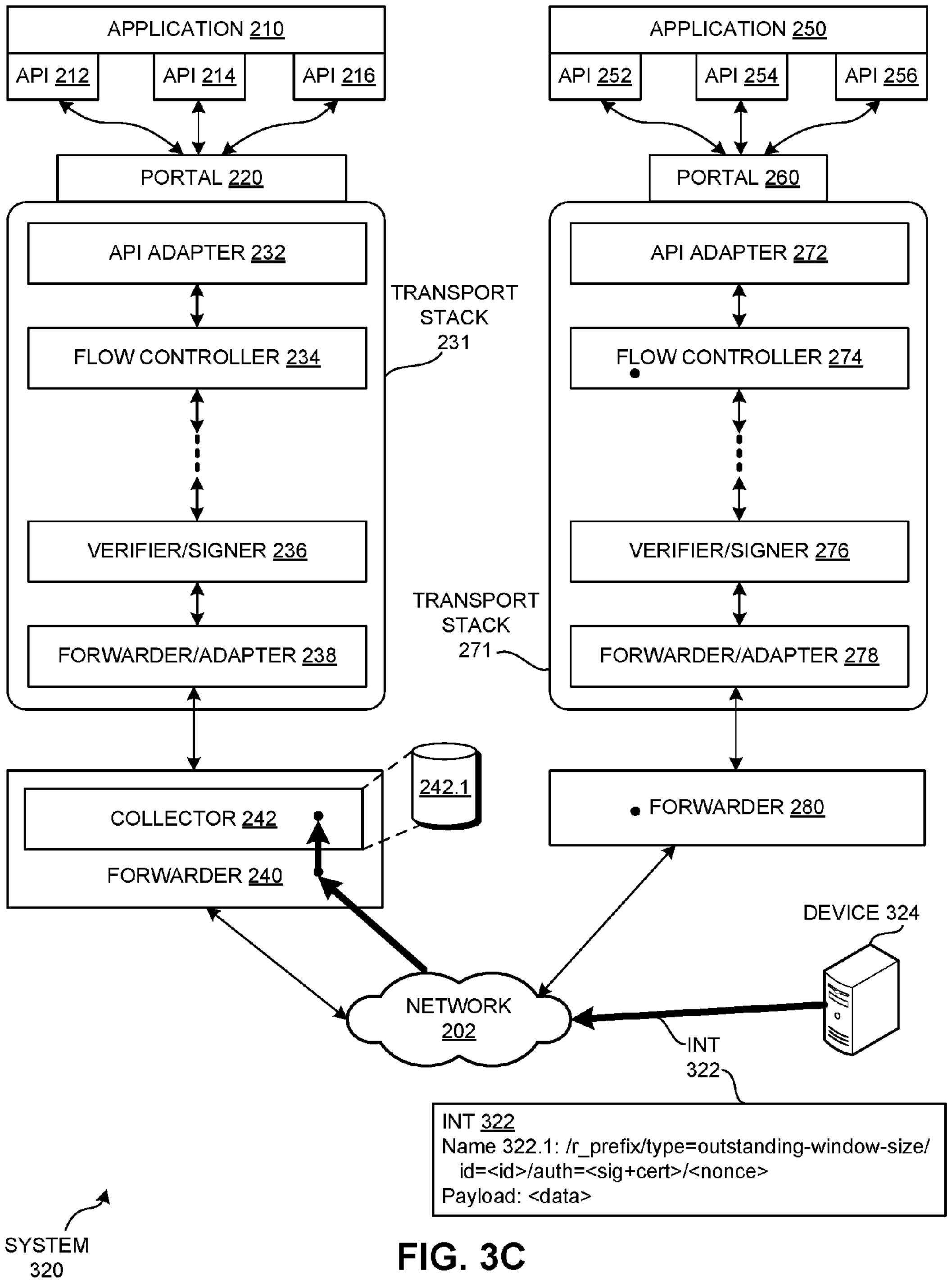

APPLICATION 210
API 212
API 214
API 216
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
TRANSPORT STACK 231
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
TRANSPORT STACK 271
COLLECTOR 242
FORWARDER 240
242.1
FORWARDER 280
NETWORK 202
DEVICE 324
INT 322
INT 322
Name 322.1: /r_prefix/type=outstanding-window-size/
id=<id>/auth=<sig+cert>/<nonce>
Payload: <data>
SYSTEM 320

FIG. 3C

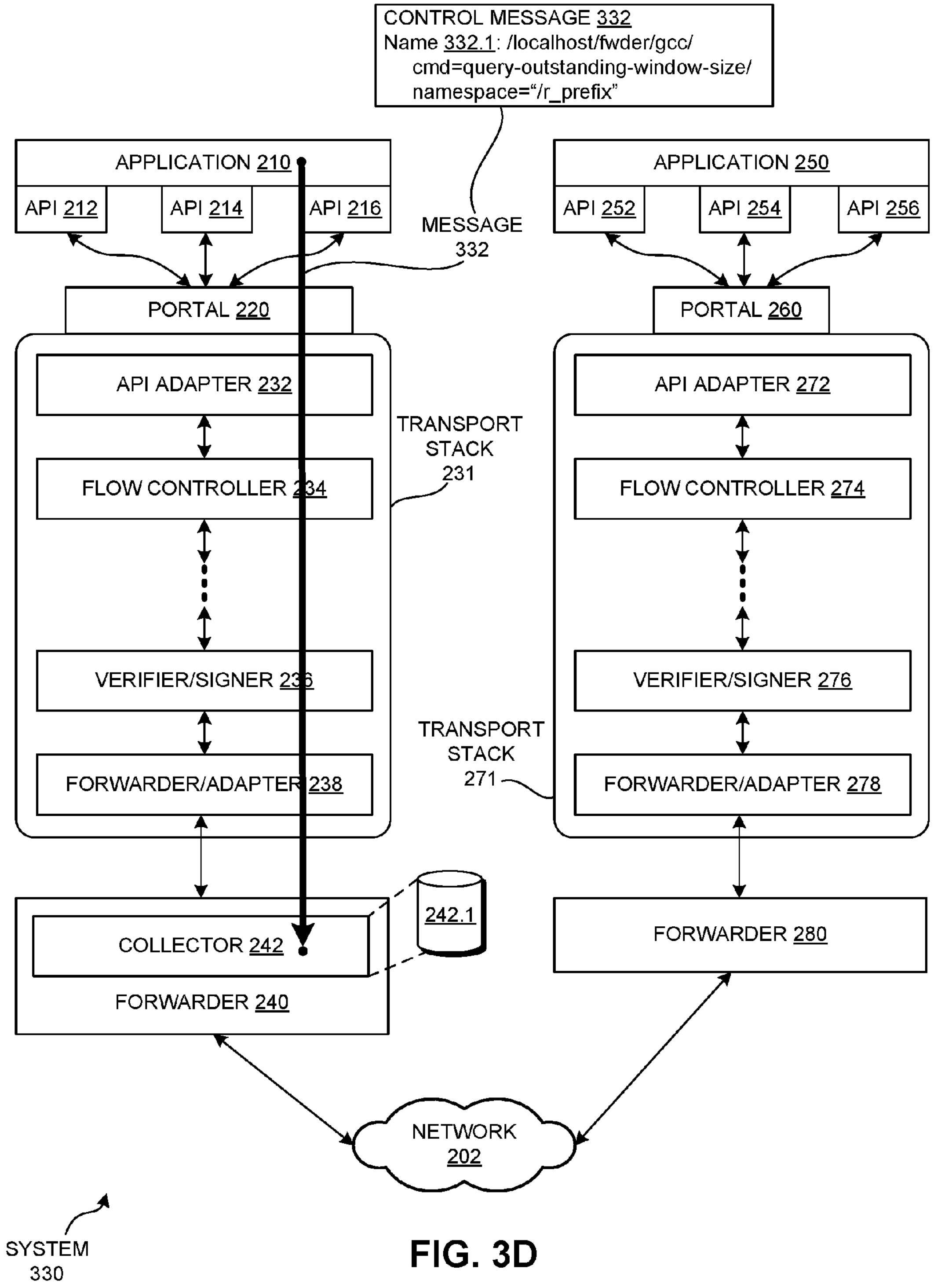

CONTROL MESSAGE 332
Name 332.1: /localhost/fwder/gcc/
cmd=query-outstanding-window-size/
namespace="/r_prefix"
APPLICATION 210
API 212
API 214
API 216
MESSAGE
332
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
TRANSPORT
STACK
231
COLLECTOR 242
242.1
FORWARDER 240
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
FORWARDER/ADAPTER 278
TRANSPORT
STACK
271
FORWARDER 280
NETWORK
202
SYSTEM
330

FIG. 3D

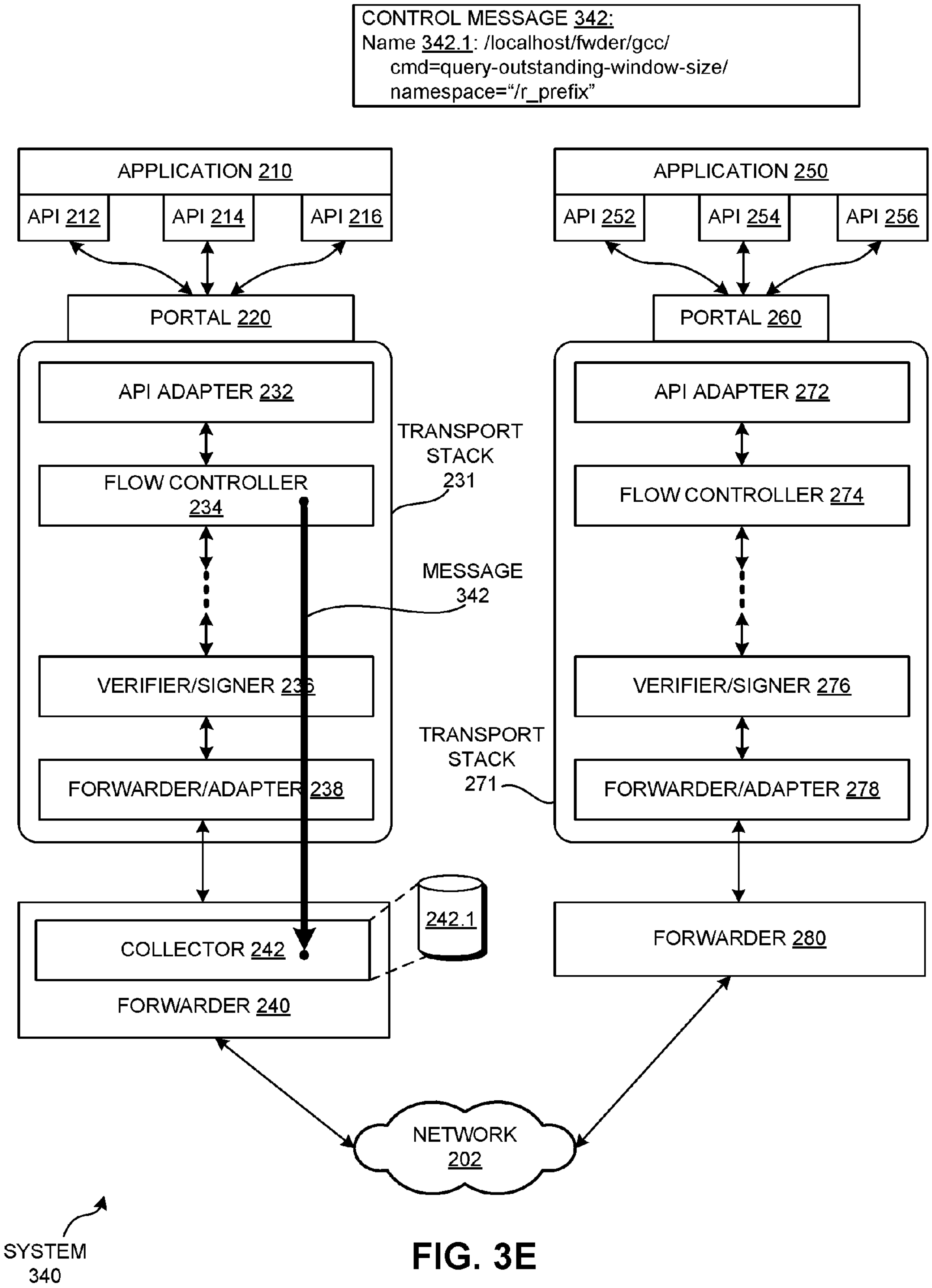

CONTROL MESSAGE 342:
Name 342.1: /localhost/fwder/gcc/
cmd=query-outstanding-window-size/
namespace="/r_prefix"
APPLICATION 210
API 212
API 214
API 216
PORTAL 220
API ADAPTER 232
FLOW CONTROLLER 234
TRANSPORT STACK 231
MESSAGE 342
VERIFIER/SIGNER 236
FORWARDER/ADAPTER 238
COLLECTOR 242
242.1
FORWARDER 240
APPLICATION 250
API 252
API 254
API 256
PORTAL 260
API ADAPTER 272
FLOW CONTROLLER 274
VERIFIER/SIGNER 276
TRANSPORT STACK 271
FORWARDER/ADAPTER 278
FORWARDER 280
NETWORK 202
SYSTEM 340

FIG. 3E

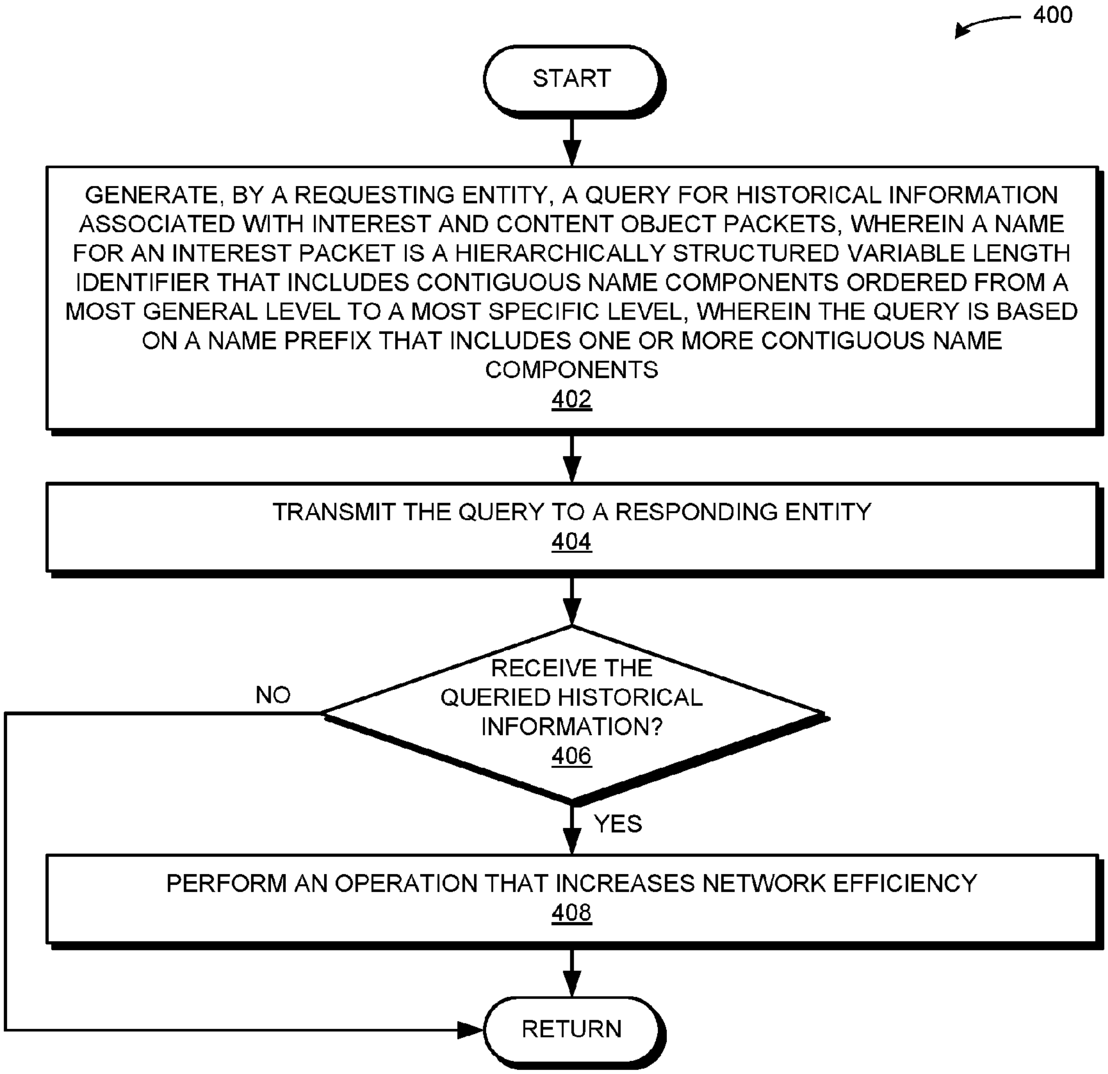

400
START
GENERATE, BY A REQUESTING ENTITY, A QUERY FOR HISTORICAL INFORMATION ASSOCIATED WITH INTEREST AND CONTENT OBJECT PACKETS, WHEREIN A NAME FOR AN INTEREST PACKET IS A HIERARCHICALLY STRUCTURED VARIABLE LENGTH IDENTIFIER THAT INCLUDES CONTIGUOUS NAME COMPONENTS ORDERED FROM A MOST GENERAL LEVEL TO A MOST SPECIFIC LEVEL, WHEREIN THE QUERY IS BASED ON A NAME PREFIX THAT INCLUDES ONE OR MORE CONTIGUOUS NAME COMPONENTS
402
TRANSMIT THE QUERY TO A RESPONDING ENTITY
404
RECEIVE THE QUERIED HISTORICAL INFORMATION?
406
NO
YES
PERFORM AN OPERATION THAT INCREASES NETWORK EFFICIENCY
408
RETURN

FIG. 4

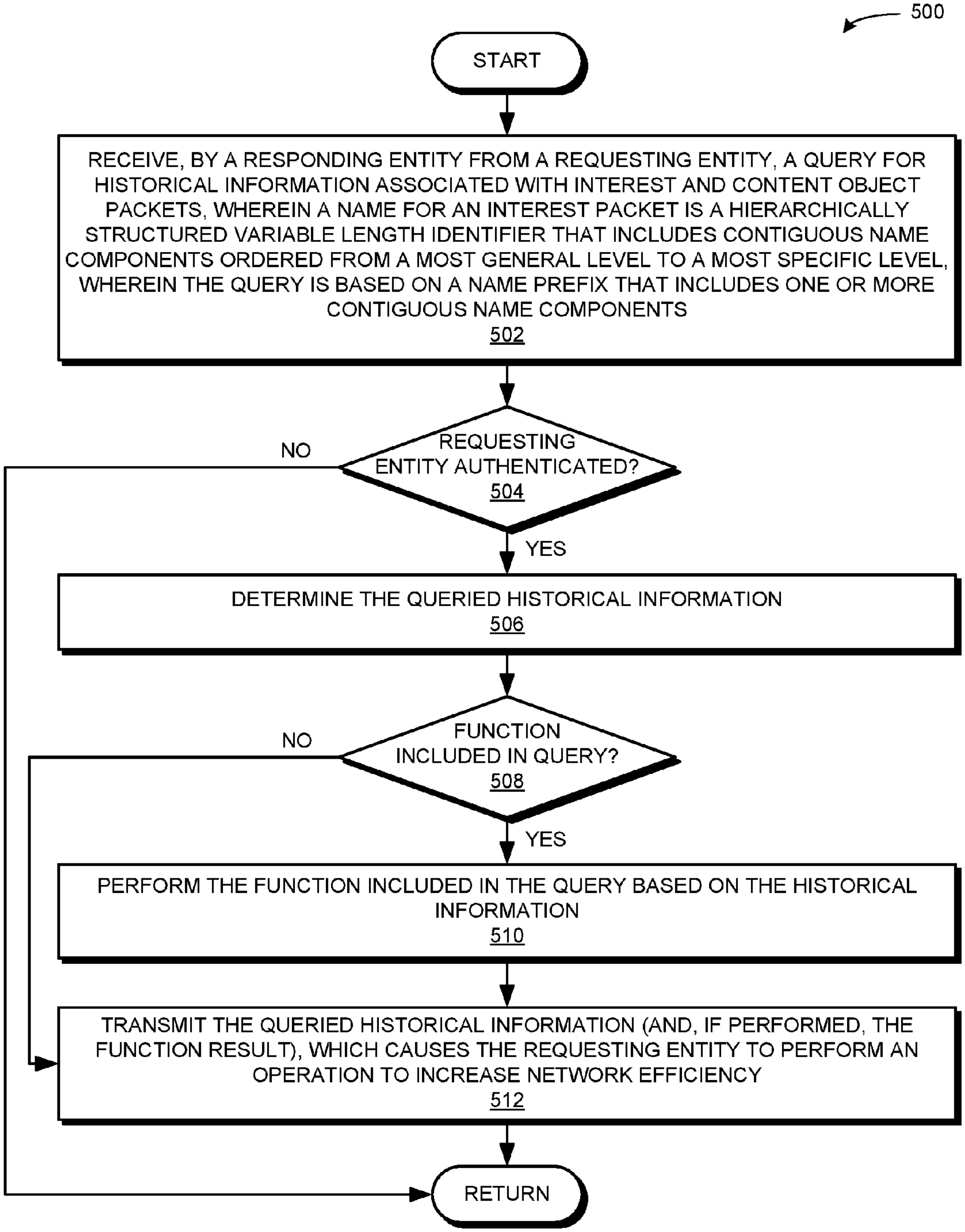

500
START
RECEIVE, BY A RESPONDING ENTITY FROM A REQUESTING ENTITY, A QUERY FOR HISTORICAL INFORMATION ASSOCIATED WITH INTEREST AND CONTENT OBJECT PACKETS, WHEREIN A NAME FOR AN INTEREST PACKET IS A HIERARCHICALLY STRUCTURED VARIABLE LENGTH IDENTIFIER THAT INCLUDES CONTIGUOUS NAME COMPONENTS ORDERED FROM A MOST GENERAL LEVEL TO A MOST SPECIFIC LEVEL, WHEREIN THE QUERY IS BASED ON A NAME PREFIX THAT INCLUDES ONE OR MORE CONTIGUOUS NAME COMPONENTS
502
REQUESTING ENTITY AUTHENTICATED?
504
NO
YES
DETERMINE THE QUERIED HISTORICAL INFORMATION
506
FUNCTION INCLUDED IN QUERY?
508
NO
YES
PERFORM THE FUNCTION INCLUDED IN THE QUERY BASED ON THE HISTORICAL INFORMATION
510
TRANSMIT THE QUERIED HISTORICAL INFORMATION (AND, IF PERFORMED, THE FUNCTION RESULT), WHICH CAUSES THE REQUESTING ENTITY TO PERFORM AN OPERATION TO INCREASE NETWORK EFFICIENCY
512
RETURN

FIG. 5

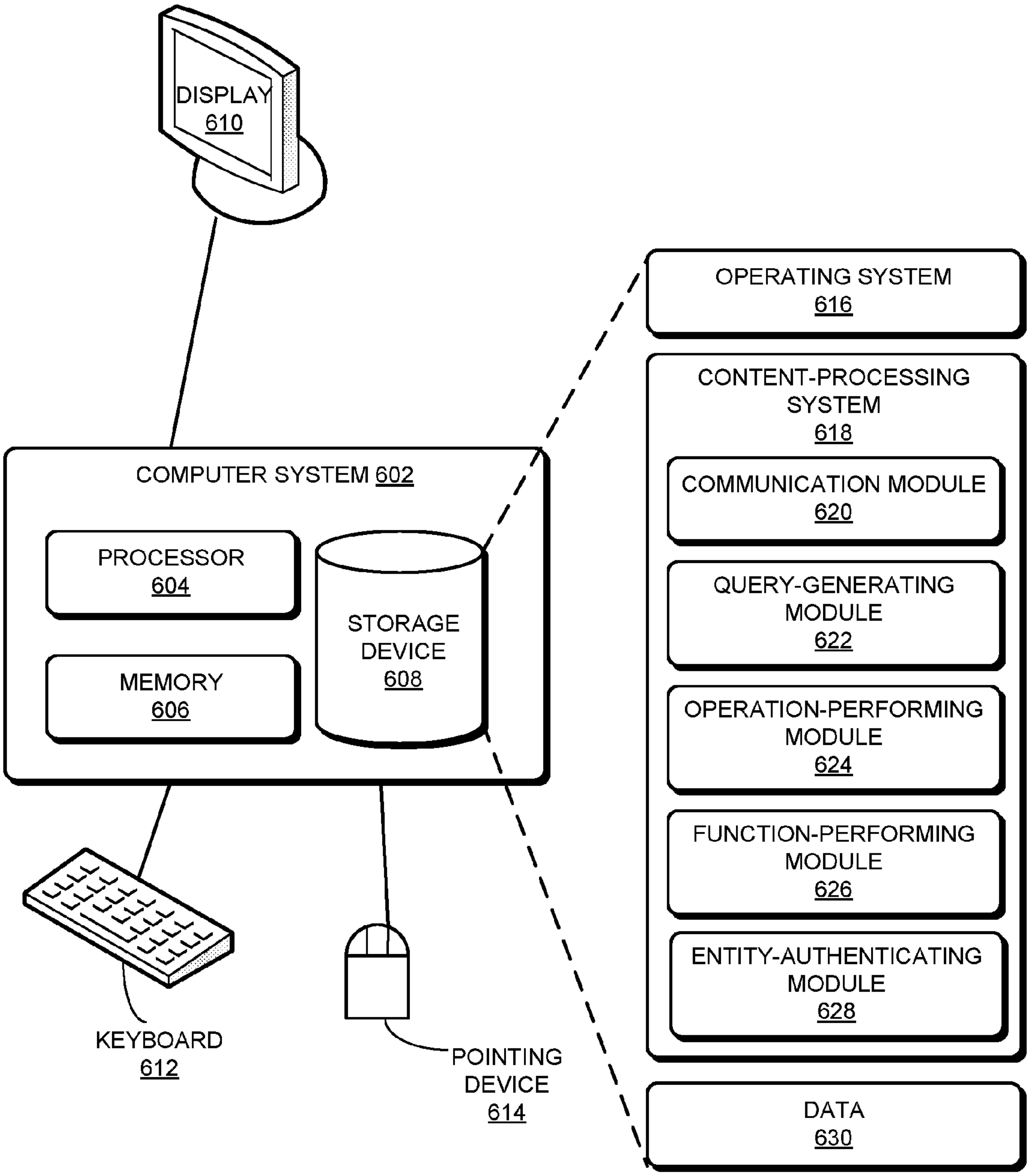

DISPLAY
610
COMPUTER SYSTEM 602
PROCESSOR
604
MEMORY
606
STORAGE
DEVICE
608
KEYBOARD
612
POINTING
DEVICE
614
OPERATING SYSTEM
616
CONTENT-PROCESSING
SYSTEM
618
COMMUNICATION MODULE
620
QUERY-GENERATING
MODULE
622
OPERATION-PERFORMING
MODULE
624
FUNCTION-PERFORMING
MODULE
626
ENTITY-AUTHENTICATING
MODULE
628
DATA
630

FIG. 6

PROTOCOL TO QUERY FOR HISTORICAL NETWORK INFORMATION IN A CONTENT CENTRIC NETWORK

RELATED APPLICATIONS

[0001] The subject matter of this application is related to the subject matter in the following applications:

[0002] U.S. patent application Ser. No. 13/847,814 (Attorney Docket No. PARC-20120537-US-NP), entitled "ORDERED-ELEMENT NAMING FOR NAME-BASED PACKET FORWARDING," by inventor Ignacio Solis, filed 20 Mar. 2013 (hereinafter "U.S. patent application Ser. No. 13/847,814");

[0003] U.S. patent application Ser. No. 12/338,175 (Attorney Docket No. PARC-20080626-US-NP), entitled "CONTROLLING THE SPREAD OF INTERESTS AND CONTENT IN A CONTENT CENTRIC NETWORK," by inventors Van L. Jacobson and Diana K. Smetters, filed 18 Dec. 2008 (hereinafter "U.S. patent application Ser. No. 12/338,175");

[0004] U.S. patent application Ser. No. 14/334,530 (Attorney Docket No. PARC-20140518US01), entitled "INTEREST RETURN CONTROL MESSAGE," by inventors Marc E. Mosko, Ignacio Solis, and Ersin Uzun, filed 17 Jul. 2014 (hereinafter "U.S. patent application Ser. No. 14/334,530"); and

[0005] U.S. patent application Ser. No. 14/746,490 (Attorney Docket No. PARC-20141532US01), entitled "TRANSPORT STACK NAME SCHEME AND IDENTITY MANAGEMENT," by inventors Christopher A. Wood and Glenn C. Scott, filed 22 Jun. 2015 (hereinafter "U.S. patent application Ser. No. 14/746,490");

the disclosures of which are herein incorporated by reference in their entirety.

BACKGROUND

[0006] Field

[0007] This disclosure is generally related to distribution of digital content. More specifically, this disclosure is related to a system for querying historical network information in a content centric network, which facilitates users of the system to increase network efficiency.

[0008] Related Art

[0009] The proliferation of the Internet and e-commerce continues to create a vast amount of digital content. Content centric network (CCN) architectures have been designed to facilitate accessing and processing such digital content. A CCN includes entities, or nodes, such as network clients, forwarders (e.g., routers), and content producers, which communicate with each other by sending interest packets for various content items and receiving content object packets in return. CCN interests and content objects are identified by their unique names, which are typically hierarchically structured variable length identifiers (HSVLI). An HSVLI can include contiguous name components ordered from a most general level to a most specific level. A CCN name prefix, or namespace, may include one or more contiguous name components beginning from the most general level.

[0010] Some transport protocols implement flow and congestion control by maintaining a window of messages (e.g., packets) sent from a client (e.g, a consumer) to a server (e.g., a content producer). Upon sending a packet, the consumer adds a packet to the window, and upon receiving a responsive packet, the consumer removes a packet from the window. For a window with a size of "w," only w messages can be outstanding at any given time. Some transport protocols (such as TCP) use a sliding window such that w is a variable that changes dynamically based on network conditions. For example, if the protocol determines congestion, e.g., due to heavy traffic of neighboring nodes, the consumer can decrease w so that fewer messages are sent to the network. Similarly, if the protocol determines that the network is not congested, the consumer can increase w so that more messages can be sent for better throughput and latency performance.

[0011] While a CCN brings many desired features to a network, some issues remain unsolved with enabling a CCN transport protocol to increase network efficiency by allowing system users to query the network for historical network information.

SUMMARY

[0012] One embodiment provides a system that facilitates querying of historical network information. During operation, the system generates a query for historical information associated with interest and corresponding content object packets, wherein a name for an interest packet is a hierarchically structured variable length identifier ("HSVLI") that includes contiguous name components ordered from a most general level to a most specific level, wherein the query is based on a name prefix that includes one or more contiguous name components. The system transmits the query to a responding entity. In response to receiving the historical information from the responding entity, the system performs an operation that increases network efficiency based on the historical information, thereby facilitating a protocol for querying the historical information to increase network efficiency.

[0013] In some embodiments, the query is an interest packet that indicates one or more of: a routable prefix which includes one or more contiguous name components beginning from the most general level; a user identifier of the requesting entity; authentication information of the requesting entity; a type for the query; a string for the query; one or more parameters for the query; a function for the query; and a random nonce.

[0014] In some embodiments, the query is an interest packet with a name that is an HSVLI that includes contiguous name components ordered from a most general level to a most specific level, wherein the name includes one or more of: a routable prefix which includes one or more contiguous name components beginning from the most general level; a user identifier of the requesting entity; authentication information of the requesting entity; a type for the query; a string for the query; one or more parameters for the query; a function for the query; and a random nonce.

[0015] In some embodiments, the method is performed by a requesting entity which is a component of a stack of communication modules, the responding entity is a local forwarder that services the stack, and the query is a control message that indicates one or more of: an identifier for the local forwarder; a component which is responsible for collecting and storing the historical information, wherein the component resides in the local forwarder; a type for the query; a string for the query; one or more parameters for the query; and a function for the query.

[0016] In some embodiments, the operation is one or more of: setting or changing a window size; setting or changing a rate of transmission for re-transmitted interests; setting or changing a rate of transmission for original interests, wherein an original interest is not a re-transmitted interest; and an operation related to increasing the efficiency of the network.

[0017] In some embodiments, the query includes a command to perform a function on the historical information, and the function includes one or more of: computing an estimate of an average round trip time for an interest and a corresponding content object based on the name prefix, wherein a round trip time begins when an interest is transmitted and ends when a corresponding content object is received, or begins when an interest is received and ends when a corresponding content object is transmitted, wherein the estimate is based on a plurality of average round trip times for a corresponding plurality of related namespaces that share at least one name prefix; computing an estimate of a size of a transmission window based on a number of outstanding interests for the name prefix for a predetermined period of time; and performing a function based on one or more interests as input, wherein an output for the function is a variable which can be stored by the system, wherein the function is defined by the system or a user of the system.

[0018] In some embodiments, the method is performed by a requesting entity which is one or more of: an application associated with a first stack, wherein the responding entity resides in or is associated with the first stack; an application associated with a second stack that is different from the first stack; a stack component of the first stack, wherein the stack component is different from the responding entity; a stack component of the second stack; and any other element or node in the network.

[0019] In some embodiments, the responding entity resides in one or more of: an application; a single stack; a shared stack; a single forwarder; a shared forwarder; and any node in a network.

[0020] In some embodiments, the historical information associated with the packets is one or more of: a round trip time that begins when an outgoing interest is transmitted and ends when a corresponding incoming content object is received; a number of outgoing interests for which a corresponding incoming content object has not been received; a number of outgoing interests for which a corresponding incoming content object is received based on a predetermined amount of time or the round trip time; a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time; a number of outgoing interests that time out based on the predetermined amount of time or the round trip time; a number of outgoing interests which are retransmitted based on the predetermined amount of time or the round trip time; a number of re-transmitted outgoing interests that time out based on the predetermined amount of time or the round trip time; a number of interest return messages received based on the predetermined amount of time or the round trip time, wherein an interest return message is received in response to an outgoing interest and is identified based on a code indicated in the message; a number of outgoing interests aggregated based on the predetermined amount of time or the round trip time; a number of active upstream paths identified for a given time; a strategy for forwarding packets; a first number of transmitted original interests, wherein an original interest is not a re-transmitted interest, and wherein the first number of original interests include names that share one or more name prefixes; a second number of transmitted original interests, wherein the second number of original interests include names that do not share any name prefixes; a first number of active entries in a forwarding information base, wherein the first number of entries include names that share one or more name prefixes; and a second number of active entries in a forwarding information base, wherein the second number of entries include names that do not share any name prefixes.

[0021] In some embodiments, the historical information associated with the packets is one or more of: a round trip time that begins when an incoming interest is received and ends when a corresponding incoming content object is transmitted; a number of incoming interests for which a corresponding outgoing content object has not been transmitted; a number of incoming interests for which a corresponding outgoing content object is transmitted based on a predetermined amount of time or the round trip time; a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time; a number of incoming interests that time out based on the predetermined amount of time or the round trip time; a number of re-transmitted incoming interests based on the predetermined amount of time or the round trip time; a number of re-transmitted incoming interests that time out based on the predetermined amount of time or the round trip time; a number of interest return messages transmitted based on the predetermined amount of time or the round trip time, wherein an interest return message is transmitted in response to an incoming interest and is identified based on a code indicated in the message; and a number of incoming interests aggregated based on a predetermined amount of time or a round trip time.

[0022] Another embodiment provides a system that facilitates querying of historical network information. During operation, the system receives a query from a requesting entity for historical information associated with interest and corresponding content object packets. In response to authenticating the requesting entity, the system transmits the queried historical information, which causes the requesting entity to perform an operation to increase network efficiency based on the historical information, thereby facilitating a protocol for querying the historical information to increase network efficiency.

[0023] In some embodiments, the requesting entity is one or more of: an application associated with a first stack, wherein the responding entity resides in the first stack; an application associated with a second stack that is different from the first stack; a stack component of the first stack, wherein the stack component is different from the responding entity; a stack component of the second stack; and any other element or node in the network.

[0024] In some embodiments, the method is performed by a responding entity which resides in one or more of: an application; a single stack; a shared stack; a single forwarder; a shared forwarder; and any node in a network.

BRIEF DESCRIPTION OF THE FIGURES

[0025] FIG. **1** illustrates an exemplary environment which facilitates querying of historical network information in a content centric network, in accordance with an embodiment of the present invention.

[0026] FIG. **2**A illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a single forwarder, in accordance with an embodiment of the present invention.

[0027] FIG. **2**B illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a shared forwarder, in accordance with an embodiment of the present invention.

[0028] FIG. **2**C illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a single transport stack of the transport framework, in accordance with an embodiment of the present invention.

[0029] FIG. **2**D illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in an application associated with the transport framework, in accordance with an embodiment of the present invention.

[0030] FIG. **2**E illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a forwarder as a message stream co-processor element, in accordance with an embodiment of the present invention.

[0031] FIG. **3**A illustrates an exemplary communication in a system which facilitates querying of historical network information in a content centric network, wherein the requesting entity is an application associated with a stack with which the collector component is not associated, in accordance with an embodiment of the present invention.

[0032] FIG. **3**B illustrates an exemplary communication in a system which facilitates querying of historical network information in a content centric network, wherein the requesting entity is a stack component of a stack with which the collector component is not associated, in accordance with an embodiment of the present invention.

[0033] FIG. **3**C illustrates an exemplary communication in a system which facilitates querying of historical network information in a content centric network, wherein the requesting entity is another network element or node, in accordance with an embodiment of the present invention.

[0034] FIG. **3**D illustrates an exemplary communication in a system which facilitates querying of historical network information in a content centric network, wherein the requesting entity is an application associated with the same stack with which the collector component is associated, in accordance with an embodiment of the present invention.

[0035] FIG. **3**E illustrates an exemplary communication in a system which facilitates querying of historical network information in a content centric network, wherein the requesting entity is a stack component of the same stack with which the collector component is associated, in accordance with an embodiment of the present invention.

[0036] FIG. **4** presents a flow chart illustrating a method by a requesting entity for facilitating querying of historical network information in a content centric network, in accordance with an embodiment of the present invention.

[0037] FIG. **5** presents a flow chart illustrating a method by a responding entity or a collector component for facilitating querying of historical network information in a content centric network, in accordance with an embodiment of the present invention.

[0038] FIG. **6** illustrates an exemplary computer system that facilitates querying of historical network information in a content centric network, in accordance with an embodiment of the present invention.

[0039] In the figures, like reference numerals refer to the same figure elements.

DETAILED DESCRIPTION

[0040] The following description is presented to enable any person skilled in the art to make and use the embodiments, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present disclosure. Thus, the present invention is not limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

Overview

[0041] Embodiments of the present invention provide a system for querying historical network information in a CCN which facilitates a system user to perform a function that increases overall network efficiency. One aspect of network efficiency is flow and congestion control. Some transport protocols implement this control by maintaining a sliding window of a size "w" that changes based on network conditions. The window size w is dynamically changed based on the perceived and measured performance of the network. For example, if the protocol determines congestion, e.g., due to heavy traffic of neighboring nodes, a consumer can decrease w so that fewer messages are sent to the network. Similarly, if the protocol determines that the network is not congested, the consumer can increase w so that more messages can be sent for better throughput and latency performance.

[0042] In a CCN transport protocol (e.g., ICP, CCTCP and other variants), a similar TCP-like mechanism is used to control flow and congestion by maintaining a window of outstanding interests. Recall that traffic in a CCN is symmetric, where a single interest returns a corresponding content object (or an interest return, as described in U.S. patent application Ser. No. 14/334,530). Thus, historical information regarding a given CCN namespace (e.g., a name prefix) may be collected by a network entity through which CCN packets (e.g., interests and content objects) flow. Embodiments of the present invention provide a system and protocol for a system user to query the network for historical network information collected by a generic "collector component." The historical information relates to outgoing interests and incoming content objects, and to incoming interests and outgoing content objects. The collector component, as the responding network entity, can reside in an application, a single or shared stack, a single or shared forwarder, or any node in the network. The system user, as the requesting entity, can be an application associated with a first stack, where the collector component resides in or is associated with the first stack, or an application associated with a second stack. The requesting entity can also be a stack component of the first stack, a stack component of the second stack, or any other element or node in the network.

[0043] The query can be in the form of an interest message or a control message. For example, if the requesting entity is an application or a stack component associated with a stack which the collector component neither resides in nor is associated with, the query can take the form of an interest message (as described below in relation to FIGS. **3A-3C**). If the requesting entity is an application or a stack component of the same stack that the collector component resides in or is associated with, the query can take the form of a control message sent to the forwarder for the local host device (as described below in relation to FIGS. **3D-3E**).

[0044] The system can collect historical information for each namespace identified in a set of messages. A namespace is a CCN prefix (i.e., one or more contiguous name components beginning from the most general level) and a CCN name may have multiple namespaces or prefixes. For example, the name "/a/b/c" has three namespaces: "/a"; "/a/b"; and "/a/b/c."

[0045] In CCN, each piece of content is individually named, and each piece of data is bound to a unique name that distinguishes the data from any other piece of data, such as other versions of the same data or data from other sources. This unique name allows a network device to request the data by disseminating a request or an interest that indicates the unique name, and can obtain the data independent from the data's storage location, network location, application, and means of transportation. The following terms are used to describe the CCN architecture:

[0046] Content Object (or "content object"): A single piece of named data, which is bound to a unique name. Content Objects are "persistent," which means that a Content Object can move around within a computing device, or across different computing devices, but does not change. If any component of the Content Object changes, the entity that made the change creates a new Content Object that includes the updated content, and binds the new Content Object to a new unique name.

[0047] Unique Names: A name in a CCN is typically location independent and uniquely identifies a Content Object. A data-forwarding device can use the name or name prefix to forward a packet toward a network node that generates or stores the Content Object, regardless of a network address or physical location for the Content Object. In some embodiments, the name may be a hierarchically structured variable-length identifier (HSVLI). The HSVLI can be divided into several hierarchical components, which can be structured in various ways. For example, the individual name components parc, home, ccn, and test.txt can be structured in a left-oriented prefix-major fashion to form the name "/parc/home/ccn/test.txt." Thus, the name "/parc/home/ccn" can be a "parent" or "prefix" of "/parc/home/ccn/test.txt." Additional components can be used to distinguish between different versions of the content item, such as a collaborative document. The HSVLI can also include contiguous name components ordered from a most general level to a most specific level.

[0048] In some embodiments, the name can include an identifier, such as a hash value that is derived from the Content Object's data (e.g., a checksum value) and/or from elements of the Content Object's name. A description of a hash-based name is described in U.S. patent application Ser. No. 13/847,814, which is herein incorporated by reference. A name can also be a flat label. Hereinafter, "name" is used to refer to any name for a piece of data in a name-data network, such as a hierarchical name or name prefix, a flat name, a fixed-length name, an arbitrary-length name, or a label (e.g., a Multiprotocol Label Switching (MPLS) label).

[0049] Interest (or "interest"): A packet that indicates a request for a piece of data, and includes a name (or a name prefix) for the piece of data. A data consumer can disseminate a request or Interest across an information-centric network, which CCN/NDN routers can propagate toward a storage device (e.g., a cache server) or a data producer that can provide the requested data to satisfy the request or Interest.

[0050] The methods disclosed herein are not limited to CCN networks and are applicable to other architectures as well. A description of a CCN architecture is described in U.S. patent application Ser. No. 12/338,175, which is herein incorporated by reference.

Exemplary Network and Communication

[0051] FIG. **1** illustrates an exemplary environment **100** which facilitates querying of historical network information in a content centric network, in accordance with an embodiment of the present invention. Computing environment **100** can include a computer network **102**, such as a CCN. Environment **100** can also include a user **106** associated with a local computing device **104**, and a remote computing device **108**. Devices **104** and **108** can have internal transport stacks (e.g., associated with transport frameworks **130** and **170**, respectively) that exchange network packets with each other over network **102**. Network packets can include interest packets and content object packets.

[0052] In a traditional IP architecture, a forwarder is an IP-based forwarder that looks at the header of a packet to determine the source and the destination for the packet, and forwards the packet to the destination. The stack performs TCP/UDP, and an application interacts with the stack via a socket. In contrast, device **104** of the present invention does not use a conventional "stack." Rather, device **104** via an application **110** can request a portal API instance corresponding to a portal **120** which corresponds to transport framework **130**. Similarly, device **108** via an application **150** can request a portal API instance corresponding to a portal **160** which corresponds to transport framework **170**.

[0053] Device **104** can include any computing device coupled to network **102**, such as a smartphone **104.1**, a tablet computer **104.2**, and/or a server or personal computer **104.***m*. Specifically, device **104** can include application **110** which communicates via portal **120** with transport framework **130**. Transport framework **130** can include stack components **134.1-134.***n*. Device **104** can also include forwarder **140** (e.g., a network interface card, or a router in a local area network) which can transfer packets between a stack (and individual stack components) of transport framework **130** and network **102**. Similarly, device **108** can include any computing device coupled to network **102**, such as a server or an end host device. Device **108** can include application **150** which communicates via portal **160** with transport framework **170**. Transport framework **170** can include stack components **174.1-174.***p*. Device **108** can also include a forwarder **180** which can transfer packets between a stack (and individual stack components) of transport framework **170** and network **102**. Forwarders **140** and **180**

can also facilitate the transfer of packets directly between individual stack components **134**.**1**-**134**.*n* and **174**.**1**-**174**.*p*, respectively.

Exemplary Transport Frameworks

[0054] In embodiments of the present invention, the collector component can be implemented in a CCN transport framework, and can reside in a forwarder (as in FIGS. **2**A, **2**B, and **2**E), in a stack (as in FIG. **2**C), or in an application (as in FIG. **2**D). FIG. **2**A illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a single forwarder, in accordance with an embodiment of the present invention. Applications **210** and **250** can reside on the same device or on difference devices which communicate via a network **202**. Application **210** can use APIs **212**, **214**, and **216** to communicate over network **202**, and APIs **212**-**216** can interact via a portal **220** with a transport framework **230**. Transport framework **230** can include one or more transport stacks which each include multiple stack components or communication modules. In FIG. **2**A, transport framework **230** depicts one transport stack (e.g., a transport stack **231**) which includes stack components **232**, **234**, **236**, and **238**. An API adapter **232** can communicate between an API and a specific transport stack and transport framework **230**. A flow controller **234** can shape and manage traffic, pipeline and transmit interests, and order content objects. A verifier/signer **236** can encode and sign content objects destined for a network element, decode and verify content objects destined for the application, encode interests destined for a network element, and decode interests destined for the application. A forwarder/adapter **238** can communicate with a forwarder **240**. Forwarder **240** can communicate with other forwarders over network **202**. A collector component **242** can reside inside forwarder **240** (or inside forwarder **280**, not shown). Other stack components (not shown) can include functionality related to security (e.g., encryption, decryption, authentication, data signing, signature verification, trust assessment, and filtering), data-processing (e.g., encoding, decoding, encapsulating, decapsulating, transcoding, compression, extraction, and decompression), and storage (e.g., data storage, data retrieval from storage, deduplication, segmentation, and versioning).

[0055] Similarly, application **250** can use APIs **252**, **254**, and **256** to communicate over network **202**, and APIs **252**-**256** can interact via a portal **260** with a transport framework **270**. Transport framework **270** can include one or more transport stacks which each include multiple stack components or communication modules. In FIG. **2**A, transport framework **270** depicts one transport stack (e.g., a transport stack **271**) which includes the following stack components: an API adapter **272**; a flow controller **274**; a verifier/signer **276**; and a forwarder/adapter **278** which can communicate with a forwarder **280**. Forwarder **280** can communicate with forwarder **240** over network **202**. Application **210** can be associated with a consumer or a client computing device, and application **250** can be associated with a producer or a content producing device.

[0056] During operation, collector **242** residing in forwarder **240** can monitor a plurality of packets which are outgoing interests and incoming content objects. For example, application **210** can generate and send an interest **290** with a name of "/a/b/c," via portal instance **220** through stack **231**. As interest **290** leaves stack **231**, it passes through forwarder **240** and collector **242**. Collector **242** can monitor the time at which interest **290** is transmitted. Interest **290** can then travel over network **202**, and through, e.g., forwarder **280** to be satisfied by application **250** associated with stack **271**. Application **250** can generate a responsive content object **292** with a name of "/a/b/c" and a payload of "Data." Content object **293** can travel via forwarder **280** to forwarder **240** over network **202**. Collector **242** can note the time that it receives responsive incoming content object **292**, and record in a storage device **242**.**1** the round trip time associated with the multiple namespaces included in the name "/a/b/c" (i.e., "/a," "/a/b," and "/a/b/c"). Collector **242** can also store in storage device **242**.**1** other historical information associated with a given namespace, as described below in the section entitled "Exemplary Historical Information." Storage device **242**.**1** can be accessed solely by collector **242** or shared with other components or elements.

[0057] Collector **242** can also monitor an incoming interest **294** (with a name of "/d/e/f" sent by application **250** via forwarder **280**) by monitoring the time at which interest **294** is received. Collector **242** can subsequently monitor and record the time that an outgoing responsive content object **296** is transmitted, where content object **296** has a name of "/d/e/f" and is sent by application **210** via forwarder **240**. Collector **242** can also monitor and record the round trip time associated with the multiple namespaces included in the name "/d/e/f" (i.e., "/d," "/d/e," and "/d/e/f") as well as other historical information.

[0058] Thus, collector **242** can obtain and store various historical information related to a given namespace. Any requesting entity (e.g., a user of the system) can subsequently query the component for the historical information. A requesting entity can be: an application associated with a first stack, where the collector component resides in the first stack (e.g., application **210**); an application associated with a second stack that is different from the first stack (e.g., application **250**); a stack component of the first stack, wherein the stack component is different from the collector component (e.g., flow controller **234**); a stack component of the second stack (e.g., flow controller **274**); and any other element or node in the network (not shown).

[0059] FIG. **2**B illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a shared forwarder, in accordance with an embodiment of the present invention. The framework in FIG. **2**B corresponds to the framework in FIG. **2**A, with the difference being that applications **210** and **250**, and stacks **231** and **271**, respectively, are both associated with forwarder **240**. In FIG. **2**B, collector **242** resides in forwarder **240** and can thus monitor all traffic that passes through forwarder **240**. Forwarder **240** is shared by applications **210** and **250**, which can reside on the same device. Collector **242** can monitor packets transmitted to and received from network **202** in a similar fashion as described above in FIG. **2**A. For example, collector **242** can monitor outgoing interests transmitted from application **210** through stack **231** via network **202** to another network node (not shown) as well as incoming responsive content objects received via network **202**. Collector **242** can also monitor incoming interests transmitted to application **250** through stack **271** via network **202** as well as outgoing responsive content objects sent via network **202**. Collector **242** can store historical information collected and associated with the monitored packets (e.g., in storage device **242.1**, not shown in FIG. **2**B).

[0060] FIG. **2**C illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a single transport stack of the transport framework, in accordance with an embodiment of the present invention. The framework in FIG. **2**C corresponds to the framework in FIG. **2**A, with the difference being that collector **242** is a stack component that resides inside stack **231**. A requesting entity can submit a query for historical information associated with a given namespace to collector **242**. A transport stack name scheme, including submitting a query directly to a stack component, is described in U.S. patent application Ser. No. 14/746,490. Collector **242** can store historical information collected and associated with the monitored packets (e.g., in storage device **242.1**, not shown in FIG. **2**C).

[0061] FIG. **2**D illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in an application associated with the transport framework, in accordance with an embodiment of the present invention. The framework in FIG. **2**D corresponds to the framework in FIG. **2**A, with the difference being that collector **242** resides in application **210**. Again, collector **242** can store historical information collected and associated with the monitored packets (e.g., in storage device **242.1**, not shown in FIG. **2**D).

Local Message Stream Co-Processor Example

[0062] FIG. **2**E illustrates an exemplary transport framework which facilitates querying of historical network information in a content centric network, wherein a collector component resides in a forwarder as a message stream co-processor element, in accordance with an embodiment of the present invention. The framework in FIG. **2**E corresponds to the framework in FIG. **2**C, and illustrates an embodiment in which the collector component is a local message stream co-processor. Recall that CCN end-hosts have a single forwarder that services all ingress and egress interest and content objects to and from applications (i.e., incoming interests and corresponding outgoing content objects, and outgoing interests and corresponding incoming content objects). A standard CCN forwarder maintains only a minimal amount of information to forward CCN messages: a forwarding information base ("FIB"), a pending interest table ("PIT"), and an optional content store ("CS" or cache). In this embodiment, forwarder **240** also includes a specific collector component which is a message stream co-processor ("MSCP") **244**. The functionality of MSCP **244** is unique to the end-host that it services and may be configured at startup or at runtime. For example, MSCP **244** may be configured to collect and store only historical information regarding interest and content object exchanges, which can include the average number of interests issued for a specific namespace or prefix for a given period of time. Other types of historical information can also be collected and stored, as described herein (e.g., in storage device **242.1**, not shown in FIG. **2**E).

[0063] Because forwarder **240**, and thus MSCP **244**, processes all messages for all applications on a given system (e.g., a CCN end-host), maintaining the privacy of the messages is a key feature. The operating system can define and limit the functionality of MSCP **244**, including the disclosure of collected historical information to authorized entities only. For example, requesting entity application **210** can transmit a query **286** to MSCP **244**, and requesting entity flow controller **234** can also transmit a query **282** to MSCP **244**. Queries **282** and **286** may include a request for historical information collected by MSCP **244**. Since the requesting entities (e.g., application **210** and flow controller **234**) are associated with the end host serviced by MSCP **244**, MSCP **244** can identify the requesting entities as authorized entities and transmit the requested historical information back in response to queries **282** and **286**. Similarly, application **250** and flow controller **274** may, respectively, submit queries **288** and **284** to MSCP **244**, and in response receive the requested historical information from MSCP **244**.

[0064] Thus, MSCP **244** can perform like a black box that consumes CCN messages for processing. The limits of this processing are unbounded. MSCP **244** can provide whatever is needed for its given end-host and associated applications. For example, MSCP **244** may be configured to count the number of processed messages, to log the names of all outgoing interest messages to a system log, or to collect any of the types of historical information described herein.

Exemplary Historical Information

[0065] Consider the following sequence of n interest messages issued by different applications on the same end-host, i.e., I_j for j=1, . . . , n:

$$
\begin{gathered}
I_1 = /a/b1/c1 \\
I_2 = /a/b1/c2 \\
I_3 = /a/b1/c3 \\
\dots \\
I_{i-1} = /a/b1/ci \\
I_i = /a/b2/c1 \\
\dots \\
I_n = /a/b2/fileN
\end{gathered}
$$

[0066] In a window that includes each of interests I_j, the "/a" namespace has n interests, the "/a/b1" namespace has i interests, and the "/a/b2" namespace has (n-i+1) interests. Each of interests I_j has a corresponding content object response, C_j.

[0067] For each of interests I_j, the collector component can collect various types of historical information based on each particular namespace, including the items in the following non-exhaustive list:

[0068] 1) A round trip time ("RTT") that begins when an outgoing/incoming interest is transmitted and ends when a corresponding incoming/outgoing content object is received;

[0069] 2) A number of outgoing/incoming interests for which a corresponding incoming/outgoing content object has not been received (e.g., outstanding window size);

[0070] 3) A number of outgoing/incoming interests for which a corresponding incoming/outgoing content object is received based on a predetermined amount of time or an RTT;

[0071] 4) A number of bytes correctly retrieved based on a predetermined amount of time or an RTT;

[0072] 5) A number of outgoing/incoming interests that time out based on a predetermined amount of time or an RTT;

[0073] 6) A number of outgoing/incoming interests which are retransmitted based on a predetermined amount of time or an RTT;

[0074] 7) A number of re-transmitted outgoing/incoming interests that time out based on a predetermined amount of time or an RTT;

[0075] 8) A number of interest return messages received/transmitted based on a predetermined amount of time or an RTT, where an interest return message is identified based on a code indicated in the message;

[0076] 9) A number of outgoing/incoming interests aggregated based on a predetermined amount of time or an RTT;

[0077] 10) A number of active upstream paths identified for a given time;

[0078] 11) A strategy for forwarding or processing packets;

[0079] 12) A number of transmitted original interests, where an original interest is not a re-transmitted interest, wherein the number of original interests include names that share one or more name prefixes ("correlated") and names that do not share any name prefixes ("uncorrelated"); and

[0080] 13) A number of active entries in a forwarding information base, where the number of entries include correlated and uncorrelated entries.

[0081] In addition, the collector component can perform a function or compute information based on the historical information. Examples include computation of the average RTT for a given namespace, an estimate of the transmission window size, and a user-defined function. In estimating the average RTT for a given namespace, consider the n interest messages issued by different applications on the same end-host, i.e., I_j for j=1, . . . , n. Recall that each of interests I_j has a corresponding content object response, C_j. Let r_j be the RTT of the issuance of the interest I_j and the retrieval of the corresponding content object C_j, and assume that historical information is maintained for windows of size d<i and d<(n-i+1). Then, for each of the three namespaces (i.e., "/a," "/a/b1," and "/a/b2"), the collector component can compute a smoothed RTT average r[/a], r[/a/b1], and r[/a/b2]. These RTT averages can be computed based on any appropriate algorithm, e.g., as a weighted moving average or exponential moving average. Additionally, while an RTT computation may be based on time, it may also be based on other information such as hop counts for retrieved content objects.

[0082] In estimating the transmission window size, recall that the transmission window size is coupled to time in that the window size changes over time, depending on the behavior of the transport protocol. The goal of this historical information is not to anticipate the behavior of the transport protocol, but rather to passively measure the effects of the transport protocol. Consider a small (and configurable) time epoch "E." Given a frame of n interests $I_1, \ldots, I_n$, the collector component computes the number of outstanding interests for each namespace in the frame. The resultant number is treated as a sample in the given timeslot of size E. When the time slot advances, the new samples are computed. For example, in timeslot **t1**, there may be x outstanding interests for namespace "N." Then, in timeslot **t2**, there may be y outstanding interests for namespace N. Thus, the average window size may be approximately computed as (x+y)/2. The accuracy of this approximation depends on both the granularity of the timeslot (e.g., E) and the width of the frame. E may also be dynamically modified based on RTT estimations. For example, if the RTT estimation for a given namespace is small, then E may correspondingly be decreased. The value of E is typically less than the time it takes for the collector component to process all interests in a given frame.

[0083] In computing a user-defined function, the collector component can accept closures for processing streams of interests. The representation of a closure conforms to the interface for the stream processing. That is, the closure is a function which accepts a sequence of interests, produces an output, and may invoke other closures in the process. This can be compared to the "map-reduce" paradigm, where a user provides custom "map" and "reduce" functions and the framework invokes them accordingly. The processing functions for an interest stream may be designed to model the functional paradigm. One example of such a user-defined function is one which estimates the frequency at which interests are used for a given namespace. Another example is a function which estimates the frequency of message failures (e.g., interest returns) for a given namespace.

Exemplary Communication Via a Query that is an Interest Message

[0084] FIG. **3**A illustrates an exemplary communication in a system **300** which facilitates querying of historical network information in a content centric network, wherein the requesting entity is an application associated with a stack with which the collector component is not associated, in accordance with an embodiment of the present invention. System **300** corresponds to the exemplary transport framework depicted and described in relation to FIG. **2**A, where the collector component (e.g., collector **242**) resides in the forwarder (e.g., forwarder **240**). Recall that a requesting entity can be: an application associated with a first stack, where the collector component resides in or is associated with the first stack (e.g., application **210**); an application associated with a second stack that is different from the first stack (e.g., application **250**); a stack component of the first stack, wherein the stack component is different from the collector component (e.g., flow controller **234**); a stack component of the second stack (e.g., flow controller **274**); and any other element or node in the network (not shown). In system **300**, the requesting entity is application **250**, which is an application associated with a stack (e.g., stack **271**) that is different from the first stack (e.g., stack **231**), where the collector component resides in or is associated with the first stack (e.g., collector **242** resides in forwarder **240** and is not associated with stack **271**).

[0085] Application **250** can generate an interest message **302** with a name **302.1** of the following format: "/r_prefix/type=outstanding-window-size/id=<id>/auth=<sig+cert>/<nonce>." The variables in name **302.1** can be defined as follows: "/r_prefix" is a routable name prefix which includes one or more contiguous name components ordered beginning from the most general level (e.g., "/a/b2/fileN" or "/parc/ccn/file1"); "type" indicates the type of command or query (e.g., an outstanding window size which corresponds to a number of outstanding interests for the routable prefix or namespace, a number of interests answered correctly in a specific period of time, and other examples as described above in the section entitled "Exemplary Historical Information"); "id" indicates a user identifier of the requesting entity (e.g., "<id>"); "auth" indicates authentication information of the requesting entity, which can include the signature and/or digital certificate of the requesting entity (e.g., "<sig+cert>"); and "<nonce>" is a randomly generated nonce to ensure both uniqueness and freshness.

[0086] In some embodiments, name **302.1** can include one or more specific namespaces of the routable prefix for which the historical information is requested. Furthermore, the variables defined above and depicted as included in the name for the interest can be included in other fields of the interest. In other words, the variables and information included in name **302.1** can alternatively be indicated in interest **302** in other fields (not shown).

[0087] In addition, name **302.1** can include a string for the query (e.g., "outstanding-window-size"), one or more parameters for the query, and a function for the query. For example, the function for the query can indicate a request for the responding entity (e.g., the collector component) to perform a function or compute information based on the historical information, such as an average RTT for a given namespace, an estimate of the transmission window size for a given namespace, or a user-defined function (as described above in the section entitled "Exemplary Historical Information"). Interest **312** can also include a payload **312.2** with a value of "<data>" (e.g., if the requesting entity needs to provide additional data for the collector component to retrieve specific historical information or to perform a function or other computation).

[0088] Collector **242**, as the responding entity, can receive interest **302**, perform a lookup in storage **242.1** for the queried historical information, and subsequently return a response which can be a content object (not shown) that includes the queried historical information. The responsive content object can travel on a reverse path as interest **302** back to application **250**. If the query includes a command to perform a function, the responsive content object can include the result of the function. The function may be performed by collector **242**, or by another responding entity.

[0089] FIG. **3**B illustrates an exemplary communication in a system **310** which facilitates querying of historical network information in a content centric network, wherein the requesting entity is a stack component of a stack with which the collector component is not associated, in accordance with an embodiment of the present invention. In system **310**, the requesting entity is flow controller **274**, which is a stack component of the second stack (e.g., flow controller **274** is a stack component of stack **271**, which is a stack that is not associated with collector **242** of forwarder **240**). Similar to application **250** of system **300** in FIG. **3**A, flow controller **274** in FIG. **3**B can generate an interest message **312** with a name **312.1** of the following format: "/r_prefix/type=outstanding-window-size/id=<id>/auth=<sig+cert>/<nonce>" (which is the same as the format for interest **302** with name **302.1** in FIG. **3**A). Similar to the communication in FIG. **3**A, collector **242** receives interest **312** and returns a responsive content object on a reverse path as interest **312** back to flow controller **274**.

[0090] FIG. **3**C illustrates an exemplary communication in a system **320** which facilitates querying of historical network information in a content centric network, wherein the requesting entity is another network element or node, in accordance with an embodiment of the present invention. In system **320**, the requesting entity is a network device **324**, which is another element or node in the network. Similar to application **250** of system **300** in FIG. **3**A, network device **324** can generate an interest message **322** with a name **322.1** of the following format: "/r_prefix/type=outstanding-window-size/id=<id>/auth=<sig+cert>/<nonce>" (which is the same as the format for interest **302** with name **302.1** in FIG. **3**A). Similar to the communication in FIG. **3**A, collector **242** receives interest **322** and returns a responsive content object on a reverse path as interest **322** back to device **324**.

Exemplary Communication Via a Query that is a Control Message

[0091] FIG. **3**D illustrates an exemplary communication in a system **330** which facilitates querying of historical network information in a content centric network, wherein the requesting entity is an application associated with the same stack with which the collector component is associated, in accordance with an embodiment of the present invention. In system **330**, the requesting entity is application **210**, which is an application associated with the first stack (e.g., transport stack **231**), where collector **242** is also associated with the first stack (e.g., collector **242** resides in forwarder **240** and is associated with transport stack **231**). Application **210** can generate a control message **332** destined for collector **242** in local forwarder **240**. Control message **332** can have a name **332.1** of the following format: "/localhost/fwder/gcc/cmd=query-outstanding-window-size/namespace=<namespace>/<nonce>." The variables in name **332.1** can be defined as follows: "/localhost" is a the name for the device on which application **210**, stack **231**, and forwarder **240** reside; "/fwder" indicates that the control message is destined for the forwarder on the local device (e.g., forwarder **240**); "/gcc" indicates the name of the specific collector component which resides inside the forwarder (e.g., "/gcc" for "general collector component" or "/mscp" for "message stream co-processor" as described in relation to FIG. **2**E); "cmd" indicates the command or query (e.g., to retrieve the outstanding window size, which corresponds to a number of outstanding interests for an indicated namespace); "namespace" indicates the name prefix on which the command or query is based (e.g., a routable name prefix which includes one or more contiguous name components ordered beginning from the most general level, such as "/a/b2/fileN" or "/parc/ccn/file1"); and "<nonce>" is a randomly generated nonce that can guarantee both uniqueness and freshness. Similar to name **302.1** in FIG. **3**A, name **332.1** in FIG. **3**D can include a string for the query, one or more parameters for the query, and a function for the query.

[0092] Collector **242**, as the responding entity, can receive control message **332**, perform a lookup in storage **242.1** for the queried historical information, and subsequently return a response that includes the queried historical information (and, if the command includes a function, the result of the function). The response, which can be a content object, can travel back on a reverse path as control message **332** back to application **210**.

[0093] FIG. **3**E illustrates an exemplary communication in a system **340** which facilitates querying of historical network information in a content centric network, wherein the requesting entity is a stack component of the same stack with which the collector component is associated, in accordance with an embodiment of the present invention. In system **340**, the requesting entity is flow controller **234**, which is a stack component of the same stack (e.g., stack **231**) with which the collector component is associated (e.g., collector **242** resides in forwarder **240** and is associated with stack **231**). Flow controller **234** can generate a control message **342** destined for collector **242** in local forwarder **240**. Control message **342** can have a name **342.1** of the following format: "/localhost/fwder/gcc/cmd=query-outstanding-window-size/namespace=<namespace>/<nonce>" (which is the same as the format for control message **332** with name **332.1** in FIG. **3**D). Similar to the communication in FIG. **3**D, collector **242** receives control message **342** and returns a response on a reverse path as control message **342** back to flow controller **234**.

Requesting Entity Queries for Historical Information

[0094] FIG. **4** presents a flow chart **400** illustrating a method by a requesting entity for facilitating querying of historical network information in a content centric network, in accordance with an embodiment of the present invention. During operation, a requesting entity (such as a stack, a stack component, or an application in a CCN end-host device) generates a query for historical information associated with interest and content object packets (operation **402**). A name for an interest packet is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, and the query is based on a name prefix that includes one or more contiguous name components. For example, the query may be for an outstanding window size for a given name prefix or namespace over a certain period of time (e.g., the number of outgoing interests for which a corresponding incoming content object has not been received for the name prefix "/a/b1"). The query can be in the form of an interest message or a control message, as described above in relation to FIGS. **3**A-**3**E.

[0095] The requesting entity transmits the query to a responding entity (operation **404**). The requesting entity determines whether it receives the queried historical information (decision **406**). If it does not, the operation returns. If the requesting entity does receive the queried historical information, the requesting entity performs an operation that increases network efficiency (operation **408**). For example, if the historical information indicates that the outstanding window size is small (implying that the flow is not congested), the requesting entity may increase its rate of transmission of interests for that namespace. Similarly, if the historical information indicates that the outstanding window size is large (implying congestion), the requesting entity may decrease its rate of interest transmission for that namespace. The requesting entity can perform other operations, such as setting or changing: a window size; a rate of transmission for original (e.g., first or initially transmitted) interests; and a rate of transmission for re-transmitted interests.

Responding Entity Responds to Query for Historical Information

[0096] FIG. **5** presents a flow chart **500** illustrating a method by a responding entity or a collector component for facilitating querying of historical network information in a content centric network, in accordance with an embodiment of the present invention. During operation, the responding entity receives from the requesting entity a query for historical information associated with interest and content object packets (operation **502**). The responding entity can be the collector component, and the requesting entity can be a stack, a stack component, or an application in a CCN end-host device. A name for an interest packet is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, and the query is based on a name prefix that includes one or more contiguous name components. The query can be in the form of an interest message or a control message, as described above in relation to FIGS. **3**A-**3**E. If the query is in the form of an interest message, the query can include a user identifier and authentication information (e.g., a signature or digital certificate) of the requesting entity. The collector component determines whether the requesting entity is authenticated based on the information included in the query (decision **504**). Note that if the query is in the form of a control message and is transmitted by an application or a stack component associated with the same stack that the collector component either resides in or is associated with, the collector component can authenticate the query by determining that the requesting entity is a component or entity associated with the same local stack.

[0097] If the requesting entity is not successfully authenticated, the operation returns. If the requesting entity is successfully authenticated, the collector component determines the queried historical information (operation **506**). The collector component can obtain the queried historical information from a local storage cache or another accessible storage medium. The collector component determines whether a function is included in the query (decision **508**). If a function is not included in the query, the collector component transmits the queried historical information to the requesting entity, which causes the requesting entity to perform an operation that increases network efficiency (operation **512**).

[0098] If a function is included in the query, the collector component performs the function included in the query based on the historical information (operation **510**). For example, the function can be for an average RTT or an estimate of the transmission window size for a given namespace. The function can also be a user-defined function. These examples are described above in the section entitled "Exemplary Historical Information." Upon performing the function, the collector component transmits the result of the function performed on the queried historical information to the requesting entity, which causes the requesting entity to perform an operation to increase network efficiency (operation **512**). Depending on the identity and authority of the requesting entity, the collector may also sanitize the requested historical information before transmitting it back to the requesting entity.

Exemplary Computer System

[0099] FIG. **6** illustrates an exemplary computer system **602** that facilitates querying of historical network information in a content centric network, in accordance with an embodiment of the present invention. Computer system **602** includes a processor **604**, a memory **606**, and a storage device **608**. Memory **606** can include a volatile memory (e.g., RAM) that serves as a managed memory, and can be used to store one or more memory pools. Furthermore, computer system **602** can be coupled to a display device

610, a keyboard **612**, and a pointing device **614**. Storage device **608** can store an operating system **616**, a content-processing system **618**, and data **630**.

[0100] Content-processing system **618** can include instructions, which when executed by computer system **602**, can cause computer system **602** to perform methods and/or processes described in this disclosure. Specifically, content-processing system **618** may include instructions for sending and/or receiving data packets to/from other network nodes across a computer network, such as a content centric network (communication module **620**). A data packet can include an interest packet or a content object packet with a name that is an HSVLI. Further, content-processing system **618** can include instructions for generating a query for historical information associated with interest and corresponding content object packets (query-generating module **622**). Content-processing system **618** can include instructions for transmitting the query to a responding entity (communication module **620**), and, in response to receiving the historical information from the responding entity (communication module **620**), performing an operation that increases network efficiency based on the historical information (operation-performing module **624**). Content-processing system **618** can further include instructions for setting or changing a window size, setting or changing a rate of transmission for re-transmitted interests, setting or changing a rate of transmission for original interests, wherein an original interest is not a re-transmitted interest, and performing an operation related to increasing the efficiency of the network (operation-performing module **624**).

[0101] Content-processing system **618** can additionally include instructions for receiving a query from a requesting entity for historical information associated with interest and corresponding content object packets (communication module **620**). Content-processing system **618** can include instructions for, in response to authenticating the requesting entity (entity-authenticating module **628**), transmitting the queried historical information (communication module **620**). Content-processing system **618** can include instructions for performing a function on the historical information, including computing an estimate of an average round trip time for an interest and a corresponding content object packet based on the name prefix, computing an estimate of a size of a transmission window for the name prefix, and performing a function defined by the system of a user of the system (function-performing module **626**).

[0102] Data **630** can include any data that is required as input or that is generated as output by the methods and/or processes described in this disclosure. Specifically, data **630** can store at least: a name; a name that is an HSVLI; one or more name components; a name prefix; a namespace; a packet that corresponds to an interest or a content object; a transport framework; a protocol or transport stack; one or more components of a transport or protocol stack; a collector component; a portal instance associated with a transport or protocol stack; historical information (as described above in the section entitled "Exemplary Historical Information"); a round trip time; a predetermined amount of time; a request or query for historical information; a control message; a routable name prefix; a user identifier; authentication information; a type for a query; a string for a query; one or more query parameters; a function for a query; a result of the function for the query; a randomly generated nonce; an identifier for a local forwarder; a window size; a rate of transmission for re-transmitted interests; and a rate of transmission for original interests.

[0103] The data structures and code described in this detailed description are typically stored on a computer-readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. The computer-readable storage medium includes, but is not limited to, volatile memory, non-volatile memory, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs), DVDs (digital versatile discs or digital video discs), or other media capable of storing computer-readable media now known or later developed.

[0104] The methods and processes described in the detailed description section can be embodied as code and/or data, which can be stored in a computer-readable storage medium as described above. When a computer system reads and executes the code and/or data stored on the computer-readable storage medium, the computer system performs the methods and processes embodied as data structures and code and stored within the computer-readable storage medium.

[0105] Furthermore, the methods and processes described above can be included in hardware modules. For example, the hardware modules can include, but are not limited to, application-specific integrated circuit (ASIC) chips, field-programmable gate arrays (FPGAs), and other programmable-logic devices now known or later developed. When the hardware modules are activated, the hardware modules perform the methods and processes included within the hardware modules.

[0106] The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What is claimed is:

1. A computer system for facilitating querying of historical network information, the system comprising:

a processor; and

a storage device storing instructions that when executed by the processor cause the processor to perform a method, the method comprising:

generating a query for historical information associated with interest and corresponding content object packets, wherein a name for an interest packet is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, wherein the query is based on a name prefix that includes one or more contiguous name components;

transmitting the query to a responding entity; and

in response to receiving the historical information from the responding entity, performing an operation that increases network efficiency based on the historical information, thereby facilitating a protocol for querying the historical information to increase network efficiency.

2. The computer system of claim **1**, wherein the query is an interest packet that indicates one or more of:

a routable prefix which includes one or more contiguous name components beginning from the most general level;

a user identifier of the requesting entity;

authentication information of the requesting entity;

a type for the query;

a string for the query;

one or more parameters for the query;

a function for the query; and

a random nonce.

3. The computer system of claim **1**, wherein the query is an interest packet with a name that is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, wherein the name includes one or more of:

a routable prefix which includes one or more contiguous name components beginning from the most general level;

a user identifier of the requesting entity;

authentication information of the requesting entity;

a type for the query;

a string for the query;

one or more parameters for the query;

a function for the query; and

a random nonce.

4. The computer system of claim **1**, wherein the method is performed by a requesting entity which is a component of a stack of communication modules, wherein the responding entity is a local forwarder that services the stack, and wherein the query is a control message that indicates one or more of:

an identifier for the local forwarder;

a component which is responsible for collecting and storing the historical information, wherein the component resides in the local forwarder;

a type for the query;

a string for the query;

one or more parameters for the query; and

a function for the query.

5. The computer system of claim **1**, wherein the operation is one or more of:

setting or changing a window size;

setting or changing a rate of transmission for re-transmitted interests;

setting or changing a rate of transmission for original interests, wherein an original interest is not a re-transmitted interest; and

an operation related to increasing the efficiency of the network.

6. The computer system of claim **1**, wherein the query includes a command to perform a function on the historical information, wherein the function includes one or more of:

computing an estimate of an average round trip time for an interest and a corresponding content object based on the name prefix, wherein a round trip time begins when an interest is transmitted and ends when a corresponding content object is received, or begins when an interest is received and ends when a corresponding content object is transmitted, wherein the estimate is based on a plurality of average round trip times for a corresponding plurality of related namespaces that share at least one name prefix;

computing an estimate of a size of a transmission window based on a number of outstanding interests for the name prefix for a predetermined period of time; and

performing a function based on one or more interests as input, wherein an output for the function is a variable which can be stored by the system, wherein the function is defined by the system or a user of the system.

7. The computer system of claim **1**, wherein the method is performed by a requesting entity which is one or more of:

an application associated with a first stack, wherein the responding entity resides in or is associated with the first stack;

an application associated with a second stack that is different from the first stack;

a stack component of the first stack, wherein the stack component is different from the responding entity;

a stack component of the second stack; and

any other element or node in the network.

8. The computer system of claim **1**, wherein the responding entity resides in one or more of:

an application;

a single stack;

a shared stack;

a single forwarder;

a shared forwarder; and

any node in a network.

9. The computer system of claim **1**, wherein the historical information associated with the packets is one or more of:

a round trip time that begins when an outgoing interest is transmitted and ends when a corresponding incoming content object is received;

a number of outgoing interests for which a corresponding incoming content object has not been received;

a number of outgoing interests for which a corresponding incoming content object is received based on a predetermined amount of time or the round trip time;

a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time;

a number of outgoing interests that time out based on the predetermined amount of time or the round trip time;

a number of outgoing interests which are retransmitted based on the predetermined amount of time or the round trip time;

a number of re-transmitted outgoing interests that time out based on the predetermined amount of time or the round trip time;

a number of interest return messages received based on the predetermined amount of time or the round trip time, wherein an interest return message is received in response to an outgoing interest and is identified based on a code indicated in the message;

a number of outgoing interests aggregated based on the predetermined amount of time or the round trip time;

a number of active upstream paths identified for a given time;

a strategy for forwarding packets;

a first number of transmitted original interests, wherein an original interest is not a re-transmitted interest, and wherein the first number of original interests include names that share one or more name prefixes;

a second number of transmitted original interests, wherein the second number of original interests include names that do not share any name prefixes;

a first number of active entries in a forwarding information base, wherein the first number of entries include names that share one or more name prefixes; and

a second number of active entries in a forwarding information base, wherein the second number of entries include names that do not share any name prefixes.

10. The computer system of claim **1**, wherein the historical information associated with the packets is one or more of:

a round trip time that begins when an incoming interest is received and ends when a corresponding incoming content object is transmitted;

a number of incoming interests for which a corresponding outgoing content object has not been transmitted;

a number of incoming interests for which a corresponding outgoing content object is transmitted based on a predetermined amount of time or the round trip time;

a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time;

a number of incoming interests that time out based on the predetermined amount of time or the round trip time;

a number of re-transmitted incoming interests based on the predetermined amount of time or the round trip time;

a number of re-transmitted incoming interests that time out based on the predetermined amount of time or the round trip time;

a number of interest return messages transmitted based on the predetermined amount of time or the round trip time, wherein an interest return message is transmitted in response to an incoming interest and is identified based on a code indicated in the message; and

a number of incoming interests aggregated based on a predetermined amount of time or a round trip time.

11. A computer system for facilitating querying of historical network information, the system comprising:

a processor; and

a storage device storing instructions that when executed by the processor cause the processor to perform a method, the method comprising:

receiving a query from a requesting entity for historical information associated with interest and corresponding content object packets, wherein a name for an interest packet is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, wherein the query is based on a name prefix that includes one or more contiguous name components;

in response to authenticating the requesting entity, transmitting the queried historical information, which causes the requesting entity to perform an operation to increase network efficiency based on the historical information, thereby facilitating a protocol for querying the historical information to increase network efficiency.

12. The computer system of claim **11**, wherein the query is an interest packet that indicates one or more of:

a routable prefix which includes one or more contiguous name components beginning from the most general level;

a user identifier of the requesting entity;

authentication information of the requesting entity;

a type for the query;

a string for the query;

one or more parameters for the query;

a function for the query; and

a random nonce.

13. The computer system of claim **11**, wherein the query is an interest packet with a name that is a hierarchically structured variable length identifier that includes contiguous name components ordered from a most general level to a most specific level, wherein the name includes one or more of:

a routable prefix which includes one or more contiguous name components beginning from the most general level;

a user identifier of the requesting entity;

authentication information of the requesting entity;

a type for the query;

a string for the query;

one or more parameters for the query;

a function for the query; and

a random nonce.

14. The computer system of claim **11**, wherein the method is performed by a responding entity which is a local forwarder that services a stack of communication modules, wherein the requesting entity is a component of a stack of communication modules, and wherein the query is a control message that indicates one or more of:

an identifier for the local forwarder;

a component which is responsible for collecting and storing the historical information, wherein the component resides in the local forwarder;

a type for the query;

a string for the query;

one or more parameters for the query; and

a function for the query.

15. The computer system of claim **11**, wherein the operation is one or more of:

setting or changing a window size;

setting or changing a rate of transmission for re-transmitted interests;

setting or changing a rate of transmission for original interests, wherein an original interest is not a re-transmitted interest; and

an operation related to increasing the efficiency of the network.

16. The computer system of claim **11**, wherein the query includes a command to perform a function on the historical information, wherein the function includes one or more of:

computing an estimate of an average round trip time for an interest and a corresponding content object based on the name prefix, wherein a round trip time begins when an interest is transmitted and ends when a corresponding content object is received, or begins when an interest is received and ends when a corresponding content object is transmitted, wherein the estimate is based on a plurality of average round trip times for a corresponding plurality of related namespaces that share at least one name prefix;

computing an estimate of a size of a transmission window based on a number of outstanding interests for the name prefix for a predetermined period of time; and

performing a function based on one or more interests as input, wherein an output for the function is a variable which can be stored by the system, wherein the function is defined by the system or a user of the system.

17. The computer system of claim **11**, wherein the requesting entity is one or more of:

an application associated with a first stack, wherein the responding entity resides in or is associated with the first stack;

an application associated with a second stack that is different from the first stack;

a stack component of the first stack, wherein the stack component is different from the responding entity;

a stack component of the second stack; and

any other element or node in the network.

18. The computer system of claim **11**, wherein the method is performed by a responding entity which resides in one or more of:

an application;

a single stack;

a shared stack;

a single forwarder;

a shared forwarder; and

any node in a network.

19. The computer system of claim **11**, wherein the historical information associated with the packets is one or more of:

a round trip time that begins when an outgoing interest is transmitted and ends when a corresponding incoming content object is received;

a number of outgoing interests for which a corresponding incoming content object has not been received;

a number of outgoing interests for which a corresponding incoming content object is received based on a predetermined amount of time or the round trip time;

a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time;

a number of outgoing interests that time out based on the predetermined amount of time or the round trip time;

a number of outgoing interests which are retransmitted based on the predetermined amount of time or the round trip time;

a number of re-transmitted outgoing interests that time out based on the predetermined amount of time or the round trip time;

a number of interest return messages received based on the predetermined amount of time or the round trip time, wherein an interest return message is received in response to an outgoing interest and is identified based on a code indicated in the message;

a number of outgoing interests aggregated based on the predetermined amount of time or the round trip time;

a number of active upstream paths identified for a given time;

a strategy for forwarding packets;

a first number of transmitted original interests, wherein an original interest is not a re-transmitted interest, and wherein the first number of original interests include names that share one or more name prefixes;

a second number of transmitted original interests, wherein the second number of original interests include names that do not share any name prefixes;

a first number of active entries in a forwarding information base, wherein the first number of entries include names that share one or more name prefixes; and

a second number of active entries in a forwarding information base, wherein the second number of entries include names that do not share any name prefixes.

20. The computer system of claim **11**, wherein the historical information associated with the packets is one or more of:

a round trip time that begins when an incoming interest is received and ends when a corresponding incoming content object is transmitted;

a number of incoming interests for which a corresponding outgoing content object has not been transmitted;

a number of incoming interests for which a corresponding outgoing content object is transmitted based on a predetermined amount of time or the round trip time;

a number of bytes correctly retrieved based on the predetermined amount of time or the round trip time;

a number of incoming interests that time out based on the predetermined amount of time or the round trip time;

a number of re-transmitted incoming interests based on the predetermined amount of time or the round trip time;

a number of re-transmitted incoming interests that time out based on the predetermined amount of time or the round trip time;

a number of interest return messages transmitted based on the predetermined amount of time or the round trip time, wherein an interest return message is transmitted in response to an incoming interest and is identified based on a code indicated in the message; and

a number of incoming interests aggregated based on a predetermined amount of time or a round trip time.

* * * * *