



US009270598B1

(12) **United States Patent**  
**Oran et al.**

(10) **Patent No.:** **US 9,270,598 B1**  
(45) **Date of Patent:** **Feb. 23, 2016**

(54) **CONGESTION CONTROL USING CONGESTION PREFIX INFORMATION IN A NAMED DATA NETWORKING ENVIRONMENT**

(71) Applicant: **CISCO TECHNOLOGY, INC.**, San Jose, CA (US)  
(72) Inventors: **David R. Oran**, Cambridge, MA (US); **Ashok Narayanan**, Lexington, MA (US)  
(73) Assignee: **CISCO TECHNOLOGY, INC.**, San Jose, CA (US)  
(\* ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 153 days.

(21) Appl. No.: **14/105,789**  
(22) Filed: **Dec. 13, 2013**

(51) **Int. Cl.**  
**H04L 12/801** (2013.01)  
**H04L 29/08** (2006.01)  
(52) **U.S. Cl.**  
CPC ..... **H04L 47/12** (2013.01); **H04L 47/11** (2013.01); **H04L 47/115** (2013.01); **H04L 67/327** (2013.01)  
(58) **Field of Classification Search**  
None  
See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

2003/0158965	A1 *	8/2003	Koester .....	709/239
2009/0113069	A1 *	4/2009	Prabhakar et al. ....	709/235
2011/0032825	A1 *	2/2011	Minkenberget al. ....	370/238
2014/0023976	A1 *	1/2014	Chian et al. ....	431/12
2015/0163127	A1 *	6/2015	Garcia-Luna-Aceves ....	370/238

**OTHER PUBLICATIONS**

C. Yi, A. Afanasyev, I. Moiseenko, L. Wand, B. Zhang, L. Zhang, A case for Stateful Forwarding Plane, Named Data Networking Technical Report NDN-0002, Jul. 1, 2012, pp. 1-16.\*  
L. Zhang, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos,, T. Abdelzaher, L. Wand, P. Crowley and E. Yeh, Named Data Networking (NDN) Project Technical Report NDN-0001, Oct. 31, 2010, pp. 1-24.\*  
Afanasyev, Alexander, et al., "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, Oct. 5, 2012; 7 pages; http://named-data.net/techreport/TR005-ndnsim.pdf.  
Carofiglio, et al., "Joint Hop-by-hop and Receiver-Driven Interest Control Protocol for Content-Centric Networks," ICN '12 Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, Helsinki, Finland, Aug. 13, 2012; pp. 37-42.  
Cisco Systems, "Cisco IOS Release 12.0(26)S, Class-Based Weighted Fair Queueing and Weighted Random Early Detection," [Retrieved and printed Dec. 6, 2013], 28 pages; http://www.cisco.com/en/US/docs/ios/12\_0s/feature/guide/fswfq26.html.  
"Named Data Networking: Motivation & Details," Named Data Networking, Architecture, [Retrieved and printed Dec. 6, 2013], 9 pages; http://named-data.net/project/archoverview/.

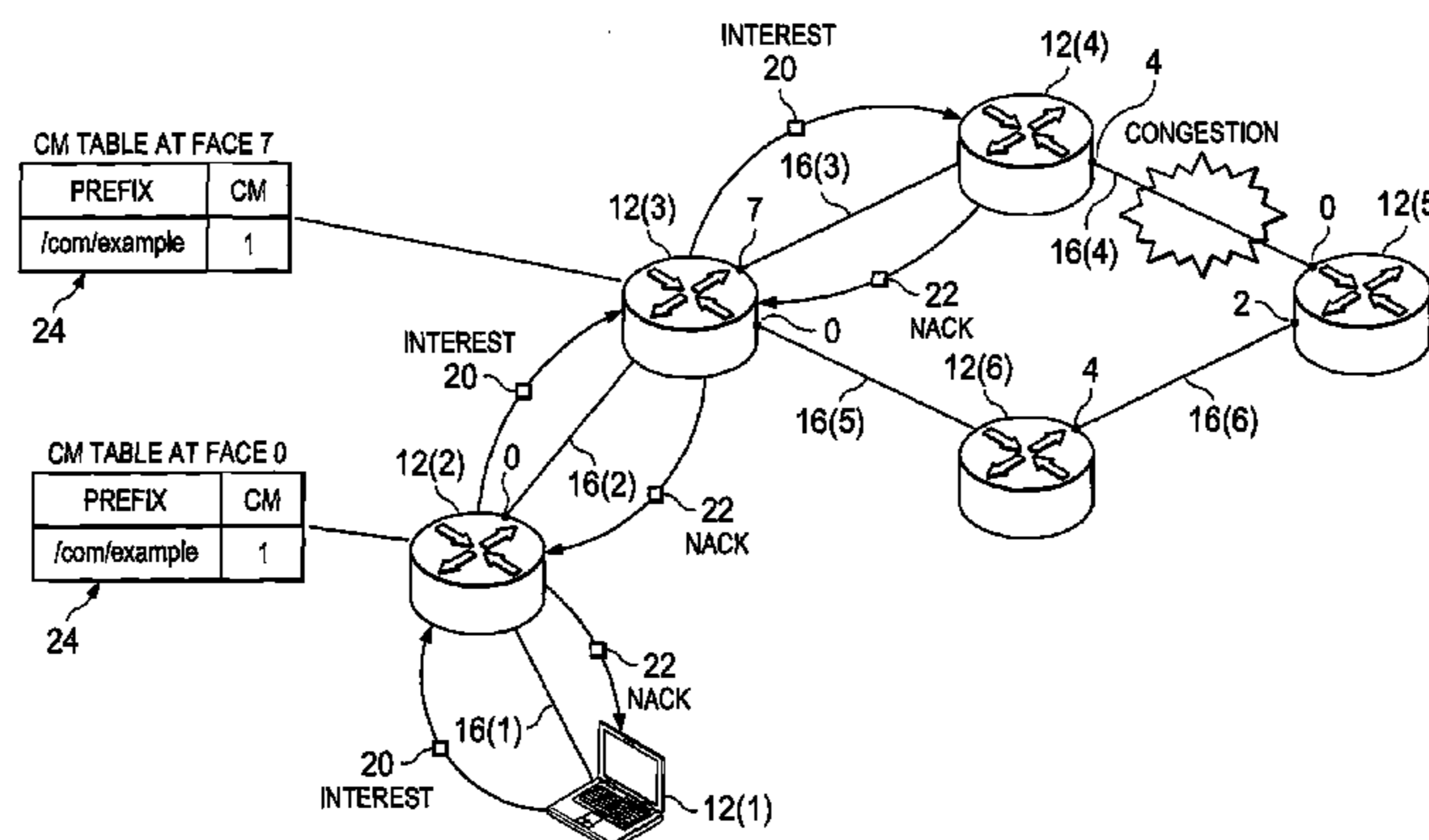
(Continued)

*Primary Examiner* — Christopher Crutchfield  
(74) *Attorney, Agent, or Firm* — Patent Capital Group

(57) **ABSTRACT**

An example method for congestion control using congestion prefix information in a Named Data Networking (NDN) environment is provided and includes sensing, at a first node, congestion preventing an interest packet from being forwarded over a link to a second node, generating a prefix marker associated with a class of traffic to which the interest packet belongs, generating a negative acknowledgement (NACK) packet that includes the prefix marker, the NACK packet being indicative of congestion for any interest packet in the class of traffic indicated by the prefix marker over any path that includes the link, and transmitting the NACK packet over the NDN environment towards a sender of the interest packet.

**16 Claims, 6 Drawing Sheets**



(56)

**References Cited**

OTHER PUBLICATIONS

Van del Pol, Ronald, "D1.3 Named Data Networking Technology Assessment," SARA Computing and Networking Services, Dec. 2011, 5 pages; <https://noc.sara.nl/nrg/publications/RoN-2011-D1.3.pdf>.

Wang, Lan, et al., "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," NDN, Technical Report NDN-0003, Jul. 25, 2012, 15 pages; <http://www.named-data.net/techreport/TR003-OSPFN.pdf>.

Wang, Yi, et al., "Scalable Name Lookup in NDN Using Effective Name Component Encoding," Proceedings, 2012 IEEE 32nd Inter-

national Conference on Distributed Computing Systems, Macau, China, Jun. 18, 2012, 10 pages.

Yi, Cheng, et al., "A Case for Stateful Forwarding Plane," NDN Technical Report NDN-0002 (2012), 16 pages; <http://www.named-data.net/techreport/TR002-forward.pdf>.

Yi, Cheng, et al., "Adaptive Forwarding in Named Data Networking," ACM SIGCOMM Computer Communication Review, vol. 42, No. 3 (Jul. 2012), 6 pages.

Zhang, Lixia, et al., "Named Data Networking (NDN) Project", NDN, Technical Report NDN-0001, Oct. 31, 2010, 26 pages.

\* cited by examiner

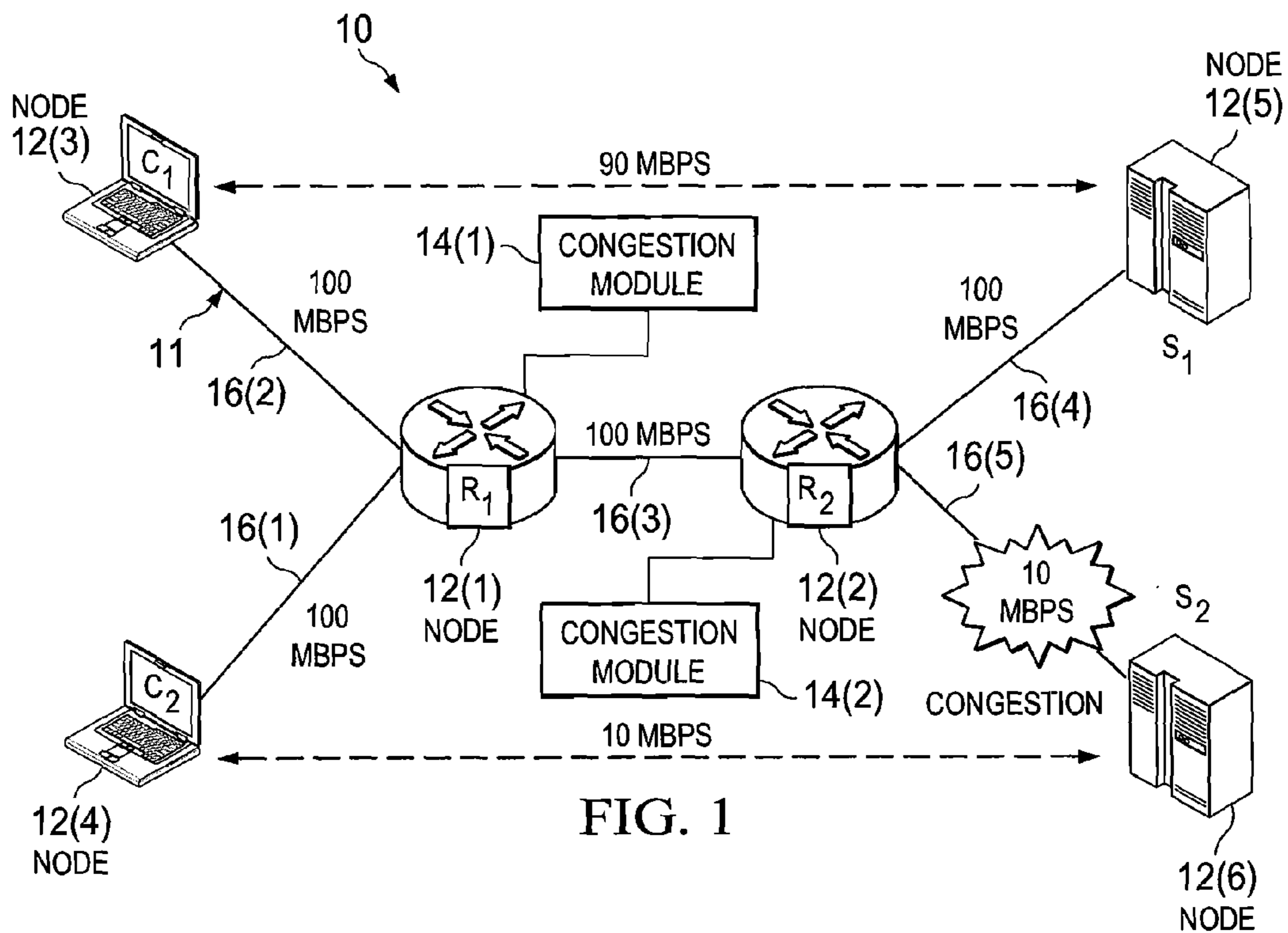


FIG. 1

24

	FIB	CM	
	Pref1	3	28
26	Pref2		
26	Pref3	1	28
26	Pref4		
	Pref5	2	28
	Pref6		

FIG. 3

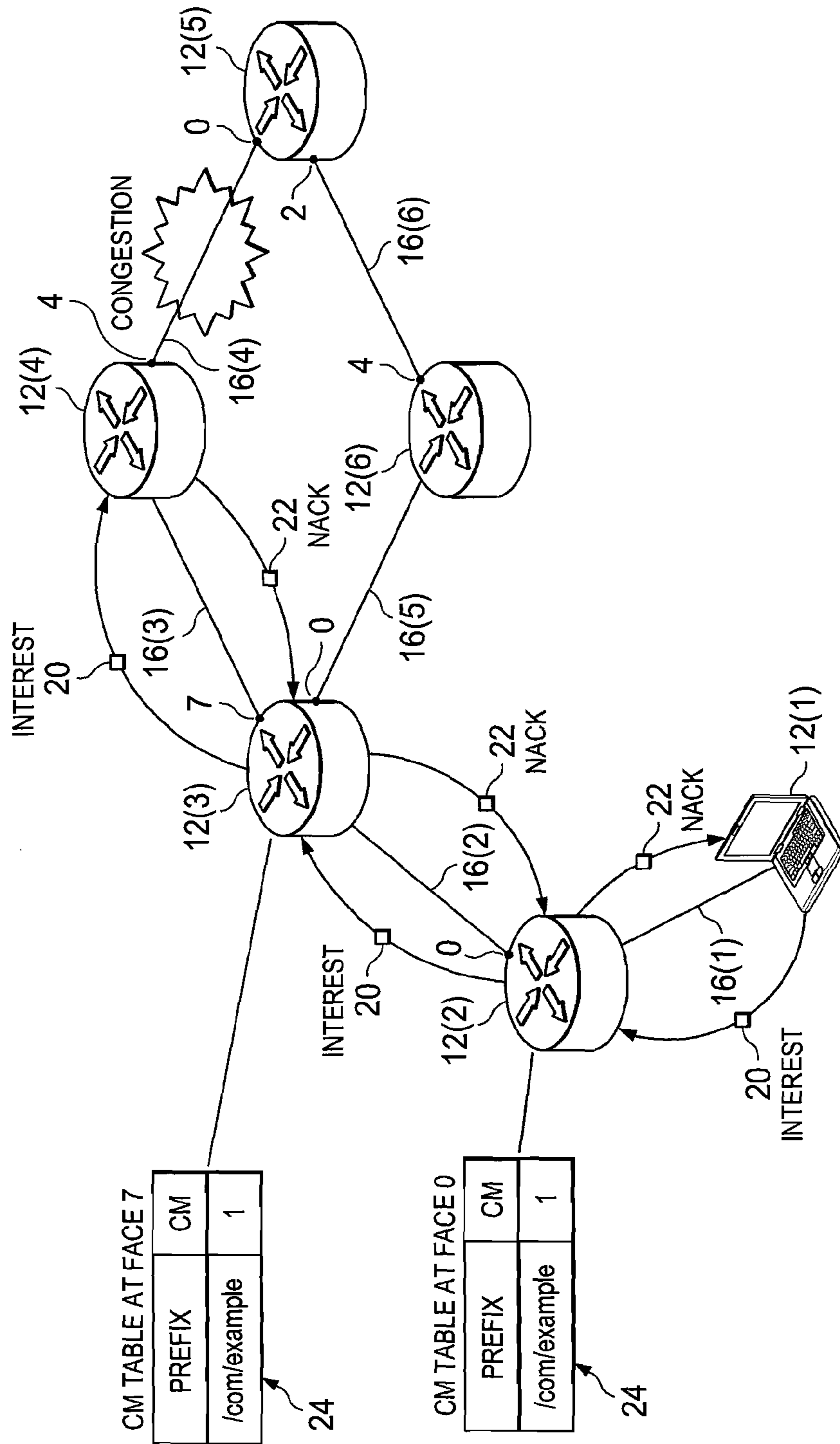


FIG. 2

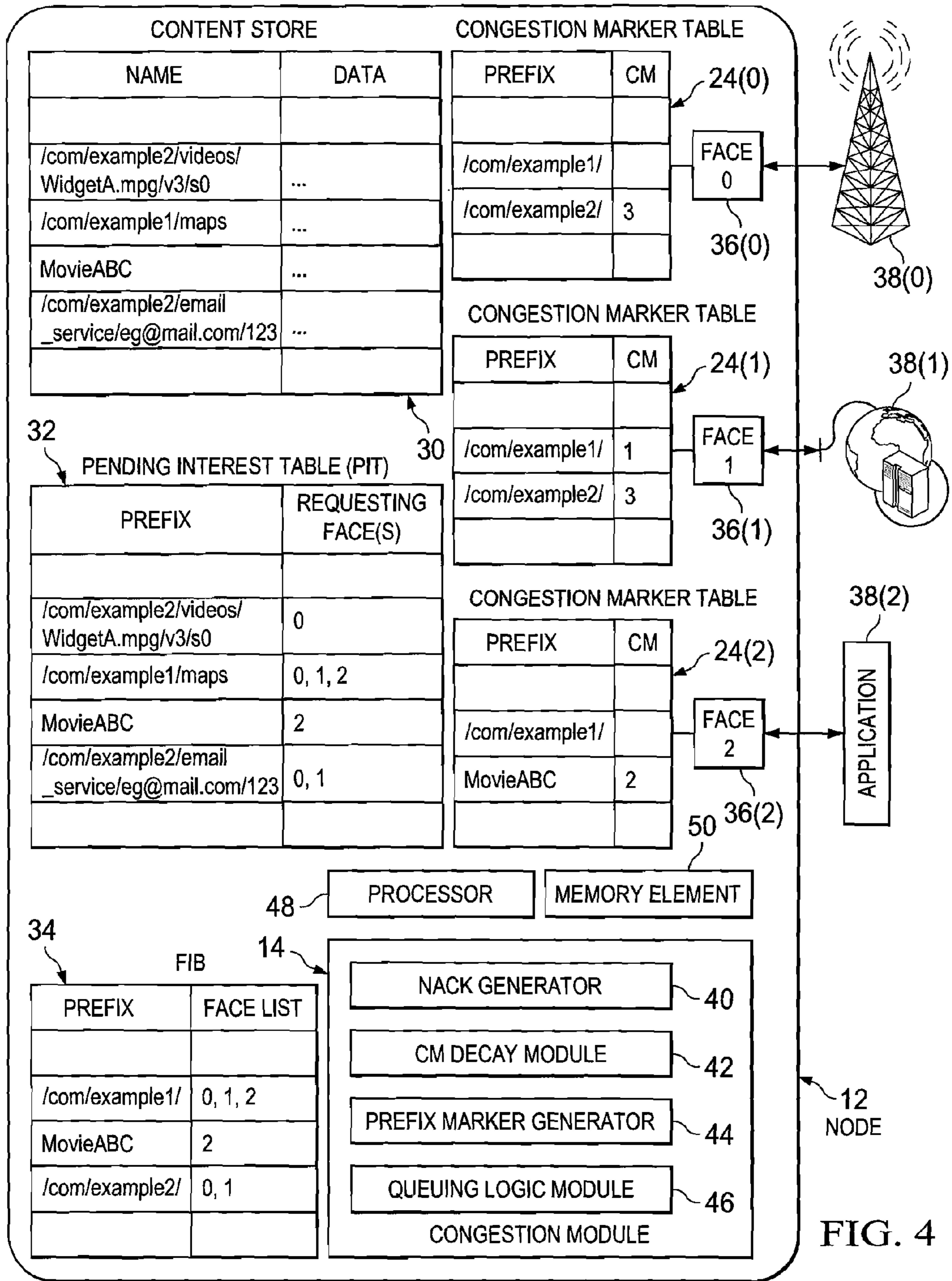


FIG. 4

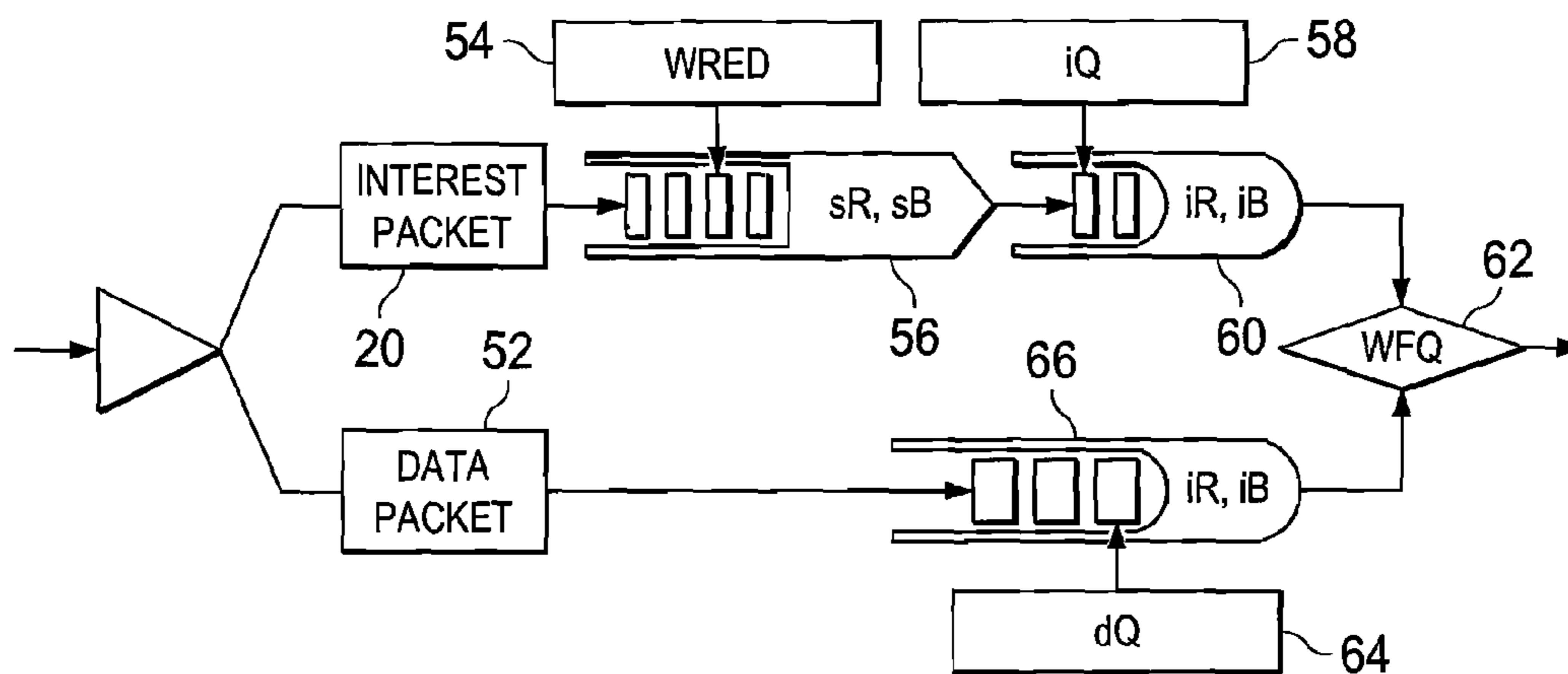


FIG. 5

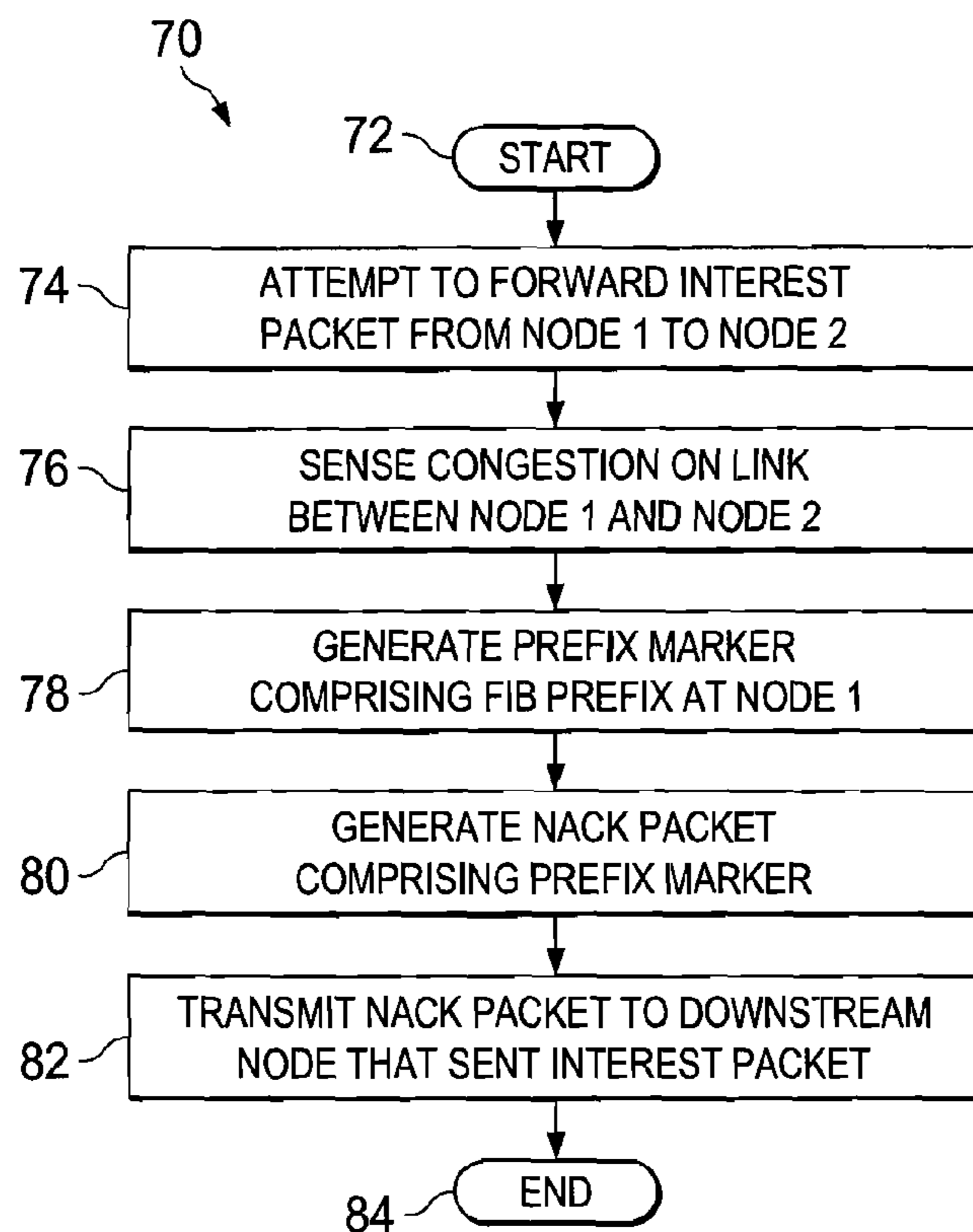


FIG. 6

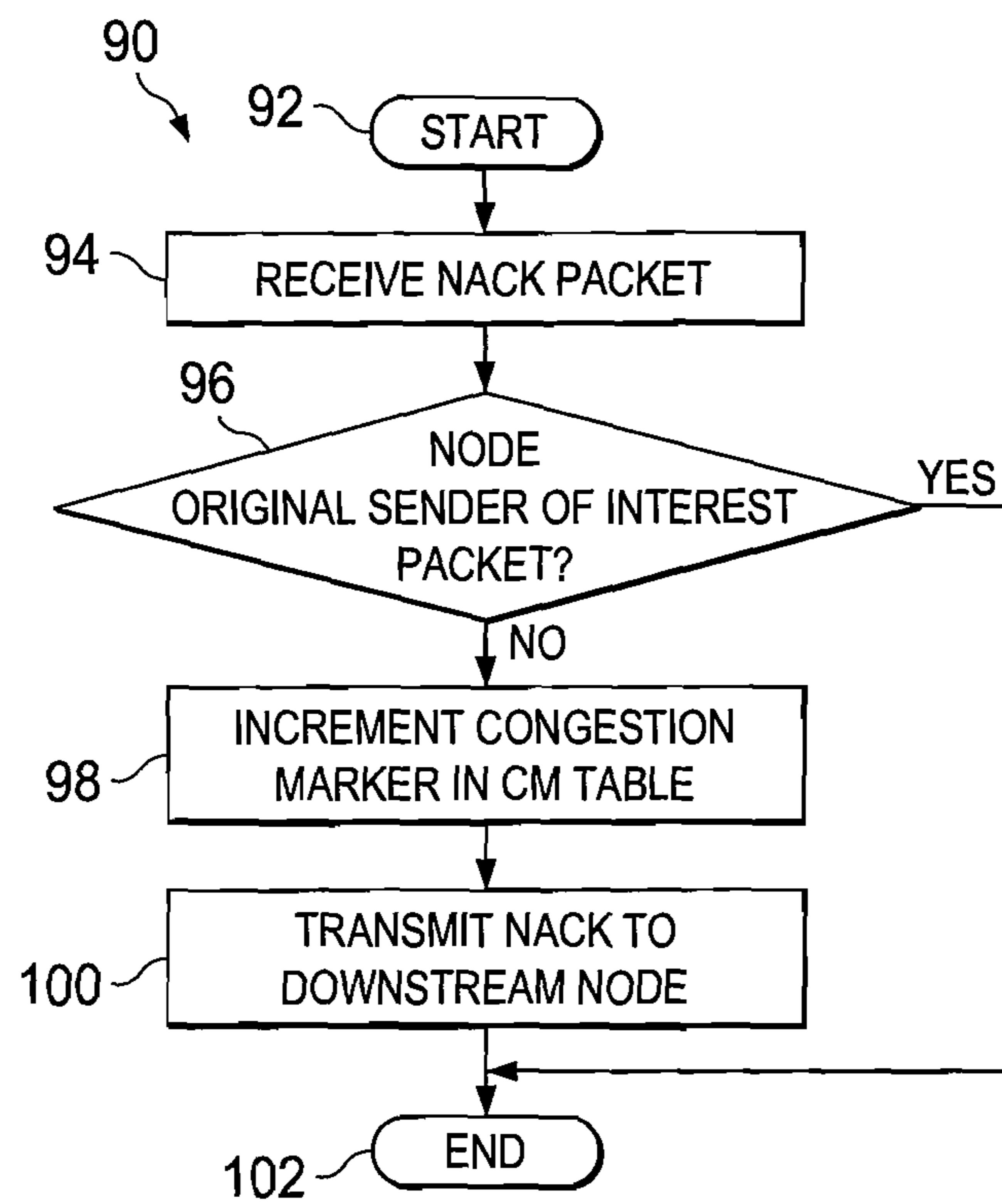


FIG. 7

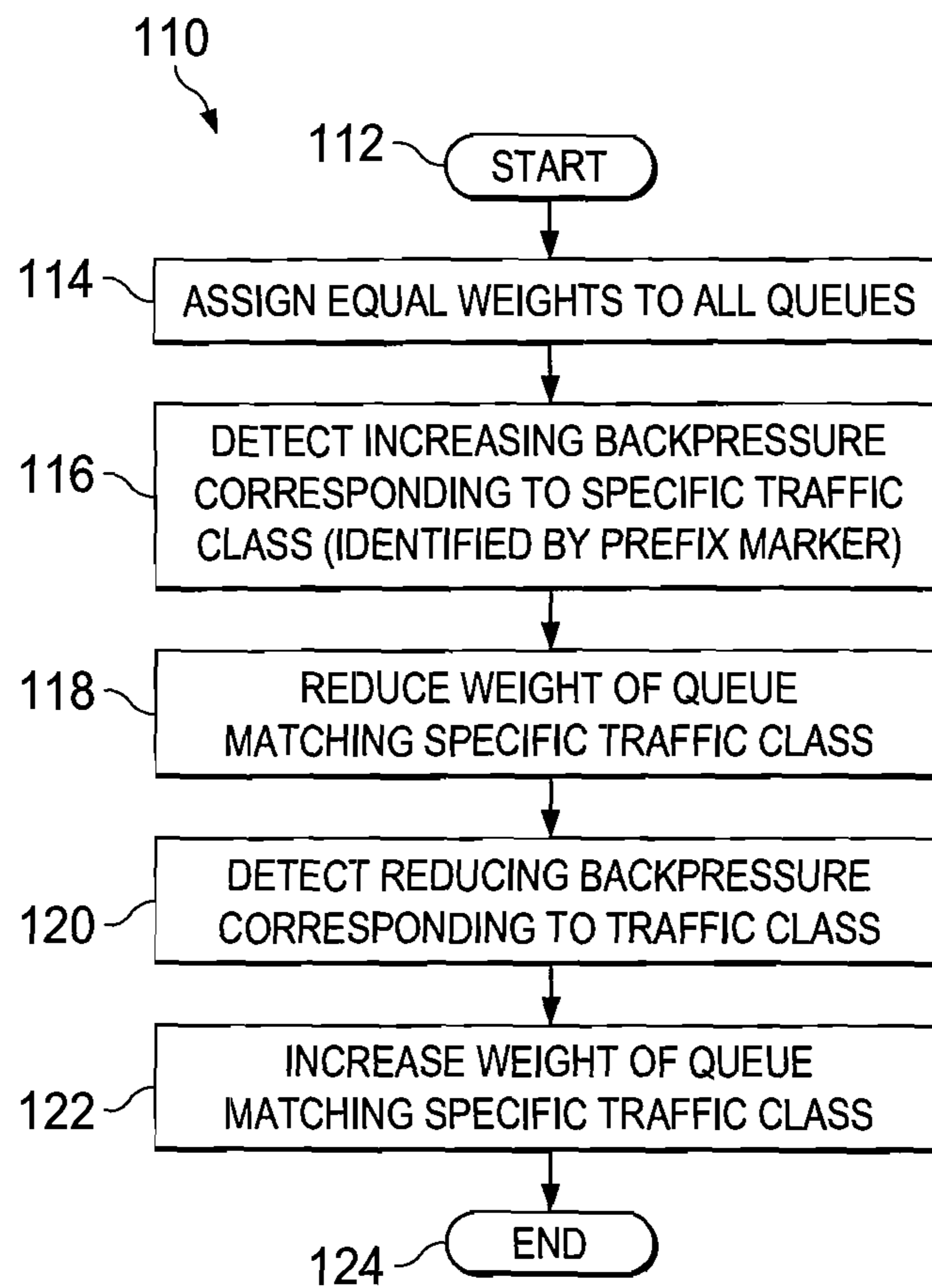


FIG. 8



## 1

**CONGESTION CONTROL USING  
CONGESTION PREFIX INFORMATION IN A  
NAMED DATA NETWORKING  
ENVIRONMENT**

TECHNICAL FIELD

This disclosure relates in general to the field of communications and, more particularly, to congestion control using congestion prefix information in a Named Data Networking (NDN) environment.

BACKGROUND

The Internet was initially designed for point-to-point communication. However, communication modes have dramatically changed since then, particularly with increased use of content distribution. For example, applications are typically written in terms of what information is being used rather than where the information is located; consequently, application specific middleware is used to map between the application's model and the Internet's model. Accordingly, there is a push towards replacing Internet's Internet Protocol (IP) architecture with content oriented networking architecture, such as Named Data Networking (NDN) and Content Centric Networking (CCN).

BRIEF DESCRIPTION OF THE DRAWINGS

To provide a more complete understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

FIG. 1 is a simplified block diagram illustrating a communication system for congestion control using congestion prefix information in a NDN environment;

FIG. 2 is a simplified block diagram illustrating example details of an embodiment of the communication system;

FIG. 3 is a simplified block diagram illustrating yet other example details of an embodiment of the communication system;

FIG. 4 is a simplified block diagram illustrating yet other example details of an embodiment of the communication system;

FIG. 5 is a simplified block diagram illustrating yet other example details of an embodiment of the communication system;

FIG. 6 is a simplified flow diagram illustrating example operations that may be associated with an embodiment of the communication system;

FIG. 7 is a simplified flow diagram illustrating other example operations that may be associated with an embodiment of the communication system; and

FIG. 8 is a simplified flow diagram illustrating yet other example operations that may be associated with an embodiment of the communication system.

DETAILED DESCRIPTION OF EXAMPLE  
EMBODIMENTS

Overview

An example method for congestion control using congestion prefix information in a NDN environment is provided and includes sensing (e.g., detecting, identifying, distinguishing, recognizing, discovering) at a first node, congestion (e.g., persistent link or queue overload) that prevents one (or more)

## 2

interest packet(s) from being forwarded over a link to a second node and generating a prefix marker associated with a class of traffic to which the interest packet belongs. In certain embodiments, the method can also include generating a negative acknowledgement (NACK) packet that includes the prefix marker, the NACK packet being indicative of congestion for any interest packet in the class of traffic indicated by the prefix marker over any path that includes the link. In addition, the method can include transmitting the NACK packet over the NDN environment towards a sender of the interest packet.

Example Embodiments

Turning to FIG. 1, FIG. 1 is a simplified block diagram illustrating a communication system 10 for congestion control using congestion prefix information in a NDN environment in accordance with one example embodiment. FIG. 1 illustrates a NDN environment 11 (generally indicated by an arrow) comprising a plurality of nodes 12(1)-12(6). In an example embodiment, nodes 12(1) and 12(2) may represent routers, nodes 12(3) and 12(4) may represent clients and nodes 12(5) and 12(6) may represent servers. Each node (e.g., 12(1), 12(2)) may include respective congestion modules (e.g., 14(1)-14(2)) for congestion control using congestion prefix information in NDN environment 11. Links 16(1)-16(5) may connect nodes 12(1)-12(6). As used herein, the term "link" refers to a communications channel (e.g., an information transfer path within a system, and the mechanism by which the path is affected) that connects two or more nodes (e.g., link 16(1) connects nodes 12(1) and 12(4)). The link may be an actual physical link or it may be a logical link that uses one or more actual physical links. Note that logical links can include any kind of communication channel, including an encapsulation over an existing link technology (e.g., Ethernet) or a tunnel over an existing network technology (e.g., IP, Transmission Control Protocol (TCP), Multi-Protocol Label Switching (MPLS), etc.)

Assume, merely for example purposes, that node 12(3) retrieves data from node 12(5) and node 12(4) retrieves data from node 12(6). Also assume, merely for example purposes, that while links 16(1)-16(4) can carry network traffic at 100 Mbps, link 16(5) between nodes 12(2) and 12(6) is experiencing congestion, such that its available bandwidth for additional traffic approaches zero. Thus, traffic flow between nodes 12(4) and 12(6) may be choked by congestion on link 16(5). According to various embodiments of communication system 10, a specific class of traffic in network 11 (e.g., traffic between nodes 12(4) and 12(6)) may be limited to 10 Mbps on link 16(3) (between nodes 12(1) and 12(2)), freeing up 90 Mbps for traffic between nodes 12(3) and 12(5) on link 16(3).

According to some embodiments, congestion control can be achieved without accurate identification of flows in any node, including endpoints (e.g., 12(2), 12(3), 12(5), and 12(6)). In various embodiments, nodes (e.g., 12(1) and/or 12(2)) can preferentially retard (e.g., slow down, throttle) transmission of certain interest packets (e.g., interest packets between 12(3) and 12(6)) based on upstream congestion (e.g., congestion on link 16(5)), and re-allocate bandwidth dedicated to the retarded interest packets in favor of other interest packets (e.g., interest packets between 12(2) and 12(5)) on the same link (e.g., link 16(3)). As used herein, an "interest packet" includes a unit of data communicated in NDN environments, wherein a consumer entity (e.g., nodes 12(3), 12(4)) asks for certain content, for example, by broadcasting its interest in the content over all available connectivity, trying different paths in some order, etc.

For purposes of illustrating the techniques of communication system 10, it is important to understand the communications that may be traversing the system shown in FIG. 1. The following foundational information may be viewed as a basis from which the present disclosure may be properly explained. Such information is offered earnestly for purposes of explanation only and, accordingly, should not be construed in any way to limit the broad scope of the present disclosure and its potential applications.

As used herein, “NDN” comprises a network architecture that allows creation of general content distribution networks that use Interest/Data exchanges rather than exchange models like rendezvous or publish/subscribe, etc. Examples of such content oriented network architecture include Named Data Networking Project’s network architecture (also called NDN) and CCN. Unlike Internet Protocol (IP) architecture, where communication endpoints are required to be named in each packet, and conversely, communication endpoints are the only named entities in each packet (e.g., IP packets can only name communication endpoints (i.e., as IP source and destination addresses)), the name in an NDN packet can be anything—an endpoint, a chunk of content, a command, etc. The names in NDN packets are hierarchically structured but otherwise arbitrary data identifiers. For example, the name can represent a chunk of data from a YouTube™ video directly, rather than embedding it in a conversation between a consuming host and the YouTube server. Thus, instead of pushing data to specific locations, NDN architecture permits data retrieval by name.

The NDN communication architecture has two prominent features: (1) traffic is receiver-driven; and (2) content retrieved in response to an interest packet traverses exactly the same links in reverse order. Communication in NDN is driven by a receiver (e.g., data consumer): to receive data, the consumer sends out an interest packet, which carries a name that identifies the desired data. A router in the network remembers the interface (e.g., a point where two interacting components meet and interact; also called face herein) from which the request is received and forwards the interest packet to a data producer (e.g., data source) by looking up the name in its Forwarding Information Base (FIB), which is populated by a name-based routing protocol.

After the interest packet reaches the data producer on the network that has the requested data, a data packet is sent back, which carries both the name and the content of the data, cryptographically bound together with an integrity hash signed by the data producer’s key. The data packet follows, in reverse, the path taken by the interest packet to reach the consumer. Neither interest packets nor data packets carry any host or interface addresses (such as IP addresses); interest packets are routed towards data producers based on the names carried in the interest packets, and data packets are returned based on the state information set up by the interest packets at each router hop.

Each intermediate NDN node (e.g., NDN router) maintains three data structures: (1) a content store for temporary caching of received data packets; (2) a pending Interest Table (PIT) for storing information about each interest packet it receives; and (3) a FIB for determining the next hop, wherein entries are entered according to name prefixes (rather than IP address prefixes). In addition, the NDN router has a strategy module that makes forwarding decisions for each interest packet. For example, when the router receives an interest packet, the strategy module first checks whether there is a matching data packet in the content store. If a match is found, the data packet is sent back to the incoming interface of the interest packet.

If the match is not found, the interest name is checked against entries in the PIT. Each PIT entry records the name, incoming interface(s) of interest packet(s), and outgoing interface(s) to which one of the interest packets has been forwarded. If the name exists in the PIT, which means that another interest packet (e.g., from another consumer) for the same name has been received and forwarded earlier, the router simply adds the incoming interface of the newly received interest packet to the existing PIT entry. If the name does not exist in the PIT, an entry is added into the PIT and the interest packet is forwarded to the next hop towards the data producer. Thus, when multiple interest packets for the same data are received, only the first interest packet is sent towards the data producer. When the data packet arrives, the router finds the matching PIT entry and forwards the data packet to all the interfaces listed in the PIT entry. The router removes the corresponding PIT entry, and optionally caches the data packet in the content store.

Turning to the FIB, the FIB entries record all the name prefixes announced in routing to a corresponding interface. Instead of announcing IP prefixes, each NDN router announces name prefixes that cover the data that the router is willing to serve. The announcement is propagated through the network via a routing protocol (e.g., border gateway protocol (BGP), Open Shortest Path First (OSPF)), and every router builds its FIB based on received routing announcements or defined locally. Any packet (e.g., interest packet or data packet) is forwarded to the interface that has the longest name match in the FIB on the content name in the packet. Each FIB entry further lists routing preferences for reaching the given name prefix for all policy-compliant interfaces (e.g., a specific interface is included, unless it is forbidden to serve the prefix by a preconfigured routing policy).

All the interfaces in the FIB entry are ranked to help the strategy module choose which interface(s) to use. Thus, the FIB entry can also record a data retrieval status (e.g., round trip time (RTT) estimate) for each interface, for example, that can serve to rank interfaces. For each prefix, the ranking of its interfaces is based on routing preference (e.g., determined by applying the routing policy and metrics to paths computed by routing), observed forwarding performance (e.g., based on whether the interface is working), and a forwarding policy set by network operator. Note that the routing policy determines which routes are available to the forwarding data plane; the forwarding policy determines the preference for each route. For example, if the forwarding policy is “the sooner the better,” interfaces with smaller RTTs will be ranked higher; if the forwarding policy is performance stability, the current working path is ranked higher. Yet another example is a higher preference for a particular neighbor, which leads to a higher percentage of interest packets being forwarded to that interface than other equally available ones.

The NDN architecture eschews prior models that employ flow-based data transfer in networks, and as such, new congestion control schemes are desirable. In at least one NDN scheme for congestion control, when an NDN node can neither satisfy nor forward the interest packet (e.g., there is no interface available for the requested name), it sends a negative acknowledgement (NACK) packet back to the downstream node that sent the interest packet. The NACK packet carries the same name as the corresponding interest packet, plus an error code explaining why the NACK packet was generated (e.g., congestion, No Path, etc.). If the downstream node has exhausted all its own forwarding options, it will propagate the NACK packet further downstream. The NACK packet notifies the downstream node of network problems quickly; the downstream node can subsequently take proper actions based

## 5

on the error code in the NACK packet, and delete the corresponding interest packet from its PIT. In the absence of packet losses, every pending interest packet is consumed by either a returned data packet or a NACK packet.

When the NDN router detects that a link has reached its load limit, it may automatically try other available links to forward the interest packets. If all the available links are congested, the router will return NACK packets to downstream routers, which then may in turn try their alternative paths. Consequently, traffic in the NDN network can automatically split among multiple parallel paths as needed to route around congestion. When excess interest packets trigger NACK packet returns from upstream routers, the router can dynamically adjust its rate limit based on the percentage of interest packets returned. Therefore, the downstream router can match its sending rate to whatever the upstream router can support. If the network reaches its capacity, the NACK packets will eventually be returned all the way back to the consumer and cause the application or transport layer in the source node to adjust the sending rate.

Another congestion control mechanism in the NDN architecture is interest-based shaping. Whereas basic TCP congestion control reacts to congestion after data packets are lost, by contrast, interest shaping proactively prevents data packet loss by regulating the interest rate in the first place. For example, an optimal interest shaping rate can be mathematically deduced if the shaper has knowledge of the data/interest size ratio, link capacity and demand in both directions over a single link. However, such currently available schemes handle a single hop and cannot be extended to multiple hops easily.

Moreover, one of the issues with multi-hop congestion control schemes in NDN architecture is that because there is no obvious flow (e.g. specified by a 5-tuple in IP architecture), it is not possible to determine a sub-class of traffic experiencing congestion multiple hops away, and therefore not possible for any node other than the node directly experiencing congestion to slow down some interest packets in preference to others in order to better utilize the network.

Communication system 10 is configured to address these issues (and others) in offering a system and method for congestion control using congestion prefix information in NDN environment 11. In a specific embodiment, an indicator of the exact FIB prefix used for forwarding over congested link 16(5) for a specific class of traffic (e.g., traffic between nodes 12(4) and 12(6)) may be included in a NACK packet sent back by node 12(2) for a corresponding interest packet due to the congestion. This prefix can be used by downstream nodes (e.g., node 12(1)) to identify the class of traffic that is likely to experience congestion if used for subsequent interest packet forwarding, and the nodes (e.g., 12(1)) can reroute, slow down, or NACK (e.g., send NACK packets corresponding to) matching interest packets accordingly.

Embodiments of communication system 10 can include a prefix marker in the NACK packet to specify the class of traffic that will see congestion and use the prefix marker as a selector in intermediate nodes (e.g., 12(1)) to slow down interest packets that can experience congestion upstream. For example, consider three nodes 12(1)-12(2)-12(6). If link 16(5) between nodes 12(2) and 12(6) is experiencing congestion, node 12(2) may see congestion for all traffic it wishes to send on link 16(5); specifically, any interest packet that reaches node 12(2) with a FIB prefix pointing towards node 12(6) may be retarded because of congestion on link 16(5). If a generic NACK packet (i.e. one lacking prefix information) is sent back to node 12(1), node 12(1) may have to retard other interest packets (e.g., associated with traffic between nodes

## 6

12(3) and 12(5)) that are sent towards node 12(2), but which are not destined to node 12(6) over congested link 16(5). Under currently existing schemes (e.g., that do not use embodiments of communication system 10), there is no information available on node 12(1) about exactly which traffic to preferentially retard.

According to one embodiment of communication system 10, FIB entries may be used to hold the necessary state for congestion control. If substantially all nodes (e.g., 12(1) and 12(2)) are running a full routing protocol, node 12(1) can determine (in the absence of any NACK packet) that any traffic going towards node 12(2), which eventually matches the FIB prefix of packets destined to node 12(6) may be retarded. Such a mechanism can work in scenarios where a routing boundary does not exist between nodes 12(1) and 12(2) (e.g. default route, summarized route, etc.). In an extreme case, routers near the requesting client can degenerate into (unnecessarily) retarding almost all traffic.

In another embodiment, the congestion information may be returned to downstream nodes (e.g., 12(1)) back from congested node 12(2) through appropriate NACK packets. For example, node 12(2) can generate a NACK packet with appropriate error codes, including information about the FIB prefix on node 12(2) that the node uses to select the face over which to forward the corresponding interest packet to node 12(6). The prefix may be returned towards the original sender (e.g., node 12(3)), thereby all downstream intermediate nodes, including the sender can recognize that any traffic sent out a face on which the NACK packet was received and matching the indicated prefix, is likely to experience congestion somewhere upstream. As used herein, the term “downstream” refers to a direction of NACK traffic that is opposite of the corresponding interest packet; the term “upstream” refers to a direction of NACK traffic that is the same as the corresponding interest packet.

Note that the FIB entry signaled back in the NACK packet may not even be present in forwarding tables elsewhere in NDN environment 12, or used downstream in any FIB. The specific FIB entry can simply be used for congestion control and relative interest prioritization/retardation. Also note that because the NACK packet already contains substantially all information being signaled, the NACK packet message can be substantially efficient (e.g., adding one small field to the NACK packet).

Merely for example purposes, consider a longer path: A---B---C---D---E---F, where A, B, C, D, E and F represent nodes in NDN environment 12. Congestion on the D--E link may be reported back to C, B, and eventually A. Because NDN architecture has no concept of host addresses, including the addresses of intermediate routers, it is probable that the addresses of D and E (or, for that matter even the server F) are not known to C, B or A. The specific message that B could process is “Throttle Content Prefix cisco.com/www towards C” since NDN architecture uses content prefixes only. An end-to-end throttling mechanism from C back to A could be used in an embodiment of communication system 10; however, such a mechanism could cause problems for end-to-end congestion control, foremost of which is that there is no defined server for consecutive content objects, therefore maintaining (and throttling) a congestion window as a representation of path state could be infeasible. In contrast, hop-by-hop throttling has the benefit that it can be done close to the congestion point, so other interest messages which traverse other non-congested paths from the same client would not be subject to NACK.

Additionally, note that B need not actually have the “cisco.com/www” route in its FIB. It can instead use a separate

queue with the prefix name, attached to an output face, independent of whether the FIB entry that actually causes traffic to be forwarded to that output face is the same, more specific, less specific, covering, or even a default route. After the congestion signal expires (e.g. based on a timer), B can forget about the entry, for example, by deleting it. Some embodiments may also use bucketing, where congestion signal prefixes are hashed into buckets (e.g., 16K buckets), with throttling applied to the entire bucket), for example, to handle a large set of congestion prefixes.

Without interest shaping, the interest packets would go to E and the data coming back would be dropped on D because the C-D link is congested (with a full buffer at D). With existing interest shaping schemes, B would forward the interests to C which would reject them (drop or NACK), but the endpoints that receive NACK packets (or observe drops) would have to co-operatively retard interest packets to avoid goodput (e.g., application level throughput) loss. In contrast, embodiments of communication system 10 may include the congested prefix information in the NACK packet, allowing both endpoints and intermediate nodes to effectively apply throttling to reduce congestion without losing goodput.

In contrast to asymmetric IP routing where a quench message cannot be guaranteed to be seen by any node on the downstream path other than the sender, NDN routing is guaranteed symmetric for each interest-data pair, so that substantially all intermediate nodes can see and act on the NACK packet. This enables sophisticated features like in-network traffic throttling and fairness enforcement. In addition, congestion-aware rerouting and unequal-cost path load balancing with spillover of traffic onto more expensive paths on demand may be implemented. Such features are not possible in IP architecture, and therefore the NDN NACK packet is more valuable than an IP source quench. Moreover, there is no unfairness with the NACK packet as there is with the quench (e.g., where it is in the interest of nodes to ignore the quench since other nodes may not throttle back and therefore gain an unfair advantage). Thus, in embodiments of communication system 10, edge routers may enforce throttling due to the NACK packet, independent of endpoint behavior.

In embodiments that use an interest shaping scheme, NACK packets are guaranteed not to add to network congestion on any link. Unlike source quench (or other explicit congestion signaling packets), the congestion control message embodied in the NACK packet cannot cause more congestion. Unlike a stateless IP forwarding plane, NDN architecture embraces a concept of a stateful forwarding plane in the network utilizing per-packet state for in-transit packets. As a result, such features that can use the NACK packet mechanism are feasible to implement in NDN environments unlike in IP environments. Embodiments of communication system 10 use the NACK packet as a congestion signal that can trigger appropriate responses in intermediate nodes in the network.

Moreover, the mechanisms included in embodiments of communication system 10 are substantially different from Internet Control Message Protocol (ICMP) source quench. Whereas ICMP source quench is sent from the network towards a sender of content, telling the sender to slow down, congestion NACK packets in embodiments of communication system 10 are sent in the opposite direction, from the network towards a requestor of content, asking to reduce the speed of the requests.

Other differences between ICMP source quench and embodiments of communication system 10 include: 1) ICMP source quench is only consumed by the sender of content, whereas congestion-NACK packets according to embodi-

ments of communication system 10 can be used by intermediate nodes for congestion control; 2) source quench identifies a single sender, whereas congestion-NACK packets according to embodiments of communication system 10 identify a content prefix that can experience congestion in a specific direction; 3) the NACK packets may be generated based on internal interest rate shaping numbers (e.g., where congestion has not yet occurred but is projected to occur) whereas ICMP source quench is generated when router queues start to overflow; and 4) ICMP source quench is incompatible with window-based congestion control protocols like transmission control protocol (TCP) as source quench goes to the sender but the receiver controls the window. Embodiments of communication system 10 may be compatible with both window-based and rate-based congestion control protocols as it signals the node (e.g., requestor) that can act on the NACK packet.

Embodiments of communication system 10 can signal congestion without consuming extra bandwidth (e.g., interest shaping schemes may typically ensure sufficient resources for the NACK packet in any case). Embodiments of communication system 10 can be quite precise in the information (e.g., FIB prefix) that is carried in the NACK packet. Different nodes in the network may or may not implement filtering or interest retardation; if so, they may use bucketing or binning to improve scalability. However, there is no loss of resolution in the NACK packet message due to bucketing or binning. Embodiments of communication system 10 can allow routers in the network to enforce 'good' client behavior, which can be useful on customer edge routers for Internet Service Providers (ISPs), thereby alleviating potential problems with misbehaving clients and co-operative congestion control. Embodiments of communication system 10 can enable hop-by-hop congestion control in NDN environment 12. Unlike some congestion control schemes in NDN architecture, embodiments of communication system 10 use the NACK in the network to do in-network traffic selection.

In NDN environment 12, each node 12(1)-12(6) may store some amount of data, whether it originated it or is simply caching something originated by another node. Each node may be connected to one or more immediate neighbors over appropriate links. Data moves from one node to the next only if requested, and each node is in control both of the rate at which it requests data and the rate at which it responds to requests. If responding to a request from a neighbor would contribute to congestion, instead of responding with the requested data, the node can respond with a NACK packet (e.g., indicating, "I'm too busy right now"). The requestor can then re-request, presumably at a slower rate, or could request via another path from another neighbor. Hence, link and queue congestion may reduce (or in some cases, may not occur). The link or queue does not remain in a state of being unsatisfactorily loaded, because each node is in control of the load its connected links carry. The throttling of interest packets may maintain a steady flow (e.g., not too large) of interest packets; the corresponding data packet flow in the reverse direction may be taken care of automatically, assuming a reasonable distribution of data segment size.

Turning to the infrastructure of communication system 10, the network topology can include any number of clients, servers, virtual machines, switches (including distributed virtual switches), routers, and other nodes inter-connected to form a large and complex network. Elements of FIG. 1 may be coupled to one another through one or more interfaces employing any suitable connection (wired or wireless), which provides a viable pathway for electronic communications. Additionally, any one or more of these elements may be

combined or removed from the architecture based on particular configuration needs. In addition to NDN protocols, communication system **10** may include a configuration capable of TCP/IP communications for the electronic transmission or reception of packets in a network. Communication system **10** may also operate in conjunction with a User Datagram Protocol/Internet Protocol (UDP/IP) or any other suitable protocol, where appropriate and based on particular needs. In addition, gateways, routers, switches, and any other suitable nodes (physical or virtual) may be used to facilitate electronic communication between various nodes in the network.

Note that the numerical and letter designations assigned to the elements of FIG. 1 do not connote any type of hierarchy; the designations are arbitrary and have been used for purposes of teaching only. Such designations should not be construed in any way to limit their capabilities, functionalities, or applications in the potential environments that may benefit from the features of communication system **10**. It should be understood that communication system **10** shown in FIG. 1 is simplified for ease of illustration.

The example network environment may be configured over a physical infrastructure that may include one or more networks and, further, may be configured in any form including, but not limited to, local area networks (LANs), wireless local area networks (WLANs), VLANs, metropolitan area networks (MANs), wide area networks (WANs), virtual private networks (VPNs), Intranet, Extranet, any other appropriate architecture or system, or any combination thereof that facilitates communications in a network. In some embodiments, each link may represent any electronic link supporting a LAN environment such as, for example, cable, Ethernet, wireless technologies (e.g., IEEE 802.11x), ATM, fiber optics, etc. or any suitable combination thereof. In other embodiments, links may represent a remote connection through any appropriate medium (e.g., digital subscriber lines (DSL), telephone lines, T1 lines, T3 lines, wireless, satellite, fiber optics, cable, Ethernet, etc. or any combination thereof) and/or through any additional networks such as a wide area networks (e.g., the Internet).

As used herein, a "node" is synonymous with apparatus and may be any network element (e.g., computers, network appliances, routers, switches, gateways, bridges, load balancers, firewalls, processors, modules, or any other suitable device, component, element, or object operable to exchange information in a network environment), client, server, peer, service, application, software program, or other object capable of sending, receiving, or forwarding information over communications channels in a network. Note that nodes may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information.

In various embodiments, congestion modules **14(1)** and **14(2)** represent software applications executing on suitable network elements, such as routers. In other embodiments, congestion modules **14(1)** and **14(2)** may be separate stand-alone modules that are connected to (e.g., plugged in, attached to, coupled to, etc.) suitable network elements, such as routers. Nodes **12(3)** and **12(4)** may represent clients such as laptop computers, smartphones, desktop computers, etc.; nodes **12(5)** and **12(6)** may represent servers, such as rack-mount servers in a data center. Note that virtually any number of nodes may be interconnected in NDN environment **12** without departing from the scope of the embodiments of communication system **10**.

Turning to FIG. 2, FIG. 2 is a simplified block diagram illustrating example details that may be associated with an embodiment of communication system **10**. Assume a NDN environment comprising nodes **12(1)**-**12(6)** as illustrated herein. Node **12(1)** may communicate with node **12(2)** over link **16(1)**; node **12(2)** may communicate with node **12(3)** over link **16(2)** at outgoing interface **0** on node **12(2)**; node **12(3)** may communicate with node **12(4)** over link **16(3)** at outgoing interface **7** on node **12(3)**; node **12(4)** may communicate with node **12(5)** over link **16(4)** at outgoing interface **4** on node **12(4)**; node **12(3)** may communicate with node **12(6)** over link **16(5)** at outgoing interface **0** on node **12(3)**; and node **12(6)** may communicate with node **12(5)** over link **16(6)** at outgoing interface **4** on node **12(6)**. Assume merely for example purposes, that node **12(1)** sends an interest packet **20** for certain content named appropriately (e.g., /com/example/video/widgetA) therein.

When interest packet **20** is received at node **12(2)**, node **12(2)** may check its content store, determine that a corresponding data is absent therein, check its PIT, enter the name in the PIT if not previously found, and forward interest packet **20** to node **12(3)** according to its FIB entry (e.g., node **12(3)** may have previously announced /com/ as a named prefix and consequently been associated at node **12(2)** with name /com/ in its FIB). Each intermediate node **12(3)**-**12(4)** may perform substantially identical functions. Assume that node **12(4)** attempts to forward interest packet **20** to node **12(5)** based on its FIB entry, which associates /com/example with an interface for link **16(4)**.

Assume that node **12(4)** senses congestion on link **16(4)**. According to embodiments of communication system **10**, node **12(4)** may generate a NACK packet **22** comprising a prefix marker indicative of a class of traffic associated with the content name (e.g., /com/example/video/widgetA) and intended to be forwarded on link **16(4)** towards node **12(5)**. The prefix marker may include the FIB prefix /com/example used by node **12(4)** to attempt to forward interest packet **20** to node **12(5)**. NACK packet **22** may be transmitted to downstream node **12(3)**.

Adding the prefix marker information into NACK packet **22** may not consume any extra bits in some embodiments. In some embodiments, in-network shaping can be implemented only at selective nodes (e.g., that process NACK packet **22**), while other nodes simply forward NACK packet **22** and operate using best-effort algorithms. Note that congestion-generated NACK packet **22** does not cause congestion, as it uses resources budgeted for (larger) data packets which may potentially never arrive (e.g., due to congestion).

Node **12(3)** may read the content name on NACK packet **22**, and associate it with interest packet **20**; node **12(3)** may read the prefix marker on NACK packet **22** and generate a congestion marker (CM) table **24** at face **7**, corresponding to the outgoing face on which interest packet **20** was sent to node **12(4)**. CM table **24** may associate the prefix marker /com/example with a CM (e.g., 1) indicative of congestion associated therewith. Note that if node **12(3)** receives another NACK packet for the same prefix marker, the CM may be incremented by 1, and so on.

Node **12(3)** may forward NACK packet **22** downstream to node **12(2)**. Node **12(2)** may read the content name on NACK packet **22**, and associate it with interest packet **20**; node **12(2)** may read the prefix marker on NACK packet **22** and generate a congestion marker (CM) table **24** at face **0**, corresponding to the outgoing face on which interest packet **20** was sent to node **12(3)**. CM table **24** at node **12(2)** may associate the prefix marker /com/example with a CM (e.g., 1) indicative of con-

## 11

gestion associated therewith. The process may continue until NACK packet 22 reaches the last node (e.g., 12(1)).

When node 12(3) receives another interest packet 20 indicative of the same prefix as in its CM table 24, node 12(3) may retard forwarding the interest packet to node 12(4). In another scenario, when node 12(3) receives another interest packet 20 indicative of the same prefix as in its CM table 24, node 12(3) may route interest packet 20 to node 12(6) over link 16(5) instead of (or in addition to) forwarding NACK packet 22 downstream to node 12(2). Node 12(6) may forward interest packet 20 to node 12(5), which may possess the corresponding data packet, which can be returned eventually to node 12(1). Thus, intermediate nodes 12(2) and 12(3) may retard or re-route interest packets based on CM table 24 appropriately.

Turning to FIG. 3, FIG. 3 is a simplified diagram illustrating certain details of an example CM table 24 according to an embodiment of communication system 10. CM table 24 may include various prefixes 26, with corresponding CMs 28. Each output face may maintain a corresponding (potentially different) CM table 24. When a congestion notification matching FIB prefix is received over the face, it accumulates a CM 28. Each CM 28 may decay over time. An output interest queue on the face may execute a queuing logic, such as Weighted Fair Queuing (WFQ) or Weighted Random Early Detection (WRED). Weights of CM-matching interests may be increased appropriately. Lowering the interest rate for sub-prefixes can work across multiple hops. As the load from client-side hops decreases, queue weights may become stable.

Turning to FIG. 4, FIG. 4 is a simplified block diagram illustrating example details of an example node 12 according to an embodiment of communication system 10. Node 12 may include a content store 30, a PIT 32, a FIB 34, and various interfaces 36 (e.g., 36(0), 36(1), and 36(2)). Each interface 36(0)-36(2) may have an associated CM table 24(0)-24(2), respectively. Each interface 36(0)-36(2) may communicate with appropriate entities 38(0)-38(2), comprising, respectively, a wireless network, a WAN such as the Internet, and one or more applications. Congestion module 14 of node 12 may include a NACK generator 40, a CM decay module 42, a prefix marker generator 44, a queuing logic module 46, a processor 48, and a memory element 50.

In a general sense, NDN architecture utilizes hierarchically structured names, e.g., a video produced by company "Example1" may have the name /com/example1/videos/WidgetA.mpg, where '/' indicates a boundary between name components. The hierarchy enables routing to scale, among other advantages. Name conventions are specific to applications but opaque to the network; thus routers do not know the meaning of a name (although they see the boundaries between components in a name), allowing each application to choose a naming scheme that fits its needs and independent of the network.

Consequently, content store 30 may store each name for which it has content; likewise, PIT 32 may store each name for which it has received interest packet at requesting faces. For example, interest packets for name /com/example1/maps may be received on all three faces 36(0)-36(2); interest packets for name MovieABC may be received on only interface 36(2); interest packets for name /com/example2/videos/WidgetA.mpg/v3/s0 may be received on face 36(0); interest packets for name /com/example2/email\_service/eg@mail.com/123 may be received on faces 36(0)-36(1); and so on.

FIB 34 may store the name prefix announced by appropriate routers along with a corresponding face list (e.g., facing the router that announced the name prefix). Thus, name /com/

## 12

example1/may correspond to faces 36(0)-36(2); MovieABC may correspond to face 36(2); and so on. Each CM table 24 may store a prefix (which need not match with any FIB entries in FIB 34) and a corresponding CM indicative of congestion experienced by packets in the class of traffic associated with the prefix in CM table 24. For example, CM table 24(0) indicates that /com/example2 prefix is associated with a CM of 3 (e.g., experiencing high congestion); therefore interest packets having the name prefix /com/example2 may be suitably retarded or re-routed away from interface 36(0). In another example, CM table 24(1) indicates that /com/example2 prefix is associated with a CM of 3 (e.g., experiencing high congestion); therefore interest packets having the name prefix /com/example2 may be suitably retarded or re-routed away from interface 36(1); /com/example1/ prefix is associated with a CM of 1 (e.g., experiencing moderate congestion).

During operation, assume that node 12 receives an interest packet on face 36(1) for content having name /com/example2/videos/WidgetA.mpg/v2/s1. Node 12 may determine, based on content store 30 that it does not have the corresponding data packet. Node 12 may also determine, based on PIT 32, that the interest packet is not associated with any previous interest packets, and may create a new entry therefor. Based on the entry corresponding to /com/example2 in FIB 34, node 12 may determine that the interest packet can be sent out over face 36(0).

Assume that node 12 senses congestion on face 36(0) for the interest packet. Prefix marker generator 44 may generate a suitable prefix marker including the FIB prefix /com/example2 therein. NACK generator 40 may generate a NACK packet that includes the prefix marker and send out the NACK packet to the downstream node from which the interest packet was received. In some embodiments, node 12 may also augment CM table 24(0) at face 36(0) with the appropriate CM for the prefix marker. CM decay module 42 may appropriately decay the CM based on a suitable algorithm. Queuing logic module 46 may recalculate the route and/or increase the weight of the queue for interest packets in the class of traffic having the FIB prefix/com/example2 to be sent out over face 36(0).

Assume that the interest packet was subsequently sent out over face 36(1), and node 12 receives another NACK packet from an upstream node indicating congestion on some link upstream therefrom for the prefix/com/example2. Node 12 may augment the CM value of /com/example2 entry in CM table 24(1) and forward the NACK packet to the next downstream node. The next time an interest packet having a name associated with /com/example2 is received, node 12 may know, based on CM tables 24(0) and 24(1), that congestion for the specific class of traffic is being experienced somewhere upstream and take appropriate action (e.g., retard the interest packet, or re-route it).

Turning to FIG. 5, FIG. 5 is a simplified block diagram illustrating example details of an embodiment of communication system 10. Incoming interest packet 20 and data packet 52 may be assigned to respective queues. A WRED module 54 may assign a suitable weight to interest packet 20 to generate a weighted queue 56. In one embodiment, the weights may be assigned based on CM 28 in CM table 24 corresponding to prefix 26 observed in interest packet 20. In other embodiments, interest packet 20 may carry hash markers based on the FIB entry used to forward them. The interest hash weights may represent congestion.

All interest queues may start out with substantially equal weights (e.g., as in a traditional fair queue). As backpressure (e.g., buildup or increase in queue size from less outflow of packets in the queue) is detected (e.g., due to congestion), the

## 13

weight (e.g., effective drain rate) of the queue which matches the FIB entry in question may be reduced proportionally. As backpressure reduces, weights may be returned to original values. For example, queue **56** may have lower thresholds for lower priority packets (e.g., interest packets experiencing congestion). A queue buildup (e.g., backpressure) may cause the lower priority packets to be dropped, protecting higher priority packets in the same queue.

Each flow of weighted interest packets are weighted for queuing purposes (e.g., per session, or by other criteria independent of CM **28**) by module **58** and placed into another queue **60** for processing by WFQ module **62**. Data packet **52** may be likewise weighted for queuing purposes by module **64** and placed into queue **66** for processing by WFQ module **62**. WFQ allows different scheduling priorities to statistically multiplexed data flows comprising interest packets and data packets. Each flow has a separate FIFO queue, namely queue **60** for interest packets, and queue **66** for data packets. In a general sense, with a link data rate of R, at any given time N active flows are serviced simultaneously, each at an average data rate of R/N; if different flows are assigned different weights for queuing purpose, each flow will experience a different flow rate based on the assigned weight.

Turning to FIG. 6, FIG. 6 is a simplified flow diagram illustrating example operations **70** that may be associated with an embodiment of communication system **10**. The operations may start at **72**, for example, when interest packet **20** is received at node **12(1)**. At **74**, an attempt may be made to forward interest packet **20** from node **12(1)** to node **12(2)**. At **76**, node **12(1)** may sense congestion on a link connecting node **12(1)** and node **12(2)**. At **78**, prefix marker generator **44** may generate a prefix marker comprising the FIB prefix at node **12(1)** used while attempting to forward interest packet to node **12(2)**. At **80**, NACK generator **40** may generate NACK packet **22** comprising the prefix marker. At **82**, NACK packet **22** may be transmitted by node **12(1)** to a downstream node that sent interest packet **20**. The operations may end at **84**.

Turning to FIG. 7, FIG. 7 is a simplified flow diagram illustrating example operations **90** that may be associated with an embodiment of communication system **10**. The operations may start at **92**, for example, when interest packet **20** is sent out. At **94**, node **12** may receive NACK packet **22** corresponding to interest packet **20**. At **96**, a determination may be made if node **12** is the original sender of interest packet **20** corresponding to NACK packet **22**. If not, CM **28** corresponding to the prefix marker in CM table **24** may be incremented at **98**. At **100**, NACK packet **22** may be transmitted to a downstream node. The operations may end at **102**. Turning back to **96**, if node **12** is the original sender of interest packet **20**, the operations may end thereupon.

Turning to FIG. 8, FIG. 8 is a simplified flow diagram illustrating example operations **110** that may be associated with an embodiment of communication system **10**. The operations may start at **112**, for example, when interest packet **20** is received. At **114**, equal weights are assigned to all queues of interest packets. At **116**, an increasing backpressure is detected corresponding to a specific traffic class (e.g., identified by the prefix marker, interest hash marker, CM **28**, etc.). At **118**, the weight of queue matching the traffic class may be reduced. At **120**, reducing backpressure corresponding to the specific traffic class may be detected (e.g., when congestion eases back up). At **122**, the weight of the queue matching the specific traffic class may be increased. The operations may end at **124**.

Note that in this Specification, references to various features (e.g., elements, structures, modules, components, steps,

## 14

operations, characteristics, etc.) included in “one embodiment”, “example embodiment”, “an embodiment”, “another embodiment”, “some embodiments”, “various embodiments”, “other embodiments”, “alternative embodiment”, and the like are intended to mean that any such features are included in one or more embodiments of the present disclosure, but may or may not necessarily be combined in the same embodiments. Note also that an ‘application’ as used herein this Specification, can be inclusive of an executable file comprising instructions that can be understood and processed on a computer, and may further include library modules loaded during execution, object files, system files, hardware logic, software logic, or any other executable modules. Furthermore, the words “optimize,” “optimization,” and related terms are terms of art that refer to improvements in speed and/or efficiency of a specified outcome and do not purport to indicate that a process for achieving the specified outcome has achieved, or is capable of achieving, an “optimal” or perfectly speedy/perfectly efficient state.

In example implementations, at least some portions of the activities outlined herein may be implemented in software in, for example, congestion module **14**. In some embodiments, one or more of these features may be implemented in hardware, provided external to these elements, or consolidated in any appropriate manner to achieve the intended functionality. The various network elements (e.g., node **12**) may include software (or reciprocating software) that can coordinate in order to achieve the operations as outlined herein. In still other embodiments, these elements may include any suitable algorithms, hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof.

Furthermore, node **12** described and shown herein (and/or their associated structures) may also include suitable interfaces for receiving, transmitting, and/or otherwise communicating data or information in a network environment. Additionally, some of the processors and memory elements associated with the various nodes may be removed, or otherwise consolidated such that a single processor and a single memory element are responsible for certain activities. In a general sense, the arrangements depicted in the FIGURES may be more logical in their representations, whereas a physical architecture may include various permutations, combinations, and/or hybrids of these elements. It is imperative to note that countless possible design configurations can be used to achieve the operational objectives outlined here. Accordingly, the associated infrastructure has a myriad of substitute arrangements, design choices, device possibilities, hardware configurations, software implementations, equipment options, etc.

In some of example embodiments, one or more memory elements (e.g., memory element **50**) can store data used for the operations described herein. This includes the memory element being able to store instructions (e.g., software, logic, code, etc.) in non-transitory media, such that the instructions are executed to carry out the activities described in this Specification. A processor can execute any type of instructions associated with the data to achieve the operations detailed herein in this Specification. In one example, processors (e.g., processor **48**) could transform an element or an article (e.g., data) from one state or thing to another state or thing. In another example, the activities outlined herein may be implemented with fixed logic or programmable logic (e.g., software/computer instructions executed by a processor) and the elements identified herein could be some type of a programmable processor, programmable digital logic (e.g., a field programmable gate array (FPGA), an erasable programmable read only memory (EPROM), an electrically erasable pro-

grammable read only memory (EEPROM)), an ASIC that includes digital logic, software, code, electronic instructions, flash memory, optical disks, CD-ROMs, DVD ROMs, magnetic or optical cards, other types of machine-readable mediums suitable for storing electronic instructions, or any suitable combination thereof.

These devices may further keep information in any suitable type of non-transitory storage medium (e.g., random access memory (RAM), read only memory (ROM), field programmable gate array (FPGA), erasable programmable read only memory (EPROM), electrically erasable programmable ROM (EEPROM), etc.), software, hardware, or in any other suitable component, device, element, or object where appropriate and based on particular needs. The information being tracked, sent, received, or stored in communication system 10 could be provided in any database, register, table, cache, queue, control list, or storage structure, based on particular needs and implementations, all of which could be referenced in any suitable timeframe. Any of the memory items discussed herein should be construed as being encompassed within the broad term 'memory element.' Similarly, any of the potential processing elements, modules, and machines described in this Specification should be construed as being encompassed within the broad term 'processor.'

It is also important to note that the operations and steps described with reference to the preceding FIGURES illustrate only some of the possible scenarios that may be executed by, or within, the system. Some of these operations may be deleted or removed where appropriate, or these steps may be modified or changed considerably without departing from the scope of the discussed concepts. In addition, the timing of these operations may be altered considerably and still achieve the results taught in this disclosure. The preceding operational flows have been offered for purposes of example and discussion. Substantial flexibility is provided by the system in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the discussed concepts.

Although the present disclosure has been described in detail with reference to particular arrangements and configurations, these example configurations and arrangements may be changed significantly without departing from the scope of the present disclosure. For example, although the present disclosure has been described with reference to particular communication exchanges involving certain network access and protocols, communication system 10 may be applicable to other exchanges or routing protocols. Moreover, although communication system 10 has been illustrated with reference to particular elements and operations that facilitate the communication process, these elements, and operations may be replaced by any suitable architecture or process that achieves the intended functionality of communication system 10.

Numerous other changes, substitutions, variations, alterations, and modifications may be ascertained to one skilled in the art and it is intended that the present disclosure encompass all such changes, substitutions, variations, alterations, and modifications as falling within the scope of the appended claims. In order to assist the United States Patent and Trademark Office (USPTO) and, additionally, any readers of any patent issued on this application in interpreting the claims appended hereto, Applicant wishes to note that the Applicant: (a) does not intend any of the appended claims to invoke paragraph six (6) of 35 U.S.C. section 112 as it exists on the date of the filing hereof unless the words "means for" or "step for" are specifically used in the particular claims; and (b) does

not intend, by any statement in the specification, to limit this disclosure in any way that is not otherwise reflected in the appended claims.

What is claimed is:

1. A method, comprising:

sensing, at a first node in a Named Data Networking (NDN) environment, congestion preventing an interest packet from being forwarded over a link to a second node;

generating a prefix marker associated with a class of traffic to which the interest packet belongs;

generating a negative acknowledgement (NACK) packet comprising the prefix marker, wherein the NACK packet indicates congestion for any interest packet in the class of traffic indicated by the prefix marker over any path that includes the link; and

transmitting the NACK packet over intermediate nodes in the NDN environment towards a sender of the interest packet, wherein the intermediate nodes re-route any interest packet in the class of traffic indicated by the prefix marker over at least one non-congested path subsequent to receiving the NACK packet.

2. The method of claim 1, wherein the prefix marker includes a forwarding information base (FIB) prefix at the first node that caused the interest packet to be attempted to be forwarded to the second node, wherein each interest packet includes at least a portion of the FIB prefix in a content name, wherein the FIB prefix is indicative of the class of traffic.

3. The method of claim 1, wherein each intermediate node retards any interest packet in the class of traffic indicated by the prefix marker subsequent to receiving the NACK packet.

4. The method of claim 1, wherein each intermediate node maintains a congestion marker (CM) table at an interface on which the NACK packet was received, wherein the CM table comprises an association between the prefix marker and a CM, wherein the CM represents a congestion level associated with the prefix marker at the interface.

5. The method of claim 4, wherein the CM decays over time.

6. The method of claim 4, wherein an output interest queue at each intermediate node executes a queuing logic to slow down any interest packet in the class of traffic indicated by the prefix marker, wherein the queuing logic increases weights of interest packets having names matching prefix markers in the CM table, wherein the respective weights are proportionally increased with the CMs.

7. The method of claim 6, wherein the queuing logic comprises at least one of weighted random early detection and weighted fair queuing.

8. The method of claim 4, wherein each interest packet includes a hash marker based on a corresponding FIB entry, wherein the hash marker represents congestion.

9. Non-transitory tangible media that includes instructions for execution, which when executed by a processor, is operable to perform operations comprising:

sensing, at a first node in a NDN environment, congestion preventing an interest packet from being forwarded over a link to a second node;

generating a prefix marker associated with a class of traffic to which the interest packet belongs;

generating a NACK packet comprising the prefix marker, wherein the NACK packet indicates congestion for any interest packet in the class of traffic indicated by the prefix marker over any path that includes the link; and

transmitting the NACK packet over intermediate nodes in the NDN environment towards a sender of the interest packet, wherein the intermediate nodes re-route any



17

interest packet in the class of traffic indicated by the prefix marker over at least one non-congested path subsequent to receiving the NACK packet.

10. The media of claim 9, wherein the prefix marker includes a FIB prefix at the first node that caused the interest packet to be attempted to be forwarded to the second node, wherein each interest packet includes at least a portion of the FIB prefix in a content name, wherein the FIB prefix is indicative of the class of traffic.

11. The media of claim 9, wherein each intermediate node maintains a CM table at an interface on which the NACK packet was received, wherein the CM table comprises an association between the prefix marker and a CM, wherein the CM represents a congestion level associated with the prefix marker at the interface.

12. The media of claim 11, wherein an output interest queue at each intermediate node executes a queuing logic to slow down any interest packet in the class of traffic indicated by the prefix marker, wherein the queuing logic increases weights of interest packets having names matching prefix markers in the CM table, wherein the respective weights are proportionally increased with the CMs.

13. A first node, comprising:

a memory element for storing data; and

a processor, wherein the processor executes instructions associated with the data, wherein the processor and the memory element cooperate, such that the first node is configured for:

sensing, at the first node in a NDN environment, congestion preventing an interest packet from being forwarded over a link to a second node;

generating a prefix marker associated with a class of traffic to which the interest packet belongs;

18

generating a NACK packet comprising the prefix marker, wherein the NACK packet indicates congestion for any interest packet in the class of traffic indicated by the prefix marker over any path that includes the link; and

transmitting the NACK packet over intermediate nodes in the NDN environment towards a sender of the interest packet, wherein the intermediate nodes re-route any interest packet in the class of traffic indicated by the prefix marker over at least one non-congested path subsequent to receiving the NACK packet.

14. The first node of claim 13, wherein the prefix marker includes a FIB prefix at the first node that caused the interest packet to be attempted to be forwarded to the second node, wherein each interest packet includes at least a portion of the FIB prefix in a content name, wherein the FIB prefix is indicative of the class of traffic.

15. The first node of claim 13, wherein each intermediate node maintains a CM table at an interface on which the NACK packet was received, wherein the CM table comprises an association between the prefix marker and a CM, wherein the CM represents a congestion level associated with the prefix marker at the interface.

16. The first node of claim 15, wherein an output interest queue at each intermediate node executes a queuing logic to slow down any interest packet in the class of traffic indicated by the prefix marker, wherein the queuing logic increases weights of interest packets having names matching prefix markers in the CM table, wherein the respective weights are proportionally increased with the CMs.

\* \* \* \* \*