



(86) Date de dépôt PCT/PCT Filing Date: 2013/03/14
(87) Date publication PCT/PCT Publication Date: 2013/09/26
(85) Entrée phase nationale/National Entry: 2014/09/16
(86) N° demande PCT/PCT Application No.: US 2013/031714
(87) N° publication PCT/PCT Publication No.: 2013/142282
(30) Priorité/Priority: 2012/03/20 (US61/613,131)

(51) Cl.Int./Int.Cl. *H04L 12/721* (2013.01),
H04L 12/741 (2013.01)
(71) Demandeur/Applicant:
RAYTHEON COMPANY, US
(72) Inventeurs/Inventors:
WANG, MU-CHENG, US;
DAVIDSON, STEVEN A., US;
CHUANG, YI-CHAO, US
(74) Agent: BERESKIN & PARR LLP/S.E.N.C.R.L.,S.R.L.

(54) Titre : ROUTAGE D'UN PAQUET DE DONNEES DANS UN RESEAU DE COMMUNICATION
(54) Title: ROUTING BY SELECTING A ROUTING TABLE FROM A PLURALITY OF ROUTING TABLES

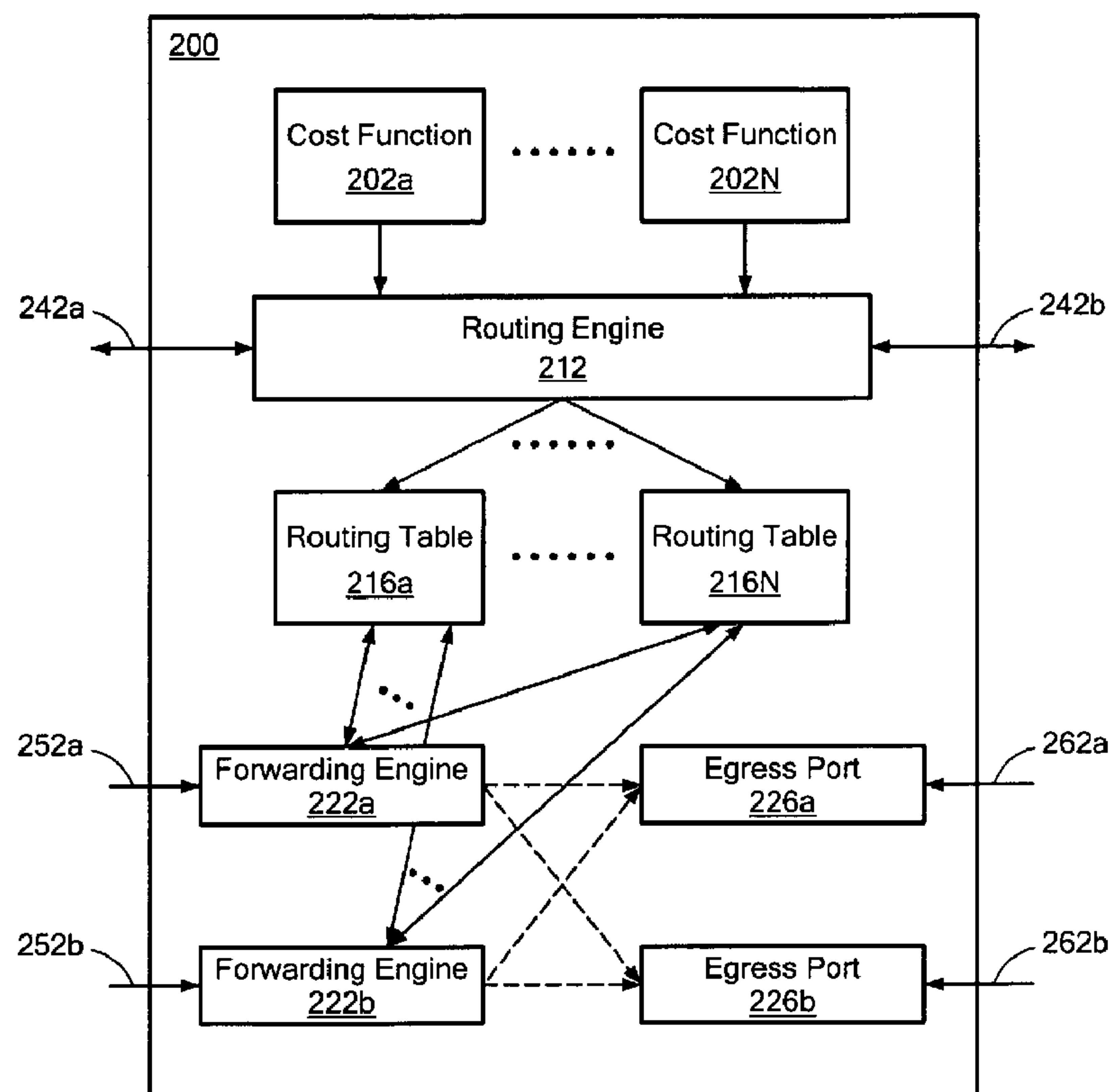


FIG. 2

(57) **Abrégé/Abstract:**

In one aspect, a method includes receiving a data packet at a routing node that includes a processor. The method also includes determining at least one value for the data packet, selecting a routing table from a plurality of routing tables stored at the routing

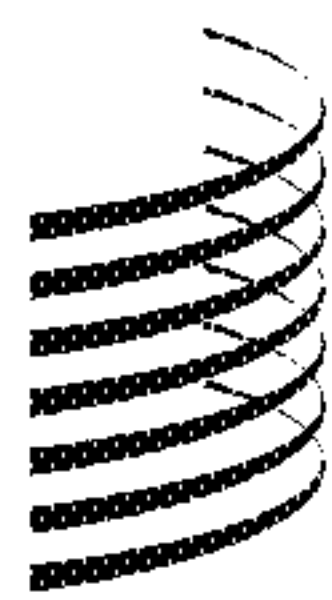


(57) **Abrégé(suite)/Abstract(continued):**

node in response to the at least one value for the packet and forwarding the data packet in response to the routing table selected. Each routing table is associated with a respective one cost function.

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

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
26 September 2013 (26.09.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/142282 A1

(51) International Patent Classification:

H04L 12/721 (2013.01) *H04L 12/741* (2013.01)

(21) International Application Number:

PCT/US2013/031714

(22) International Filing Date:

14 March 2013 (14.03.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/613,131 20 March 2012 (20.03.2012) US

(71) Applicant: RAYTHEON COMPANY [US/US]; 870 Winter Street, Waltham, Massachusetts 02451-1449 (US).

(72) Inventors: WANG, Mu-Cheng; 1 Wright Terrace, Acton, Massachusetts 01720 (US). DAVIDSON, Steven, A.; 2 Ashley Circle, Acton, Massachusetts 01720 (US). CHUANG, Yi-Chao, S.; 21 Black Horse Drive, Acton, Massachusetts 01720-2046 (US).

(74) Agents: MOOSEY, Anthony, T. et al.; Daly, Crowley, Mofford & Durkee, LLP, 354A Turnpike St., Suite 301A, Canton, Massachusetts 02021 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: ROUTING A DATA PACKET IN A COMMUNICATION NETWORK

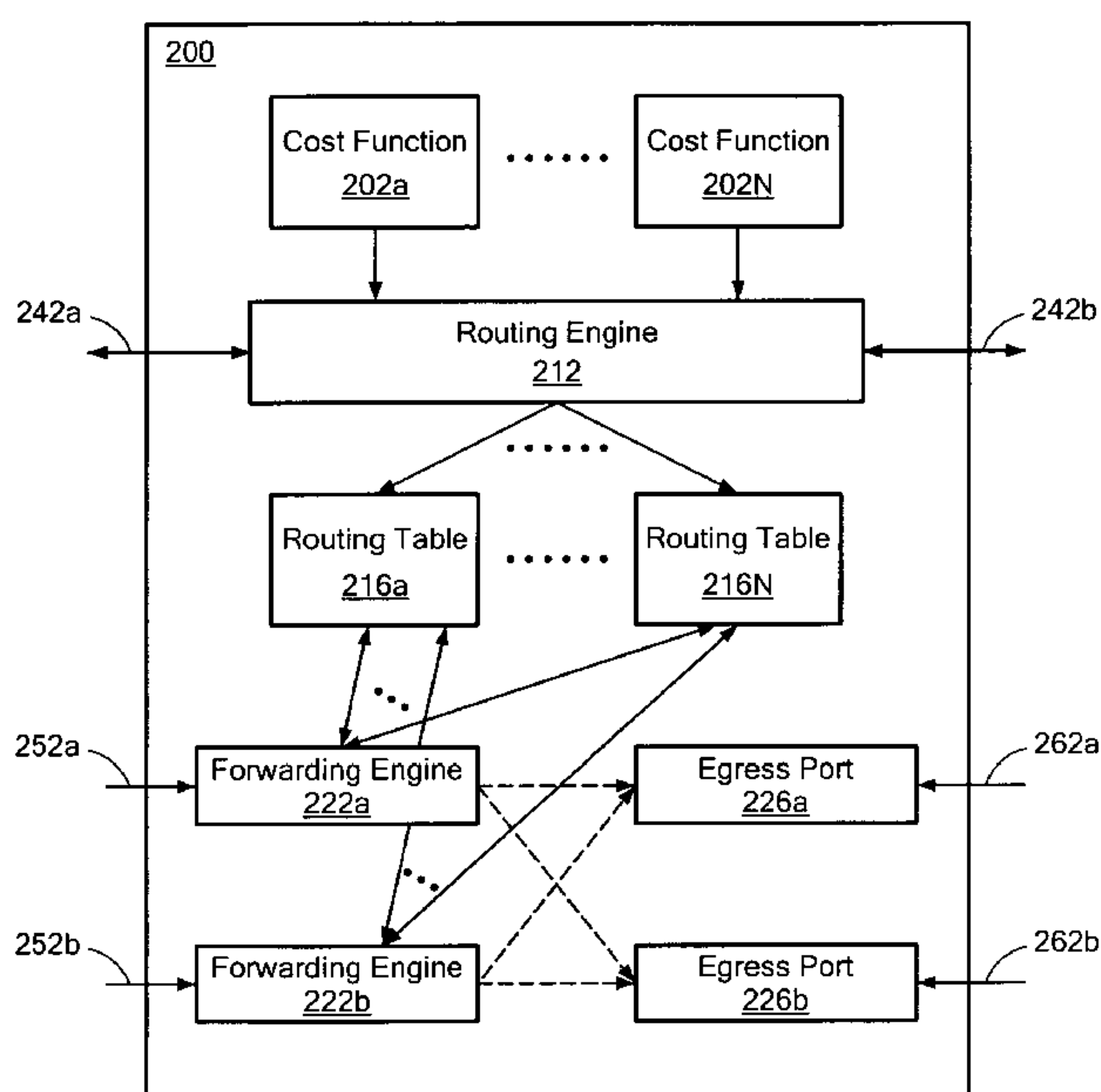


FIG. 2

(57) Abstract: In one aspect, a method includes receiving a data packet at a routing node that includes a processor. The method also includes determining at least one value for the data packet, selecting a routing table from a plurality of routing tables stored at the routing node in response to the at least one value for the packet and forwarding the data packet in response to the routing table selected. Each routing table is associated with a respective one cost function.

WO 2013/142282 A1



— *as to the applicant's entitlement to claim the priority of
the earlier application (Rule 4.17(iii))*

Published:

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

ROUTING BY SELECTING A ROUTING TABLE FROM A PLURALITY OF ROUTING TABLES

BACKGROUND

A communication network includes multiple routers. The routers are located at
5 subnet boundaries that are located between a sender and a receiver. The routers transfer
data packets originating from the sender to the intended receiver. Often a
communication network has multiple possible paths between the sender and the receiver,
but only one single path is chosen to send data between the sender and the receiver.

10

SUMMARY

In one aspect, a method includes receiving a data packet at a routing node that
includes a processor. The method also includes determining at least one value for the
data packet, selecting a routing table from a plurality of routing tables stored at the
routing node in response to the at least one value for the packet and forwarding the data
15 packet in response to the routing table selected. Each routing table is associated with a
respective one cost function.

In another aspect, a routing node includes electronic hardware circuitry
configured to receive a data packet at a routing node, determine at least one value for the
data packet; select a routing table from a plurality of routing tables stored at the routing
20 node in response to the at least one value for the packet and forward the data packet in
response to the routing table selected. Each routing table is associated with a respective
one cost function.

In a further aspect, an article includes a non-transitory computer-readable
medium that stores computer-executable instructions. The instructions causing a
25 machine to receive a data packet at a routing node, determine at least one value for the

data packet, select a routing table from a plurality of routing tables stored at the routing node in response to the at least one value for the packet and forward the data packet in response to the routing table selected. Each routing table is associated with a respective one cost function.

5 One or more of the aspects above may include one or more of the following features. Receiving a data packet at a routing node may include receiving the data packet at the routing node from a first link and forwarding the data packet in response the value of the data packet may include forwarding the data packet to a second link. The plurality of routing tables may be combined into a combined table incorporating value-to route
10 associations and selecting a routing table from a plurality of routing tables may include selecting the combined table. Determining the at least one value of the data packet may include determining at least one value located in a header of the data packet.
Determining at least one value located in a header of the data packet may include determining a Differentiated Services (DiffServ) code point (DSCP) value in the data
15 packet. Determining at least one value located in a header of the data packet may include determining at least one of a port number value or ID, or a source-destination pair value.

BRIEF DESCRIPTION OF THE DRAWINGS

20 FIG. 1 is a block diagram of an example of a communication network.
FIG. 2 is a block diagram of a routing node.
FIG. 3 is a flowchart of an example of a process to forward a data packet.
FIG. 4 is a block diagram of a computer on the process of FIG. 3 may be
implemented.

25

DETAILED DESCRIPTION

Described herein is an approach that enables a routing node to forward a data packet based on the data packet itself that, for example, has the added benefit of spreading the traffic load across the multiplicity of possible paths. In this invention, a routing node includes a plurality of routing tables with each routing table corresponding to a respective cost function (versus conventional routing where only one routing table is used). Based on a value in the data packet a routing table is selected that determines where the data packet is routed.

Referring to FIG. 1, a communication network 100 includes nodes 102a-102h, file transfer protocol (FTP) transceivers 108a-108b and voice transceivers 110a-110b. The FTP transceiver 108a and the voice transceiver 110a are coupled to the node 102a. The node 102a is coupled to the node 102b by a link 118a, and is coupled to the node 102c by a link 118b. The node 102b is coupled to the node 102d by a link 118c and is coupled to the node 102e by a link 118d. The node 102c is coupled to the node 102d by a link 118f and is coupled to the node 102e by a link 118e. The node 102d is coupled to the node 102f by a link 118h and is coupled to the node 102g by a link 118i. The node 102e is coupled to the node 102f by a link 118g and is coupled to the node 102h by a link 118j. The node 102f is coupled to the FTP transceiver 108b and the voice transceiver 110b. Each of the links 118a-118j may be one of wired links, fiber optic links, wireless links or a combination of the three (or any other media that can carry IP traffic).

As can be observed in FIG. 1, there are a number of paths between the nodes 102a and 102f that data packets can travel. In prior approaches there would be a single “best” path chosen regardless of whether the packets were voice data or FTP data.

However, as described herein, a route is chosen for each data packet based on the characteristics (delivery needs) of the data packet.

Referring to FIG. 2, in one example of how it may be implemented, a routing node 200 includes cost functions 202a-202N, a routing engine 212, routing tables 216a-216N, forwarding engines 202a-202b and egress ports 226a-226b. Each routing table 216a-216N corresponds to a respective one of the cost function 202a-202N (e.g., the routing table 216a corresponds to the cost function 202a; the routing table 216N corresponds to the cost function 202N). In one example, the routing engine 212 generates a routing table 216a-216N for each cost function 202a-202N.

For example, once all the cost functions are defined, the router builds the routing tables 216a-216N. For every given cost function 202a-202N, each one corresponding to one (each) of the data characteristics to be accommodated on the network, the Routing Engine 212 calculates the cost metric for each candidate route. Then, the Routing Engine 212 builds a routing table by selecting the best paths (interfaces) for the data packet's destination. This process repeats until all routing tables are built. To perform the packet forwarding, the routing node 200 first selects the routing table by using the value determined for the packet by methods that include one of various packet classification schemes available (e.g., Differentiated Services (DiffServ) Code Point (DSCP), port number or ID, source-destination pair, and so forth). Then, the routing node 200 selects a forwarding path (interface or egress port) based on the routing table and on the destination address. If multiple paths exist for the targeted address, the routing node 200 supports equal-cost or unequal-cost load balancing. The routing node 200 distributes traffic evenly or proportionally with respect to the cost metric among those routes, making them equal in cases where the metrics are of equivalent value.

The routing engine 212 receives topology and link state updates through the connections 242a, 242b and updates the routing tables 216a-216N based on current network conditions (e.g., loading, capacity, delay/latency and so forth).

In other examples, the cost functions 202a-202N can (optionally) be stored in a central location for ease of network management and provided to the node 200 for local storage and use. A cost function is thus provided by a user to establish importance of certain parameters. In another example, a cost function may be based at least one of bandwidth, load, delay, reliability and so forth parameters and the user may weight these parameters in a cost function. However, different types of data packets may not function efficiently in a communication network using only one particular cost function. For example, one can construct a generic cost function for mobile ad-hoc networks (MANET), such as:

$$F_{MANET} = \left[\frac{K1 \times (1 - \text{utilization}) \times \text{bandwidth}}{100000000} + \frac{K2}{\text{latency}} \right] \times \frac{K3}{\text{BER} + K4}$$

Then, a user will select a suite of K (henceforth described as a "vector") that applies differently depending on traffic class of the packet being routed. For example, consider two traffic streams, i.e., FTP and voice.

For FTP traffic, the user sets FTP's K -vector to (2,0,1,1) to weight bandwidth and load. Thus,

$$F_{FTP} = \left[\frac{2 \times (1 - \text{utilization}) \times \text{bandwidth}}{100000000} \right] \times \frac{1}{\text{BER} + 1}$$

For voice traffic, the K -vector can be set to (0,1,1,1) to weight its delay sensitiveness. Thus,

$$F_{Voice} = \frac{1}{\text{latency}} \times \frac{1}{\text{BER} + 1}$$

The K values of one traffic type would compromise the performance of the other traffic type because these different traffic types warrant different K -vector. As will be shown further herein, different types of data packets may function more efficiently in a network using a different cost function.

5 The links 252a, 252b provide data packets to a respective one of the forwarding engine 222a, 222b. The forwarding engines 222a, 222b, based on one or more values in a data packet determines the appropriate routing table to use (i.e., the appropriate cost function to use) and provides the data packet to the appropriate egress port 226a, 226b. The egress ports 226a, 226b provide data packets to a respective link 262a, 262b.

10 Referring to FIG. 3, an example of a process to route data packets is a process 300. Process 300 receives a data packet (302). For example, the router 200 receives a packet from one of the links 252a, 252b.

Process 300 determines a value(s) from the data packet (308). For example, the forwarding engine 222a determines a value(s) from the data packet. In one example, the value corresponds to a traffic class in the header of the data packet. In one particular example, the value is a Differentiated Services (DiffServ) code point (DSCP) value. DiffServ uses a 6-bit Differentiated Services Field (DS field) in the IP header for packet classification purposes which generates up to 64 (2^6) values. Thus, there may be up to 64 routing tables using a 6-bit Differentiated Services Field as the value. Other values may include, but are not limited to, a port number or ID, source-destination pair, and so forth.

Process 300 selects a routing table based on the value(s) from the data packet (314). For example, the forwarding engine 222a selects a routing table based on the DSCP value in the data packet. Each routing table corresponds to one cost function and each entry in the table describes the best route for a given destination address (for that

25

particular traffic type). In some examples, there may exist multiple best routes in the table for a given destination if there are equally good.

Process 300 determines a destination address from a header of the data packet (322). For example, the forwarding engine 222a determines a destination address by
5 using the destination address in the IP header of the data packet.

Process 300 selects the egress port from the selected routing table based on the destination address (330). For example, the forwarding engine 222a selects one of the egress ports 226a, 226b by looking up the destination address in the selected routing table.

10 Process 300 forwards the data packet to the selected egress port (338). For example, the forwarding engine 222a forwards the data packet the selected egress ports.

Referring to FIG. 4, in one example, a routing node 200' includes a processor 402, a volatile memory 404, a non-volatile memory 406 (e.g., hard disk) and the user interface (UI) 408 (e.g., a graphical user interface, a mouse, a keyboard, a display, touch
15 screen and so forth). The non-volatile memory 406 stores computer instructions 412, an operating system 416 and data 418 such as cost functions 422 and routing tables 428. In one example, the computer instructions 412 are executed by the processor 402 out of volatile memory 404 to perform all or part of the processes described herein (e.g., process 300).

20 The processes described herein (e.g., process 300) are not limited to use with the hardware and software of FIG. 4; they may find applicability in any computing or processing environment and with any type of machine or set of machines that is capable of running a computer program. The processes described herein may be implemented in hardware, software, or a combination of the two. The processes described herein may be
25 implemented in computer programs executed on programmable computers/machines that

each includes a processor, a non-transitory machine-readable medium or other article of manufacture that is readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code may be applied to data entered using an input device to perform
5 any of the processes described herein and to generate output information.

The system may be implemented, at least in part, via a computer program product, (e.g., in a non-transitory machine-readable storage medium such as, for example, a non-transitory computer-readable medium), for execution by, or to control the operation of, data processing apparatus (e.g., a programmable processor, a computer, or
10 multiple computers). Each such program may be implemented in a high level procedural or object-oriented programming language to work with the rest of the computer-based system. However, the programs may be implemented in assembly, machine language, or Hardware Description Language. The language may be a compiled or an interpreted language and it may be deployed in any form, including as a stand-alone program or as a
15 module, component, subroutine, or other unit suitable for use in a computing environment. A computer program may be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network. A computer program may be stored on a non-transitory machine-readable medium that is readable by a general or special purpose programmable
20 computer for configuring and operating the computer when the non-transitory machine-readable medium is read by the computer to perform the processes described herein. For example, the processes described herein may also be implemented as a non-transitory machine-readable storage medium, configured with a computer program, where upon execution, instructions in the computer program cause the computer to operate in
25 accordance with the processes. A non-transitory machine-readable medium may include

but is not limited to a hard drive, compact disc, flash memory, non-volatile memory, volatile memory, magnetic diskette and so forth but does not include a transitory signal *per se*.

The processes described herein are not limited to the specific examples described.

5 For example, the process 300 is not limited to the specific processing order of FIG. 3.

Rather, any of the processing blocks of FIG. 3 may be re-ordered, combined or removed, performed in parallel or in serial, as necessary, to achieve the results set forth above.

In some examples, multiple routing tables may be combined in to a single routing table. In these examples, value-to-route associations are incorporated (directly or
10 indirectly) into the combined routing table thereby enabling the appropriate route selection to be made.

The processing blocks (for example, in the process 300) associated with implementing the system may be performed by one or more programmable processors executing one or more computer programs to perform the functions of the system. All or
15 part of the system may be implemented as, special purpose logic circuitry (e.g., an FPGA (field-programmable gate array) and/or an ASIC (application-specific integrated circuit)). All or part of the system may be implemented using electronic hardware circuitry that include electronic devices such as, for example, at least one of a processor, a memory, programmable logic devices or logic gates.

20 Elements of different embodiments described herein may be combined to form other embodiments not specifically set forth above. Other embodiments not specifically described herein are also within the scope of the following claims.

What is claimed is:

25

1. A method comprising:

receiving a data packet at a routing node comprising a processor;

determining at least one value for the data packet;

selecting a routing table from a plurality of routing tables stored at the routing

5 node in response to the at least one value for the packet, each routing table associated with a respective one cost function; and

forwarding the data packet in response to the routing table selected.

2. The method of claim 1 wherein receiving a data packet at a routing node

10 comprising a processor comprises receiving the data packet at the routing node from a first link, and

wherein forwarding the data packet in response the value of the data packet comprises forwarding the data packet to a second link.

15 3. The method of claim 1, further comprising combining the plurality of routing tables into a combined table, the combined table incorporating value-to route associations, and

wherein selecting a routing table from a plurality of routing tables comprises selecting the combined table.

20

4. The method of claim 1 wherein determining the at least one value of the data packet comprises determining at least one value located in a header of the data packet.

5. The method of claim 4 wherein determining at least one value located in a header of the data packet comprises determining a Differentiated Services (DiffServ) code point (DSCP) value in the data packet.

5 6. The method of claim 4 wherein determining at least one value located in a header of the data packet comprises determining at least one of a port number value or ID, or a source-destination pair value.

7. A routing node, comprising:
10 electronic hardware circuitry configured to:
receive a data packet at a routing node;
determine at least one value for the data packet;
select a routing table from a plurality of routing tables stored at the
routing node in response to the at least one value for the packet, each routing
15 table associated with a respective one cost function; and
forward the data packet in response to the routing table selected.

8. The apparatus of claim 7 wherein the circuitry comprises at least one of a processor, a memory, a programmable logic device or a logic gate.

20 9. The apparatus of claim 7 wherein the circuitry to receive a data packet at a routing node comprising a processor comprises circuitry to receive the data packet at the routing node from a first link, and

wherein the circuitry to forward the data packet in response the value of the data
25 packet comprises circuitry to forward the data packet to a second link.

10. The apparatus of claim 7, further comprising circuitry to combine the plurality of routing tables into a combined table, the combined table incorporating value-to route associations, and

wherein the circuitry to select a routing table from a plurality of routing tables
5 comprises circuitry to select the combined table.

11. The apparatus of claim 7 wherein the circuitry to determine the at least one value of the data packet comprises circuitry to determine at least one value located in a header of the data packet.

10

12. The apparatus of claim 11 wherein the circuitry to determine the at least one value located in a header of the data packet comprises circuitry to determine a Differentiated Services (DiffServ) code point (DSCP) value in the data packet.

15

13. The apparatus of claim 11 wherein the circuitry to determine at least one value located in a header of the data packet comprises circuitry to determine at least one of a port number value or ID, or a source-destination pair value.

20

14. An article comprising:
a non-transitory computer-readable medium that stores computer-executable instructions, the instructions causing a machine to:

receive a data packet at a routing node;

determine at least one value for the data packet;

select a routing table from a plurality of routing tables stored at the routing node in response to the at least one value for the packet, each routing table associated with a respective one cost function; and

forward the data packet in response to the routing table selected.

5

15. The article of claim 14 wherein the instructions causing the machine to receive a data packet at a routing node comprising a processor comprises instructions causing the machine to receive the data packet at the routing node from a first link, and

wherein the instructions causing the machine to forward the data packet in response the value of the data packet comprises instructions causing the machine to forward the data packet to a second link.

10

16. The article of claim 14, further comprising instructions causing the machine to combine the plurality of routing tables into a combined table, the combined table incorporating value-to route associations, and

15

wherein the instructions causing the machine to select a routing table from a plurality of routing tables comprises instructions causing the machine to select the combined table.

20

17. The article of claim 14 wherein the instructions causing the machine to determine the at least one value of the data packet comprises instructions causing the machine to determine at least one value located in a header of the data packet.

18. The apparatus of claim 17 wherein the instructions causing the machine to determine the at least one value located in a header of the data packet comprises

25

instructions causing the machine to determine a Differentiated Services (DiffServ) code point (DSCP) value in the data packet.

19. The apparatus of claim 17 wherein the instructions causing the machine to
5 determine at least one value located in a header of the data packet comprises instructions causing the machine to determine at least one of a port number value or ID, or a source-destination pair value.

1/4

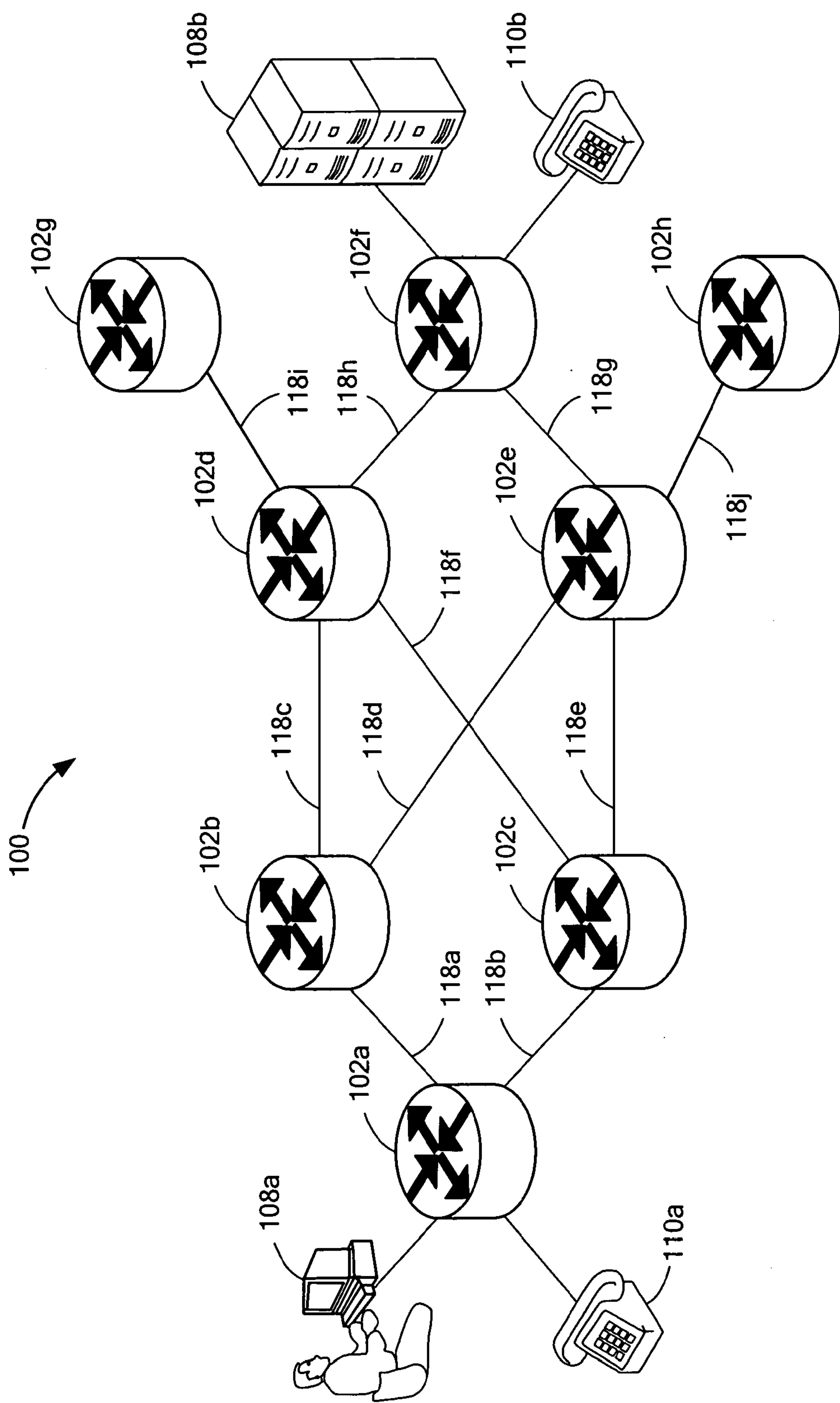
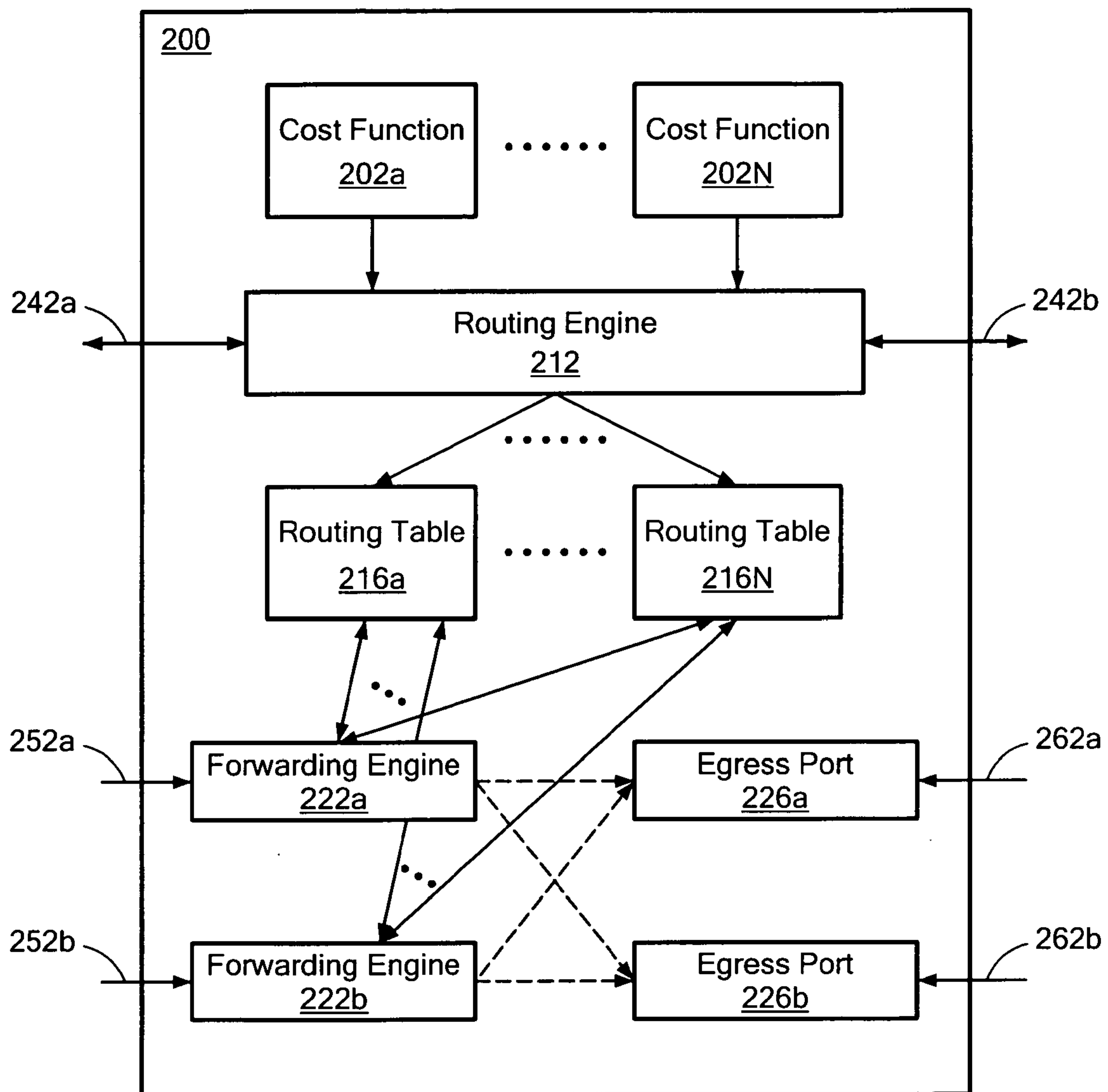
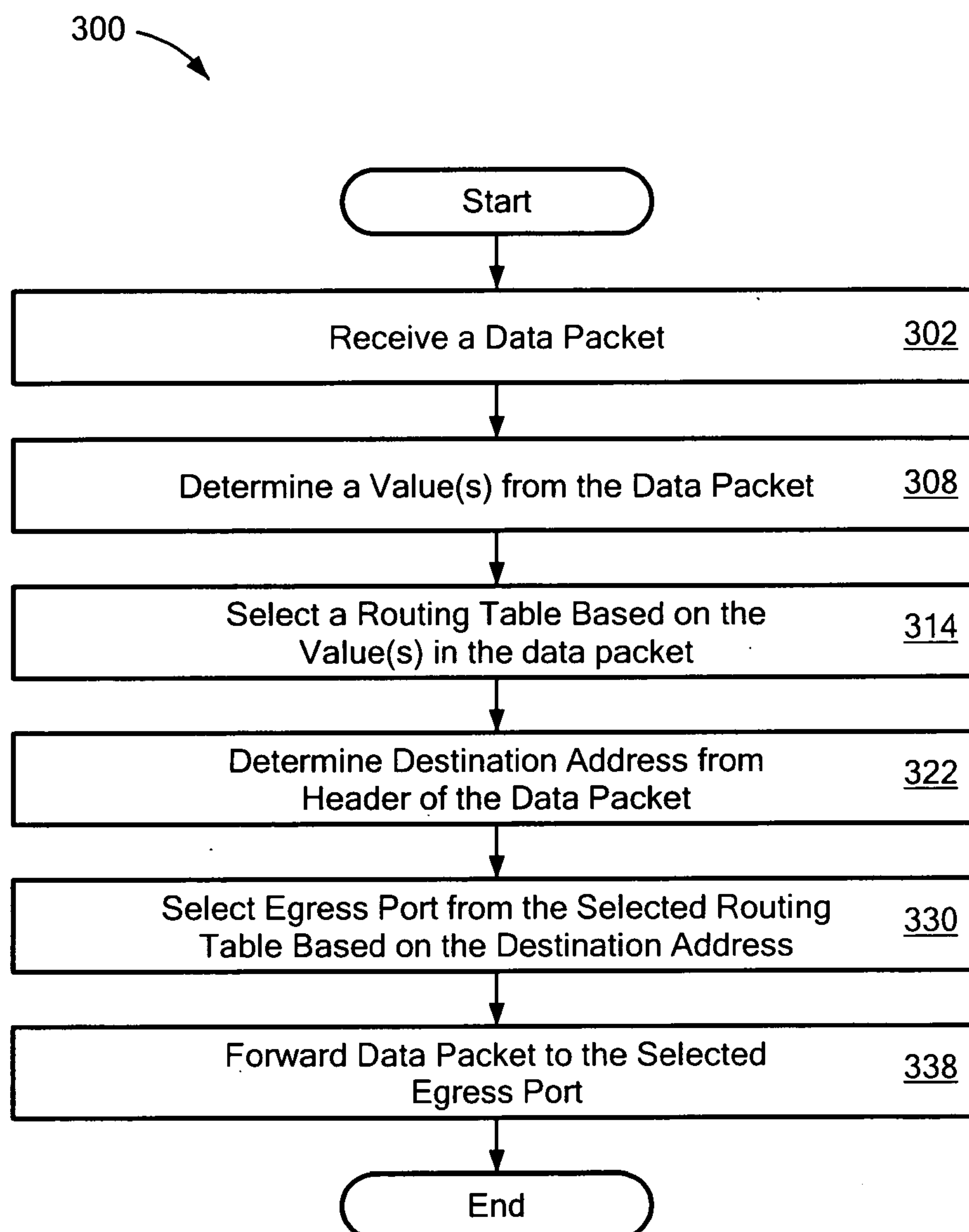


FIG. 1

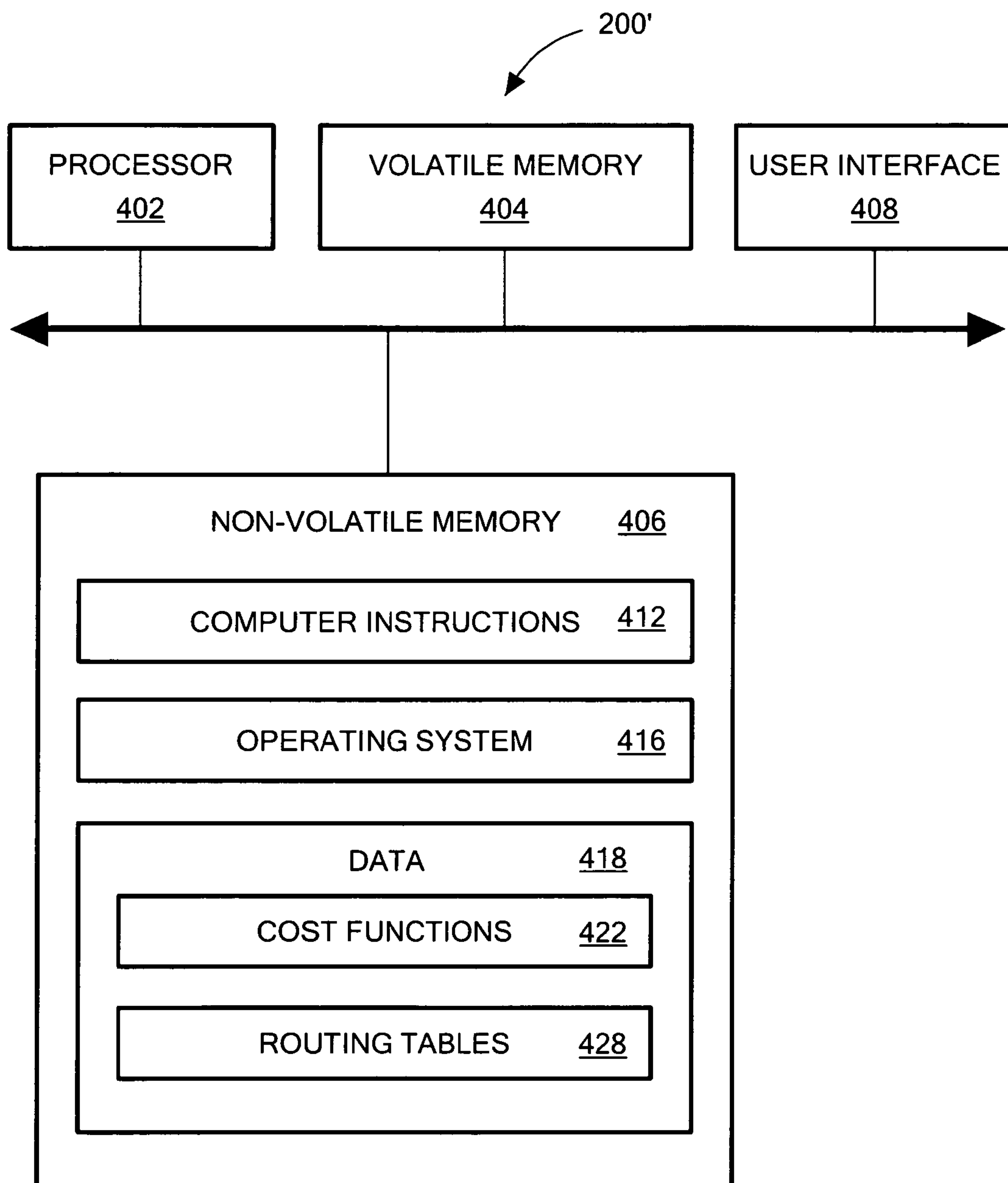
2/4

**FIG. 2**

3/4

**FIG. 3**

4/4

**FIG. 4**

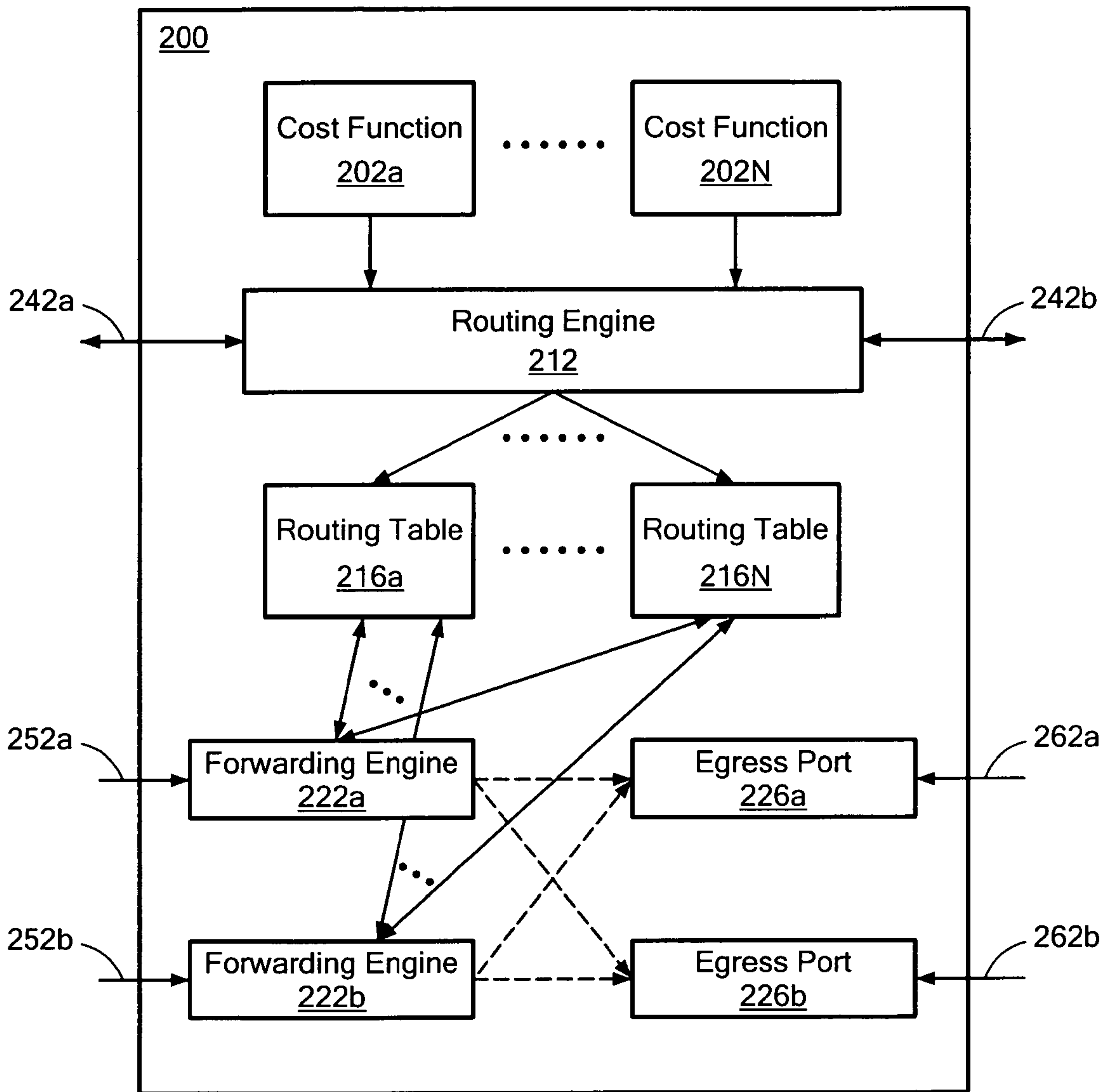


FIG. 2