

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
15 December 2005 (15.12.2005)

PCT

(10) International Publication Number
WO 2005/119971 A2

(51) International Patent Classification⁷: **H04L 12/28**

(21) International Application Number:
PCT/US2005/017463

(22) International Filing Date: 18 May 2005 (18.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/858,920 1 June 2004 (01.06.2004) US

(71) Applicant (for all designated States except US): **CISCO TECHNOLOGY, INC.** [US/US]; 170 W. Tasman Drive, San Jose, CA 95134-1706 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PREVIDI, Stefano Benedetto** [BE/IT]; Via del Serafico 200, I-00142 Rome (IT). **FILSFILS, Clarence** [BE/BE]; DeKleetlaan 6, B-1831 Diegem (BE). **VASSEUR, Jean-Philippe** [FR/US]; 300 Beaver Brook Road, Boxborough, MA 01719 (US).

(74) Agents: **PALERMO, Christopher, J.** et al.; Hickman Palermo Truong & Becker LLP, 2055 Gateway Place, Suite 550, San Jose, CA 95110-1089 (US).

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

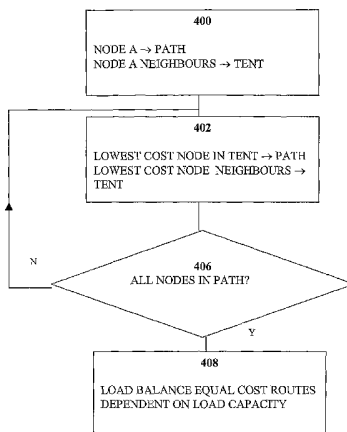
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR FORWARDING DATA IN A DATA COMMUNICATIONS NETWORK



(57) Abstract: A method is disclosed of forwarding data in a data communications network comprising a plurality of nodes and links therebetween, each link having an associated usage cost and a load capacity. A forwarding node identifies a lowest cost route, comprising one or more links, to a destination node. If a plurality of lowest cost routes is identified, then the forwarding node allocates data for forwarding along each of the lowest cost routes dependent on a relative load capacity of that route.

METHOD AND APPARATUS FOR FORWARDING DATA IN A DATA COMMUNICATIONS NETWORK

FIELD OF THE INVENTION

[0001] The present invention generally relates to forwarding data. The invention relates more specifically to a method and apparatus for forwarding data in a data communications network.

BACKGROUND OF THE INVENTION

[0002] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

[0003] In computer networks such as the Internet, packets of data are sent from a source to a destination via a network of elements including links (communication paths such as telephone or optical lines) and nodes (usually routers directing the packet along one or more of a plurality of links connected to it) according to one of various routing protocols.

[0004] One class of routing protocol is the link state protocol. The link state protocol relies on a routing algorithm resident at each node. Each node on the network advertises, throughout the network, links to neighboring nodes and provides a cost associated with each link, which can be based on any appropriate metric such as link bandwidth or delay and is typically expressed as an integer value. A link may have an asymmetric cost, that is, the cost in the direction AB along a link may be different from the cost in the direction BA. Based on the advertised information in the form of a link state packet (LSP) each node constructs a link state database (LSDB), which is a map of the entire network topology and from that constructs generally a single optimum route to each available node based on an appropriate algorithm such as, for example, a shortest path first (SPF) algorithm. As a result a "spanning tree" (SPT) is constructed, rooted at the node and showing an optimum path including intermediate nodes to each available destination node. The results of the SPF are stored in a routing information base (RIB) and based on these results the forwarding information base (FIB) or forwarding table is updated to control forwarding of packets appropriately. When there is a network change an LSP

representing the change is flooded through the network by each node adjacent the change, each node receiving the LSP and sending it to each adjacent node.

[0005] As a result, when a data packet for a destination node arrives at a node (the "first node"), the first node identifies the optimum route to that destination and forwards the packet to the next node along that route. The next node repeats this step and so forth.

[0006] Generally, data is forwarded along a single link to an adjacent node but in some instances an "equal cost path split" occurs in which two or more equal lowest cost routes are available. In that case the forwarding node will implement load balancing whereby the load is shared equally between the links.

[0007] Problems arise in networks when congestion occurs in parts of the network. A known solution to the problem is to avoid such congestion by engineering the costs of the links to reflect their capacity. One such approach is described in B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in Proc. IEEE INFOCOM, pp. 519-528, 2000 ("*Thorup et al*") in which the cost of a link is inversely proportional to its capacity or bandwidth as a result of which less traffic is routed over low capacity links.

[0008] However problems arise with the approach set out in *Thorup et al.* In particular load will either be spread evenly between equal cost routes or not at all in that arrangement.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0010] FIG. 1 is a representation of a network illustrating a method of forwarding data;

[0011] FIG. 2 is a representation of an alternative network illustrating a method of forwarding data;

[0012] FIG. 3 is a representation of a network illustrating in more detail a method of forwarding data;

[0013] FIG. 4 is a flow diagram that illustrates the steps involved in forwarding data according to the method;

[0014] FIG. 5 is a representation of an alternative network illustrating in more detail a method of forwarding data; and

[0015] FIG. 6 is a block diagram illustrating a computer system upon which a method of forwarding data may be implemented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0016] A method and apparatus for forwarding data in a data communication network is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0017] Embodiments are described herein according to the following outline:

- 1.0 General Overview
- 2.0 Structural and Functional Overview
- 3.0 Method of Forwarding Data
- 4.0 Implementation Mechanisms—Hardware Overview
- 5.0 Extensions and Alternatives

* * *

1.0 GENERAL OVERVIEW

[0018] The needs identified in the foregoing Background, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises, in one aspect, a method of forwarding data in a data communications network comprising a plurality of nodes and links therebetween, each link having an associated usage cost and a load capacity. A forwarding node identifies a lowest cost route, comprising one or more links, to a destination node. If a plurality of lowest cost routes is identified the forwarding node allocates data for forwarding along each of the lowest cost routes dependent on a relative load capacity of that route.

[0019] In other aspects, the invention encompasses a computer apparatus and a computer-readable medium configured to carry out the foregoing steps.

2.0 STRUCTURAL AND FUNCTIONAL OVERVIEW

[0020] In overview a method of forwarding data can be understood with reference to FIG. 1 which depicts an illustrative network diagram to which the method is applied. The network includes node A (reference numeral 12), node B (reference numeral 14) and node Z (reference numeral 16). Nodes A and B are joined by link 18, nodes A and Z are joined by link 20 and nodes B and Z are joined by link 22. Link 18 is connected at an interface j to node A and link 20 is connected at an interface i to node A. The cost of links 18 and 22 is 1 and the cost of link 20 is 2. All links have a load capacity or bandwidth of 10GB.

[0021] It will be seen that the cost of forwarding data from node A to node Z is the same whether the route is via link 20 (cost 2) or via links 18 and 22 and node B (cost $1+1=2$). As a result the load is shared between the two routes AZ, ABZ and as both routes have an equal bandwidth of 10GB congestion is minimized.

[0022] Referring to FIG. 2 which depicts an alternative illustrative network diagram to which the method is applied, it will be seen that the nodes, links, costs and load capacities are the same as in FIG. 1 except that node B and node Z are connected by link 24 having a bandwidth of 2.5GB. Once again the cost of routing data from node A to node Z is equal whether via link 20 or via links 18 and 24. However in this case the route ABZ via links 18 and 24 has a minimum bandwidth of 2.5GB which is thus taken as the route bandwidth. Thus if the load was shared evenly between the two routes then congestion of the lower bandwidth route could occur at the same time as under use of the higher bandwidth route.

[0023] Accordingly the method described herein allocates the load proportionally with bandwidth. In the specific case shown in FIG. 2 the load is shared in the ratio 10GB:2.5GB i.e. in the ratio 80:20 between route AZ and route ABZ. As a result weighted fair load balancing is achieved. In addition this is achieved using existing routes computed according to the SPF and hence avoiding looping.

3.0 METHOD AND APPARATUS FOR FORWARDING DATA

[0024] Implementation of the method will be described in more detail with reference to the SPF algorithm (also known as Dijkstra's algorithm). The algorithm is well understood by the skilled reader and well documented in pre-existing documentation and therefore is not described in detail here but only in summary, for the purposes of clarity.

[0025] In general a node, say node A, calculates its least cost (shortest path) routes to each other node on the network from its LSDB. To do this, all nodes on the network are divided into three categories, unknown (UNK), tentative or candidate (TENT) and known (PATHS). For each node the cost of reaching that node from node A, and the interface along which it is reached at that cost (representing the next hop) are stored. Initially node A is assigned to category TENT and all other nodes are assigned to UNK. The algorithm then carries out repeatedly the step of moving the lowest cost node in TENT to PATH and moving all neighbors of that lowest cost node together with their cost via the corresponding interface from node A to TENT. As a result, in the first step node A is moved to PATH and each of its neighbors to TENT. In the next step the lowest cost node in TENT is moved into PATH and its neighbors are moved into TENT. As a result the nodes in UNK and TENT gradually deplete until all nodes have been moved to PATH.

As a result PATH comprises a list of the lowest cost interface (or next hop) for routing data from node A to each other node.

[0026] Reference is now made to FIG. 3 which is a network diagram illustrating a network to which the method described herein can be applied and FIG. 4 which is a flow diagram illustrating in more detail implementation of the method. Implementation of the SPF algorithm to achieve weighted fair load balancing according to the present method can be better understood with respect to FIG. 3 and FIG. 4 in the case of matched bandwidth routes.

[0027] In particular FIG. 3 shows a network similar to that of FIG. 1 and FIG. 2 but in which an additional node C, reference numeral 26, is introduced between nodes A and Z, joined to node Z via link 28 having cost 1 and capacity 10GB. The cost of link 20 from interface i of node A to node C is reduced to 1. In addition node B has two interfaces k, l both connecting node B to node Z via respective links 22, cost 1, bandwidth 10GB and 24, cost 1, bandwidth 2.5GB.

[0028] In order to calculate the SPF at node A, the algorithm described above is followed but with the introduction of an extra parameter associated with a node in UNK/TENT/PATH “minimum bandwidth along the best path to this node”.

[0029] The node is thus categorized X(a, b, c) where:

X is the node name

a is the root node's outgoing interface used by the best path from the root node to X

b is the minimum link bandwidth across that path

c is the path metric (cost) of that path.

[0030] It will be appreciated that the root node is the node at which the SPF computation is taking place – in the present case, node A. As will be discussed in more detail below, where multiple paths sharing the same root node's outgoing interface share the same metric, the “b” term (capacity) of the resulting path is the maximum of the “b” terms of the underlying paths.

[0031] Implementation of the algorithm can be understood with reference to FIG. 4 in conjunction with FIG. 3. In block 400 a first node under consideration, such as node A, is transferred to PATH, and the neighbors of the first node are transferred to TENT together with their associated parameters as shown in Table 1 below. As a result the table is as shown below in Table 1 in which, for example, it can be seen that node C is shown as reachable via interface i with a bandwidth of 10GB and cost 1. The remaining nodes are maintained in UNK.

Table 1

UNK	TENT	PATH
Z	C (i, 10G, 1) B (j, 10G, 1)	A

[0032] In block 402 the lowest cost node in TENT is once again transferred to PATH and its neighbors are transferred to TENT. Referring to Table 2 below it will be seen that as a result node C has been moved to PATH as a result of which the least cost route to node C from A is known to be via interface i at cost 1, bandwidth 10GB. Node Z has been transferred to TENT together with the corresponding bandwidth and cost via interface i.

Table 2

UNK	TENT	PATH
-	B (j, 10G, 1) Z (i, 10G, 2)	A C (i, 10G, 1)

[0033] In block 406 the algorithm assesses whether all nodes have been transferred to PATH. If not then step 402 is repeated. Otherwise in block 408 the forwarding node (or not node), node A, carries out load balancing proportional to the bandwidth ratio as discussed in more detail below.

[0034] In relation to the algorithm computed at node A in relation to the network of FIG. 3, therefore, step 402 is repeated and node B is transferred to PATH. As a result an alternative route to Z via interface j (and hence node B) is introduced into TENT as can be seen in Table 3.

Table 3

UNK	TENT	PATH
-	Z (i, 10G, 2) or Z (j, 10G, 2)	A C (i, 10G, 1) B (j, 10G, 1)

[0035] Block 402 is repeated once again and in this case node Z is transferred to PATH by either of interfaces i or j as the cost is the same in each case, namely 2. However as the bandwidth via either route ACZ (interface i) or ABZ (interface j) is the same, then the load is spread 50/50 between the interfaces as shown in Table 4.

Table 4

UNK	TENT	PATH
-	-	A C (i, 10G, 1) B (j, 10G, 1) Z (i, 10G, 2) or Z (j, 10G, 2)

[0036] It will be noted that the link between nodes B and Z is assigned the higher of the two available bandwidths via interface k (link 24) and interface l (link 22), namely 10GB. It will be further noted that node B will itself implement weighted load balancing between interfaces k and l proportional to their respective capacities, that is in ratio 80/20 as discussed in more detail below. Accordingly, in an alternative implementation, the link between nodes B and Z can be assigned the sum of the bandwidth via each interface sharing the same cost metric, namely 12 GB and the remaining steps herein carried out accordingly. Of course if the links carry different costs, then only the lower cost and associated bandwidth are assigned.

[0037] Implementation of the method in an instance where alternative routes have unequal capacities can be better understood with reference to FIG. 5 which is a network diagram illustrating an alternative network. FIG. 5 corresponds effectively to an overlay of the networks of FIG. 1 and FIG. 2 but with the addition of a node D, reference numeral 30 connected between node B and node Z by link 32 having cost 1 and capacity 2.5GB. Node B is linked to node D via links 22 and 24 of costs 1 and respective bandwidths 10GB, 2.5GB. The cost of link 20 between nodes A and Z is 3.

[0038] Once again implementing the algorithm described with reference to FIG. 4, in block 400 node A is moved to PATH and nodes Z and B together with the associated interfaces, bandwidths and costs are moved to TENT as shown in Table 5 below.

Table 5

UNK	TENT	PATH
D	Z (i, 10G, 3) B (j, 10G, 1)	A

[0039] In block 402 the lowest cost node, node B is moved from TENT to PATH and node D is moved from UNK to TENT. As two links are provided from node B to node D, the maximum bandwidth represented, i.e. 10G, is associated with node D in TENT as shown in Table 6.

Table 6

UNK	TENT	PATH
-	Z (i, 10G, 3) D (j, 10G, 2)	A B (j, 10G, 1)

[0040] Block 402 is once again repeated and node D transferred to PATH. As a result node Z via interface j is added to TENT such that both routes to node Z are provided in TENT either via interface i (route AZ) or interface j (route ABDZ) as can be seen in Table 7 below.

Table 7

UNK	TENT	PATH
-	Z (i, 10G, 3) or Z (j, 2.5G, 3)	A B (j, 10G, 1) D (j, 10G, 2)

[0041] Block 402 is repeated once more and both routes to node Z are transferred to PATH as shown in Table 8 below.

Table 8

UNK	TENT	PATH
-	-	A B (j, 10G, 1) D (j, 10G, 3) Z (i, 10G, 3) or Z (j, 2.5G, 3)

[0042] As a result node A will load balance packets for node Z via the two possible routes AZ (interface i), ABDZ (interface j), but proportional to the ratio of their bandwidth. Accordingly 80% of traffic will go via node AZ and 20% via ABDZ and the traffic is assigned to the interfaces accordingly in ratio 10GB:2.5GB i.e. 80/20. Once again node B will implement load balancing itself between interfaces k and l in the ratio of the respective capacities of the links.

[0043] As a result of the method described herein network capacity is maximized without affecting the SPF calculations nor introducing potential loops.

[0044] It will be appreciated that the method can be implemented in one or more of hardware or software elements, or firmware using for example microcode, or any combination thereof, such that detailed explanation is not required by the skilled reader. Once the SPF algorithm has been completed and the ratios assigned to the respective

interfaces, allocation of the data between the interfaces can be achieved in any appropriate manner, for example as described in “Troubleshooting Load Balancing Over Parallel Links using Cisco Express Forwarding” which is available at the time of this writing on the file “technologies_tech_note09186a0080094806.shtml” in the directory “en/US/tech/tk827/tk831” of the domain cisco.com of the World Wide Web, the entire contents of which are incorporated herein by reference for all purposes as if set forth fully herein.

4.0 IMPLEMENTATION MECHANISMS -- HARDWARE OVERVIEW

[0045] FIG. 6 is a block diagram that illustrates a computer system 140 upon which the method may be implemented. The method is implemented using one or more computer programs running on a network element such as a router device. Thus, in this embodiment, the computer system 140 is a router.

[0046] Computer system 140 includes a bus 142 or other communication mechanism for communicating information, and a processor 144 coupled with bus 142 for processing information. Computer system 140 also includes a main memory 146, such as a random access memory (RAM), flash memory, or other dynamic storage device, coupled to bus 142 for storing information and instructions to be executed by processor 144. Main memory 146 may also be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 144. Computer system 140 further includes a read only memory (ROM) 148 or other static storage device coupled to bus 142 for storing static information and instructions for processor 144. A storage device 150, such as a magnetic disk, flash memory or optical disk, is provided and coupled to bus 142 for storing information and instructions.

[0047] A communication interface 158 may be coupled to bus 142 for communicating information and command selections to processor 144. Interface 158 is a conventional serial interface such as an RS-232 or RS-422 interface. An external terminal 152 or other computer system connects to the computer system 140 and provides commands to it using the interface 158. Firmware or software running in the computer system 140 provides a terminal interface or character-based command interface so that external commands can be given to the computer system.

[0048] A switching system 156 is coupled to bus 142 and has an input interface and a respective output interface (commonly designated 159) to external network elements. The external network elements may include a plurality of additional routers 160 or a local network coupled to one or more hosts or routers, or a global network such as the Internet having one or more servers. The switching system 156 switches information traffic

arriving on the input interface to output interface 159 according to pre-determined protocols and conventions that are well known. For example, switching system 156, in cooperation with processor 144, can determine a destination of a packet of data arriving on the input interface and send it to the correct destination using the output interface. The destinations may include a host, server, other end stations, or other routing and switching devices in a local network or Internet.

[0049] The computer system 140 implements as a router acting as a forwarding node the above described method of forwarding data. The implementation is provided by computer system 140 in response to processor 144 executing one or more sequences of one or more instructions contained in main memory 146. Such instructions may be read into main memory 146 from another computer-readable medium, such as storage device 150. Execution of the sequences of instructions contained in main memory 146 causes processor 144 to perform the process steps described herein. One or more processors in a multi-processing arrangement may also be employed to execute the sequences of instructions contained in main memory 146. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the method. Thus, embodiments are not limited to any specific combination of hardware circuitry and software.

[0050] The term “computer-readable medium” as used herein refers to any medium that participates in providing instructions to processor 144 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 150. Volatile media includes dynamic memory, such as main memory 146. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 142. Transmission media can also take the form of wireless links such as acoustic or electromagnetic waves, such as those generated during radio wave and infrared data communications.

[0051] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0052] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 144 for execution. For example,

the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 140 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector coupled to bus 142 can receive the data carried in the infrared signal and place the data on bus 142. Bus 142 carries the data to main memory 146, from which processor 144 retrieves and executes the instructions. The instructions received by main memory 146 may optionally be stored on storage device 150 either before or after execution by processor 144.

[0053] Interface 159 also provides a two-way data communication coupling to a network link that is connected to a local network. For example, the interface 159 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, the interface 159 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, the interface 159 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0054] The network link typically provides data communication through one or more networks to other data devices. For example, the network link may provide a connection through a local network to a host computer or to data equipment operated by an Internet Service Provider (ISP). The ISP in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet". The local network and the Internet both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on the network link and through the interface 159, which carry the digital data to and from computer system 140, are exemplary forms of carrier waves transporting the information.

[0055] Computer system 140 can send messages and receive data, including program code, through the network(s), network link and interface 159. In the Internet example, a server might transmit a requested code for an application program through the Internet, ISP, local network and communication interface 158. One such downloaded application provides for the method as described herein.

[0056] The received code may be executed by processor 144 as it is received, and/or stored in storage device 150, or other non-volatile storage for later execution. In this manner, computer system 140 may obtain application code in the form of a carrier wave.

5.0 EXTENSIONS AND ALTERNATIVES

[0057] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0058] It will be appreciated that the method can be implemented in relation to any size or type of network and that any appropriate routing protocol and mechanism can be adopted to implement the invention. For example it will be appreciated that any appropriate routing protocol can be used such as Intermediate System – Intermediate System (IS-IS) or Open Shortest Path First (OSPF) and in networks using any appropriate forwarding mechanism such as Internet Protocol (IP). Indeed the invention can be implemented using any least cost routing technique. The method steps set out can be carried out in any appropriate order and aspects from the examples and embodiments described juxtaposed or interchanged as appropriate.

[0059] Furthermore it will be appreciated that although in an optimization the approach described above is implemented at each node in a network it can be implemented at only one or a sub-set of all of the nodes with equal effect and without affecting routing across the network adversely.

CLAIMS

What is claimed is:

- 1 1. A method of forwarding data in a data communications network comprising a
2 plurality of nodes and links therebetween, each link having an associated usage cost and a
3 load capacity, the method comprising the steps, performed at a forwarding node, of:
4 identifying a lowest cost route, comprising one or more links, to a destination
5 node; and
6 if a plurality of lowest cost routes is identified, then allocating data for forwarding
7 along each of the lowest cost routes dependent on a relative load capacity
8 of that route.
- 1 2. A method as claimed in Claim 1 in which the load capacity of a particular route
2 among the lowest cost routes is selected as the lowest load capacity of any link along the
3 particular route.
- 1 3. A method as claimed in Claim 1 in which, where a plurality of links connects
2 adjacent nodes, the load capacity between those nodes is set as the maximum load
3 capacity of any one of those links.
- 1 4. A method as claimed in claim 1 in which, where a plurality of links connects
2 adjacent nodes, the load capacity between these nodes is set as the sum of the load
3 capacity of each of these links.
- 1 5. A method as claimed in Claim 1 in which data is allocated proportionally to a
2 ratio of load capacities of each of the lowest cost routes.
- 1 6. A method as claimed in Claim 1 in which each of the lowest cost routes is
2 identified using a shortest path first algorithm.
- 1 7. A method as claimed in Claim 6 in which load capacity is included as a parameter
2 associated with each node interface in the shortest path first algorithm.

1 8. A computer readable medium comprising one or more sequences of instructions
2 for forwarding data in a data communications network which when executed by one or
3 more processors, cause the one or more processors to perform the steps of the method of
4 any of claims 1, 2, 3, 4, 5, 6 or 7.

1 9. An apparatus for forwarding data in a data communications network comprising a
2 plurality of nodes and links therebetween, each linking having an associated usage cost
3 and a load capacity, the apparatus comprising:
4 means for identifying a lowest cost route, comprising one or more links, to a
5 destination node; and
6 means for allocating data for forwarding along each of the lowest cost routes
7 dependent on a relative load capacity of that route if a plurality of lowest
8 cost routes is identified.

1 10. An apparatus as claimed in claim 9, in which the load capacity of a particular
2 route among the lowest cost routes is selected as the lowest load capacity of any link
3 along the particular route.

1 11. An apparatus as claimed in claim 9 in which, where a plurality of links connects
2 adjacent nodes, the load capacity between those nodes is set as the maximum load
3 capacity of any one of those links.

1 12. An apparatus as claimed in claim 9 in which, where a plurality of links connects
2 adjacent nodes, the load capacity between those nodes is set as the sum of the load
3 capacity of each of those links.

1 13. An apparatus as claimed in claim 9 in which the means for allocating data
2 allocates data proportionally to a ratio of load capacities of each of the lowest cost routes.

1 14. An apparatus as claimed in claim 9 in which the means for identifying a lowest
2 cost route uses a shortest path first algorithm.

1 15. An apparatus as claimed in claim 14 in which load capacity is included as a
2 parameter associated with each node interface in the shortest path first algorithm.

1 16. An apparatus for forwarding data in a data communications network, the
2 apparatus comprising:
3 one or more processors; and
4 a network interface communicatively coupled to the one or more processors and
5 configured to communicate one or more packet flows among the one or
6 more processors and a network and a computer readable medium
7 comprising one or more sequences of instructions for forwarding data in a
8 data communications network which when executed by one or more
9 processors, cause the one or more processors to perform the steps of the
10 method of any of claims 1, 2, 3, 4, 5, 6 or 7.

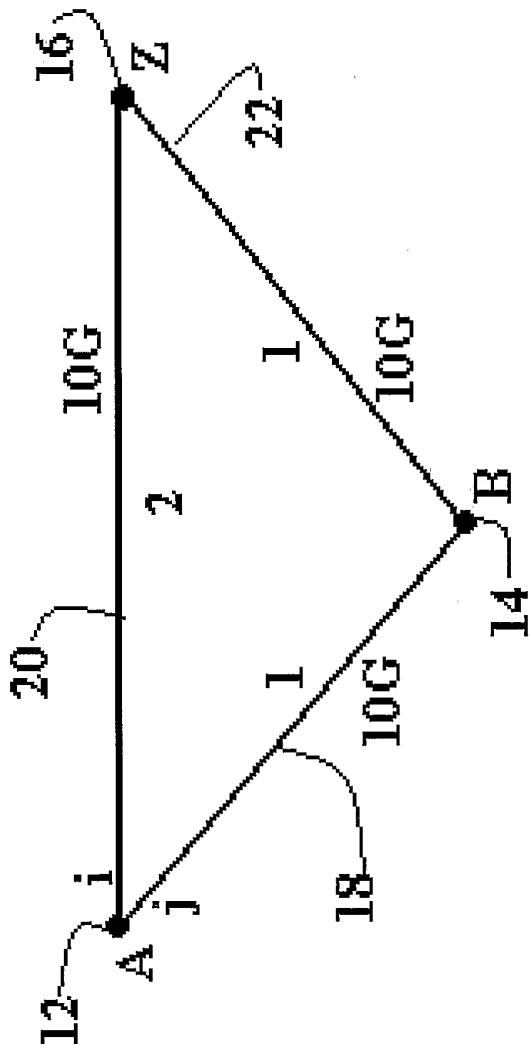


FIG. 1

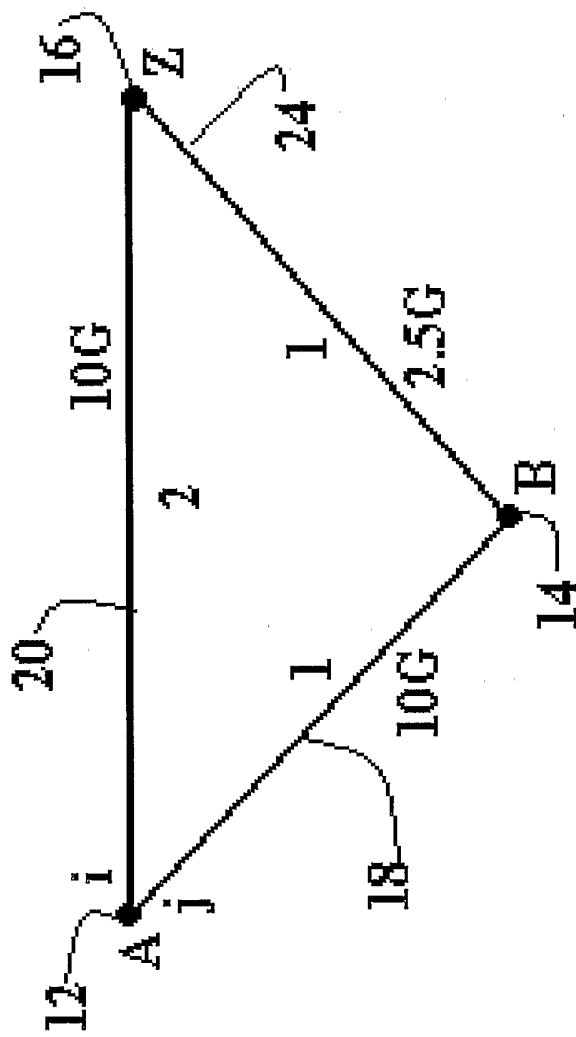


FIG. 2

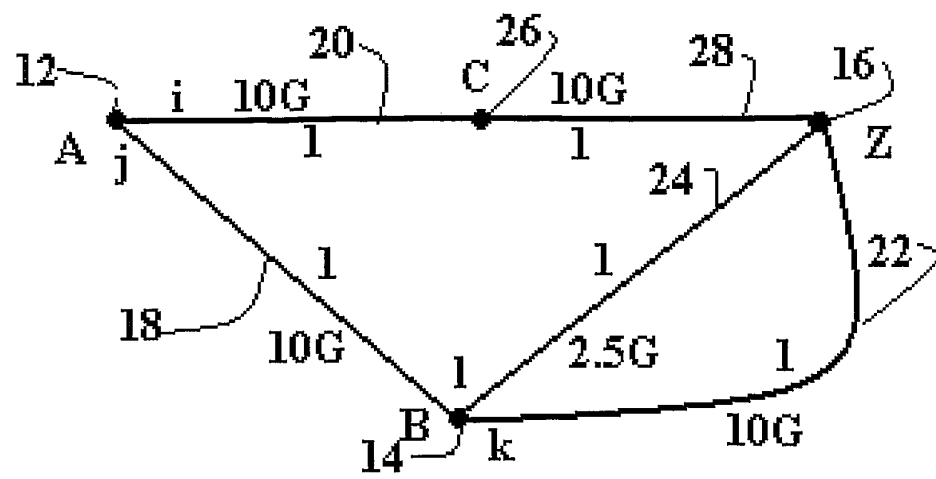


FIG. 3

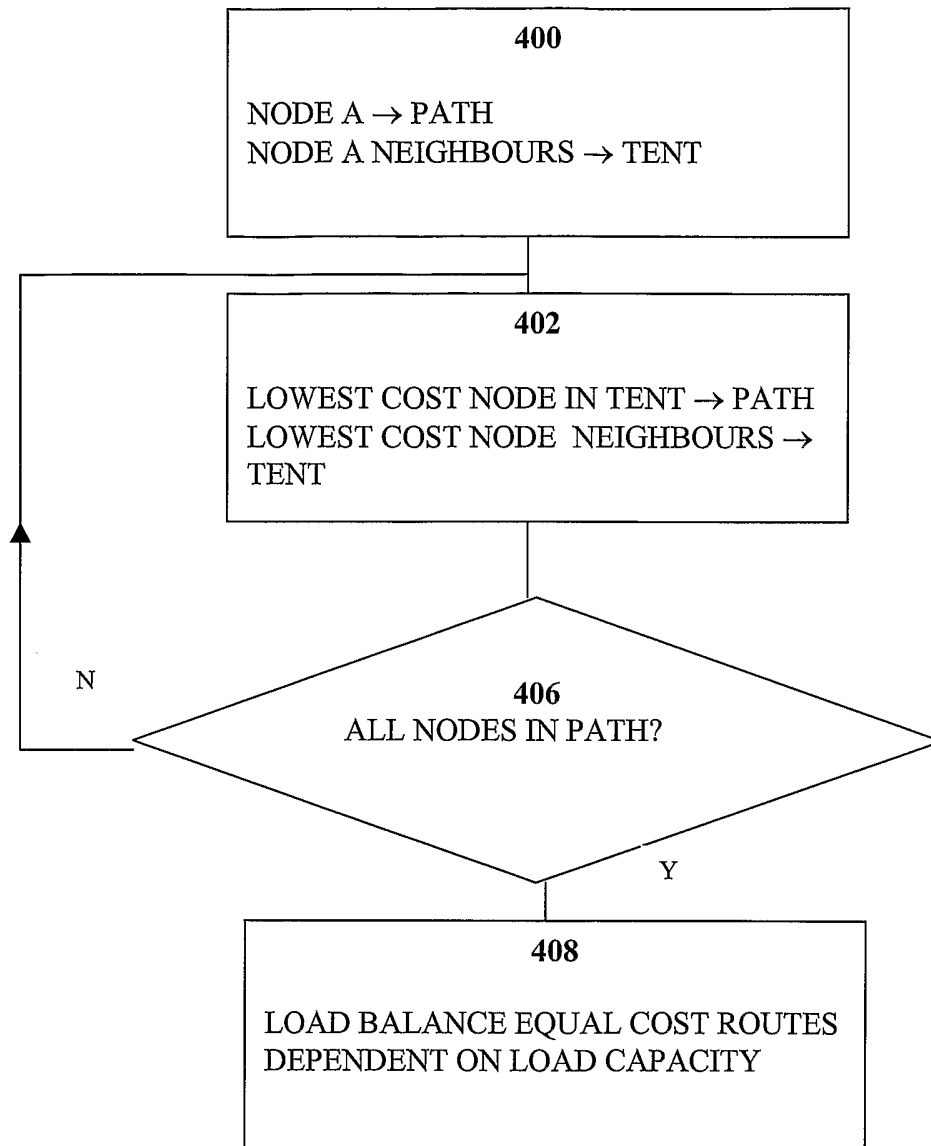


FIG. 4

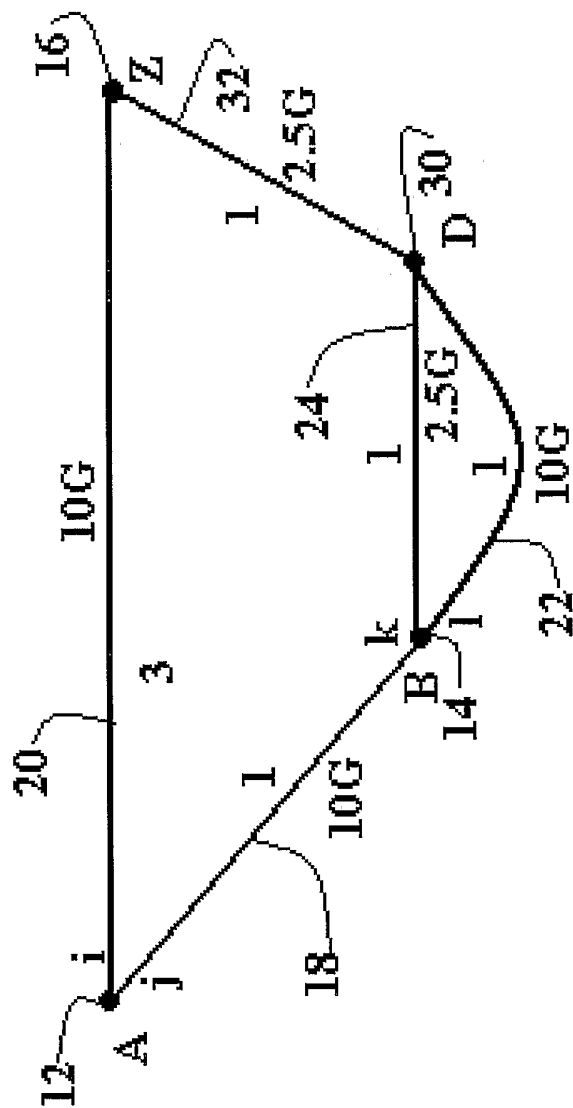


FIG. 5

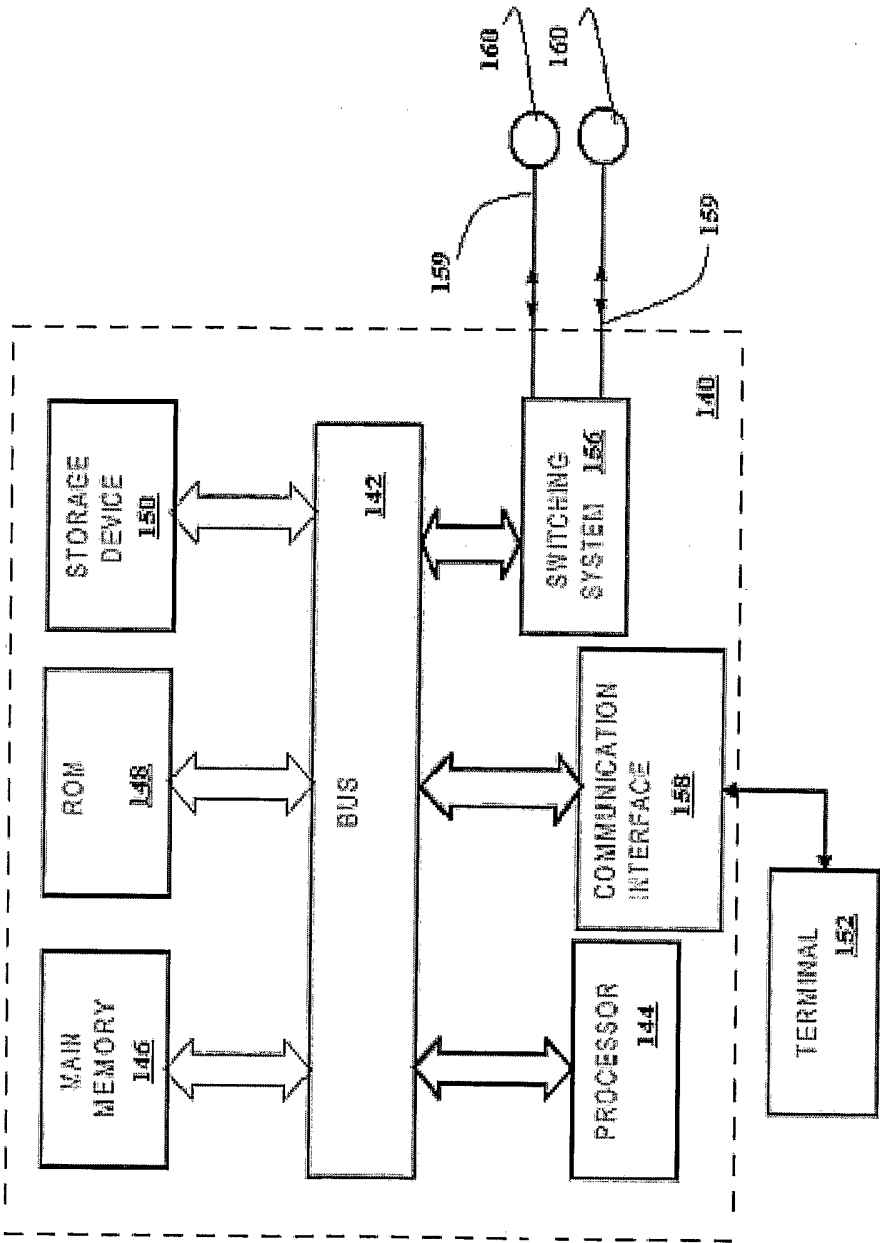


FIG. 6