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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
08 December 2022 (08.12.2022)

WIPO | PCT

(10) **International Publication Number**
WO 2022/253437 A1

(51) **International Patent Classification:**
H04L 45/121 (2022.01) *H04L 45/64* (2022.01)
H04L 45/12 (2022.01)

(21) **International Application Number:**
PCT/EP2021/064888

(22) **International Filing Date:**
02 June 2021 (02.06.2021)

(25) **Filing Language:** English

(26) **Publication Language:** English

(71) **Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) **Inventors: FIASCHI, Giovanni**; Vaxmoravägen 21A, SE-192 48 SOLLENTUNA (SE). **LIN, Yun**; No 5 Lize East Street Chaoyang District, Beijing, 100102 (CN). **NORDENSKY, Patrik**; Söndagsvägen 76, 123 60 Farsta (SE). **CECCARELLI, Daniele**; Strandvägen 4 D, SE-191 35 SOLLENTUNA (SE).

(74) **Agent: ERICSSON**; Patent Development Torshamnsgatan 21-23, 164 80 STOCKHOLM (SE).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ST, 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, KM, ML, MR, NE, SN, TD, TG).

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

(54) **Title:** METHODS, APPARATUS AND COMPUTER-READABLE MEDIA FOR PATH ROUTING AND TRAFFIC ALLOCATION IN A COMMUNICATION NETWORK

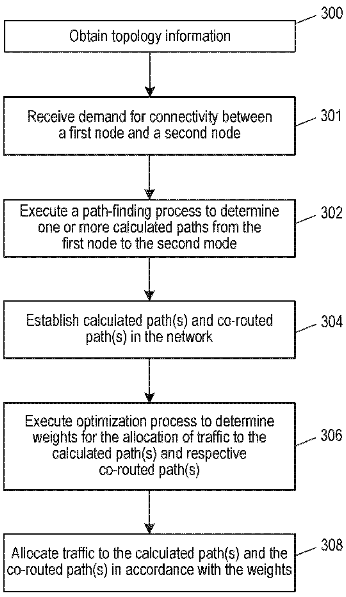

Fig. 3

(57) **Abstract:** A method is disclosed for configuring a communication network. The communication network comprises a plurality of nodes, with pairs of nodes being connected by respective links. Each link is associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow. The method comprises: receiving a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node; executing a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions; establishing the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and allocating the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

WO 2022/253437 A1

METHODS, APPARATUS AND COMPUTER-READABLE MEDIA FOR PATH ROUTING AND TRAFFIC ALLOCATION IN A COMMUNICATION NETWORK

Technical field

Embodiments of the present disclosure relate to communication networks, and particularly relate to methods, apparatus and computer program products for determining paths through a communication network, and allocating traffic flows to paths through a communication network.

Background

The problem of traffic routing in a network has been the subject of research for many years. Routing is performed for many types of networks, including circuit-switched networks and packet-switched networks. Conventional approaches to routing generally involve the assessment of multiple possible paths between two nodes of the network (e.g. an ingress or source node and an egress or destination node), and the selection of one particular path for the traffic based on some metric or cost criteria. For example, the shortest path may be selected, or the path leading to minimal cost.

However, this approach may not always lead to the optimal solution for the traffic, or for the network as a whole. For example, a demand for connectivity may specify one or more constraints to be satisfied by the network (e.g., a maximum latency). If a particular path satisfies the constraints, it may be beneficial for the network to allocate traffic to the particular path even if the path is not otherwise “optimal” in the sense of being shortest, minimal cost etc.

To distribute traffic in a better way, exploiting more network resources, avoiding congestion, and obtaining increased resilience, many solutions have been proposed. Traffic engineering is a class of solutions to the problem of path routing and traffic allocation, generally based on explicit path calculation and set up. These explicit paths may differ from the shortest path provided by most commonly used routing protocols.

One problem that is currently faced by engineers working in the field of network path calculation is that current routing protocols consider traffic flows moving in a single direction only, i.e. from a first network node to a second network node. However, the traffic flows in modern communication networks are often bi-directional, i.e., comprising traffic flowing from the first network node to the second network node, and also traffic flowing in the opposite direction, from the second network node to the first network node. Further, there is often a requirement that the traffic flowing in opposite directions in this manner experiences the same delay, either due to the nature of the application (e.g., voice or video calls, online gaming, etc), or to allow one-way delay calculation based on round-trip time measurements. To fulfill this requirement, the traffic flowing in either direction will be routed on the same path.

Conventional routing algorithms based on lowest-latency paths meet this requirement intrinsically, as the path having the lowest latency in one direction will inherently have the same lowest latency in the opposite direction. However, it is not clear how traffic engineering algorithms, which enable a wider range of paths to be generated for traffic through the network, should be applied or adapted to account for bi-directional traffic flows requiring identical latency.

It is an object of embodiments of the disclosure to address this and other problems.

Summary

A first aspect of the disclosure provides a method for configuring a communication network. The communication network comprises a plurality of nodes, with pairs of nodes being connected by respective links. Each link is associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow. The method comprises: receiving a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node; executing a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions; establishing the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and allocating the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

The disclosure also provides machine-readable mediums (e.g., computer programs, computer program products, etc) and apparatus for performing the methods described above and herein.

For example, one aspect provides a network management node for configuring a communication network. The communication network comprises a plurality of nodes, with pairs of nodes being connected by respective links, each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow. The network management node comprises processing circuitry and a non-transitory computer-readable medium storing instructions which, when executed by the processing circuitry, cause the network management node to: receive a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node; execute a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions; establish the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and allocate the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

A further aspect provides a computer program product comprising a non-transitory machine-readable medium storing instructions. A communication network comprises a plurality of nodes, with pairs of nodes being connected by respective links, each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow. When the instructions are executed by processing circuitry of a network management node, they cause the network management node to: receive a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node; execute a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions;

establish the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and allocate the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

The disclosure provides methods, apparatus and machine-readable mediums for calculating, establishing and allocating traffic to paths through a network. According to embodiments of the disclosure, a path-finding process seeks find paths by minimizing a sum of a cost metric for each link in a calculated path in both forward and backward directions. That is, when calculating a path for a demand belonging to a demand pair, specifying traffic flowing in opposite directions, the path-finding process utilizes the cost metric for each link in both directions when determining the path for just one of the demands in the demand pair. In this way, optimal paths may be chosen through the network for paired traffic demands and paired traffic flows belonging to those demands. Further, traffic is allocated to the paths in a manner which accounts for the need for paired traffic flows to have the same latency, flowing in opposite directions, despite the potential for multiple paths to be defined for each traffic demand.

<u>Brief description of the drawings</u>

For a better understanding of examples of the present disclosure, and to show more clearly how the examples may be carried into effect, reference will now be made, by way of example only, to the following drawings in which:

Figure 1 is a schematic diagram showing a LTE system;

Figure 2 is a schematic diagram of a traffic allocation system according to embodiments of the disclosure;

Figure 3 is a flowchart of a method according to embodiments of the disclosure;

Figure 4 is a schematic diagram showing application of the method according to embodiments of the disclosure to an example network;

Figure 5 is a flowchart of a method of calculating paths according to embodiments of the disclosure;

Figure 6 is a flowchart of a method of adjusting weights according to embodiments of the disclosure;

Figure 7 is a flowchart of a method of allocating traffic to paths according to embodiments of the disclosure; and

Figure 8 is a schematic diagram of a network management node according to embodiments of the disclosure.

Detailed description

Figure 1 shows a telecommunications network in which embodiments of the disclosure may be implemented according to one example.

The illustrated network is a Third Generation Partnership Project (3GPP) cellular network implementing the fourth generation (4G) or Long-Term Evolution (LTE) standard. The network comprises one or more wireless devices (also referred to as user equipments, UEs) connected to a radio access network via a radio interface. In the LTE system, the radio access network is called the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), and the radio interface is called LTE-Uu. The radio access network is connected to one or more core network nodes by respective interfaces, and in this way the UE is provided with service by the network as a whole. User data is transmitted from the radio access network to a serving gateway (SGW) in the core network, via an interface called S1-U. This interface is generally known as the backhaul network, i.e., the connection between the radio access network and the core network.

In LTE, user data is transmitted over the backhaul network in Internet Protocol (IP) data packets, using a tunnelling protocol known as general packet radio service (GPRS) tunnelling protocol (GTP), e.g., GTPv1-U protocol (see 3GPP, TS 29.281). The S1-U interface thus transports 4G data traffic flows, and typically these traffic data flows will be bi-directional; that is, many of the services accessed by the UE will involve the transmission of user data from the UE to the core network, and from the core network to the UE.

Thus, in some embodiments of the disclosure, the methods described herein are implemented to calculate paths and allocate traffic to paths in a backhaul network. The backhaul network may be implemented in a 4G network as shown in Figure 1, or in any other wireless network including Global System for Mobile Communications (GSM) including General Packet Radio Service (GPRS) and Enhanced Data Rates for GSM Evolution (EDGE), Universal Mobile Telecommunications Systems (UMTS) including High Speed Packet Access (HSPA), LTE Advanced and LTE Advanced Pro, and the fifth generation (5G) standard currently under development, New Radio (NR).

Further, those skilled in the art will appreciate that the methods described herein may be implemented in communication networks other than backhaul networks. Indeed, aspects of the disclosure may apply to any part of a telecommunications network, including fronthaul networks (i.e., the network connections within base stations having a distributed architecture).

Figure 2 is a schematic diagram of a traffic allocation system 200 according to embodiments of the disclosure. The system 200 comprises a client network 202 and a transport network 204 providing connectivity for the client network 202. In one particular embodiment, the client network 202 may comprise a radio access network such as that described above with respect to Figure 1 (E-UTRAN), while the transport network 204 may comprise the backhaul network (S1-U). However, as noted above, in other embodiments, the embodiments of the disclosure may calculate paths and allocate traffic for any telecommunications network.

The transport network 204 comprises a plurality of nodes coupled to each other via respective links. A node may be connected to any number of other nodes of the network 204. In one example, the transport network 204 may utilize optical communication techniques, with the nodes comprising optical switches or other hardware and the links comprising optical links such as optical fibres.

A management node 206 for the system 200 is communicatively coupled to both the client network 202 and the transport network 204. The management node 206 may be a software defined networking (SDN) controller or SDN function, for example. The management node 206 is operative to receive connectivity demands from the client network 202, and allocate traffic for those connectivity demands to one or more paths through the transport network 204. Further detail is provided below.

In step 210, the management node 206 obtains topology information for the transport network 204. For example, the topology information may comprise one or more of:

identities of a plurality of nodes in the network; an indication of the links between respective pairs of the plurality of nodes (e.g., an indication as to which node identities are connected to which other node identities); an indication of the penalty or cost associated with each link and/or node (e.g. latency); and an indication of the capacity of each link and/or node. The topology information may be visualized as a graph of nodes interconnected by links.

The management node 206 may further obtain, in step 210, an indication of the traffic usage over each link of the transport network 204. Note that the traffic usage over each link will be a sum of the traffic from all paths through the network 204 which utilize that link.

In step 208, the management node 206 obtains, from the client network 202, a plurality of demands for connectivity between nodes of the client network 202 or between attachment points of the transport network. Thus each connectivity demand may comprise identities of a first node (e.g. a source or ingress node) and a second node (e.g., a destination or egress node), with traffic to be routed between those nodes via the transport network 204.

Each connectivity demand may further specify one or more constraints to be satisfied by the routing through the transport network 204. For example, the connectivity demand may specify a maximum latency to be associated with the routing. In this case, the constraint is satisfied by a particular path through the network 204 if the total latency of the path is less than the maximum latency. In this context, the constraint may be specified by the user or client associated with the connectivity demand.

In other embodiments, each connectivity demand may alternatively or additionally be associated with a constraint which is specified by an operator of the transport network 204. For example, although optical fibres provide an excellent transmission medium with high bandwidth, optical signals will generally deteriorate as they cross optical links (e.g. fibres) and nodes (e.g. switches) owing to various effects such as group velocity dispersion, fibre loss, adjacent channel cross-talk, self phase modulation, etc. Eventually the optical signals may degrade to the point where they are unreadable. Where the transport network 204 comprises optical links and optical nodes, therefore, it may be beneficial to limit the light path length and convert the optical signals to electrical signals at certain points (before onward transmission as optical signals). A constraint may be associated with each connectivity demand defining a maximum light path length through the network before optical-to-electrical conversion.

Conversely, optical-to-electrical conversion is associated with increased cost in operating the transport network 204 (e.g. through the necessity of providing electro-optic converters, etc) and reduced capacity in the transport network (as the electro-optic converters may become bottlenecks for traffic passing through the network). A constraint may therefore be associated with each connectivity demand limiting the number of electro-optic converters in each path. For example, the constraint may define a maximum number of electro-optic converters per unit length within the path.

Further examples of possible constraints comprise a maximum number of crossed nodes within the path (e.g., to limit node resource usage or to limit the path length for path computation purposes), and a maximum number of segments used to express the path routing (e.g. after label stack optimization in a Segment Routing network). The latter constraint may be applicable particularly if the nodes or routers in the path have a limitation as to the label stack depth they are capable of processing.

Each connectivity demand may further be associated with or comprise an indication of the traffic usage for that connectivity demand (e.g., an amount of traffic which is to use the connectivity). For example, the traffic usage per demand may be measured by the client or the management node 202 where paths for the connectivity demand have already been established for a period of time (e.g. when the process described herein is used to update existing traffic allocations). The traffic usage may be an average of the traffic usage measured over a particular time window, for example. Thus, the management node may additionally receive in step 208 an indication of a measured amount of traffic flowing for each connectivity demand.

According to embodiments of the disclosure, at least one of the connectivity demands comprises a demand for bi-directional, co-routed paths through the transport network 204. That is, connectivity is required from a first network node to a second network node in one direction, and also in the reverse direction from the second network node to the first network node. Further, the traffic flowing in either direction is required to experience the same latency. Thus, the traffic flowing in either direction should be co-routed; that is, traffic flowing from the first network node to the second network should follow the same path through the transport network 204 as traffic flowing from the second network node to the first network node, but in the opposite direction.

In step 212, the management node computes one or more paths satisfying the constraints for each connectivity demand, and determines weights according to which traffic for each connectivity demand should be allocated to each of the computed paths. In particular, according to embodiments of the disclosure, the management node calculates paths and allocates traffic to those paths in a manner which accounts for the need for co-routing of bi-directional connectivity demands.

These paths and their respective weights may be provided to the transport network 204, and particularly to the first nodes (e.g., the source or ingress nodes), so that traffic for each demand can be allocated between one or more paths in accordance with the weights. Alternatively, particularly where the traffic comprises one or more indivisible traffic flows, the management node may itself allocate those flows to the one or more paths. In these embodiments, mapping between the traffic flows and the paths may be provided to the transport network 204 and/or the client network 202, such that the traffic flows can be directed on the allocated paths. Further detail regarding these and other aspects of the disclosure is provided below.

Figure 3 is a flowchart of a method according to embodiments of the disclosure. The method may be performed in a management node for a telecommunications network, such as the SDN controller 206 described above with respect to Figure 2.

In step 300, the management node obtains topology information for telecommunications network. This step may be largely similar to step 210 described above. Thus, in one embodiment, the topology information may comprise one or more of: identities of a plurality of nodes in the network; an indication of the links between respective pairs of the plurality of nodes (e.g., an indication as to which node identities are connected to which other node identities); an indication of the penalty or cost metric associated with each link and/or node (e.g. latency, etc); and an indication of the capacity of each link and/or node. The management node may obtain such information through communications with the network itself (e.g. via one or more control interfaces between the management node and the network), or through receipt of configuration information from a designer or operator of the network. The topology information may be updated periodically or whenever a change is made to the network topology.

In the following the nodes are indicated with indices. A link connecting node ***i*** with node ***j*** is indicated with $\boldsymbol{l_{ij}}$.

A link $\boldsymbol{l_{ij}}$ has capacity ***c(***$\boldsymbol{l_{ij}}$***)***, which can be expressed e.g. as its bandwidth in bits per second.

A link $\boldsymbol{l_{ij}}$ has cost metric ***penalty(***$\boldsymbol{l_{ij}}$***)***, which can be expressed in different units of measure (e.g., seconds or submultiples for time penalties such as delay or latency, or other operator-defined costs).

A node ***i*** may also have a cost metric ***penalty(i)*** in the same unity of measure as the link penalty. For example, packets crossing routers may experience some latency.

In this example, the penalty is assumed to be additive, that is to say, the resulting penalty of crossing one or more links and one or more nodes is the sum of their individual penalties. For example, penalties such as latency may be additive. Those skilled in the art will appreciate that different penalties may accumulate in different and/or non-linear ways. For example, optical impairments may accumulate in a non-linear fashion. The present disclosure is not limited in that respect.

The concepts of the link, capacity and penalty defined above are uni-directional, that is, considered in only a single direction. As noted above, embodiments of the present disclosure consider the impact of bi-directional connectivity demands on the network. Thus additional quantities are defined as follows:

A ***link pair*** is a pair of unidirectional links between the same nodes in opposite directions. The ***paired link*** $\boldsymbol{l_{ji}}$ of a link $\boldsymbol{l_{ij}}$ is the link of its pair in the opposite direction. It may be assumed that any two nodes in the network are either connected by a link pair or not connected, so that every link has a paired link to form a link pair.

The ***paired cost*** of a uni-directional link is the sum of its cost metric ***penalty(***$\boldsymbol{l_{ij}}$***)*** and the cost of its paired link ***penalty(***$\boldsymbol{l_{ji}}$***)***. Two links in the same link pair therefore have the same paired cost. Paired links may be assumed to have the same latency but may have different (unidirectional) cost and capacity.

In step 301, the management node obtains one or more demands for connectivity over the network. This step may be largely similar to step 208 described above. Thus the plurality of demands for connectivity may relate to connectivity between nodes of a client network or between attachment points of a transport network. Each connectivity demand may comprise or be associated with identities of a first node (e.g. a source or ingress node) and a second node (e.g., a destination or egress node), with traffic to be routed between those nodes via the transport network 204. Each connectivity demand may further be associated with one or more constraints to be satisfied by the routing through the telecommunication network (e.g., one or more of the constraints described above), and an indication of traffic usage of the connectivity.

The connectivity demand between nodes ***i*** and ***j*** is indicated with $\boldsymbol{d_{ij}}$. The connectivity demand $\boldsymbol{d_{ij}}$ may be associated with a measured usage $\boldsymbol{u(d_{ij})}$. The connectivity demand $\boldsymbol{d_{ij}}$ may further be associated with a maximum acceptable penalty $\boldsymbol{max\_penalty(d_{ij})}$ (i.e., a constraint). A path may not be selected for the demand if it exceeds that maximum acceptable penalty.

Connectivity usages need to be comparable among each other (e.g., use the same or similar units), and similarly link capacities need to be comparable among each other. However, it will be understood by those skilled in the art that connectivity usage and link capacity may not need to be directly comparable with each other, i.e. defined using the same or similar units.

The information for a connectivity demand may be sent by a client network to the management node, e.g., as in step 208 described above.

Connectivity demands may be unidirectional. A unidirectional demand is an ordinary connectivity requirement from one node or client point to another.

One or more of the demands may be part of a ***demand pair***. A demand pair is a pair of unidirectional demands (also referred to as ***demand pair components***) between the same nodes but in opposite directions. A demand pair should be satisfied by a path pair to allow the same differential delay in both directions.

Demands that are components of a demand pair are called ***paired demands***. Demands may be unidirectional or paired by request. Paired demands are assumed to have equal penalty (e.g., latency) requirements.

In step 302, the management node executes, for each connectivity demand, a path-finding process to generate a list of possible paths through the telecommunications network from the first node to the second node which satisfy the constraint or constraints (e.g. a maximum latency). In one embodiment, the management node calculates one or more non-looping paths which satisfy the constraint or constraints (such as the shortest path, or the k shortest paths where k is an integer equal to or greater than one). In one particular embodiment, the management node calculates all non-looping paths which satisfy the constraint or constraints

As noted above, at least one of the connectivity demands is a demand pair, comprising a first demand pair component between nodes of the network in one direction, and a second demand pair component between those nodes but in the opposite direction. According to embodiments of the disclosure, the path finding process for a demand pair component of the demand pair seeks or is configured to minimize a sum of the cost metric for each link in the calculated path in both directions. That is, the path-finding process finds paths that satisfy the constraint for the first demand pair component, but utilizes the paired cost when calculating whether or not that constraint is met.

For example, in one embodiment, the path-finding process finds the shortest path between nodes of the network (or the k shortest paths), utilizing the paired cost for each link in the path.

In this context, a path $\boldsymbol{p_k(d_{ij})}$ for demand $\boldsymbol{d_{ij}}$ is defined as the set of adjacent nodes and links it crosses. The penalty of a path is the sum of the cost metrics of all nodes and links it crosses. $\boldsymbol{P(d_{ij})}$ is the set of paths satisfying the requirements for $\boldsymbol{d_{ij}}$. Where a demand is a paired demand (i.e., is one component of a demand pair), the penalty of the path is the sum of the paired cost metrics of all nodes and links that it crosses.

One way of computing paths in a network is by using a shortest path (minimal cost) algorithm. This computes only one path per demand. However, this has the drawback of potentially leading to bottlenecks in case of unbalanced traffic on the demands and/or on the background. Again, where the demand forms part of a demand pair, the computation utilizes the paired cost of each link rather than the unidirectional cost of the link.

The equal cost multi-path (ECMP) concept is also already used in IP networks. All the paths with minimal cost are used, if there are ties in the minimal cost calculation. While providing some alternatives, the number of paths per demand is not explicitly controllable. This number does not depend on the node distance and may be too low or too high for the network operation needs.

The k-shortest paths is a well-known extension of the shortest path algorithm, which finds k paths between two nodes with non-increasing cost. Those skilled in the art will appreciate that several algorithms are known for computing the *k* shortest paths and the present disclosure is not limited in that respect. In one embodiment, the Eppstein algorithm may be used to discover the *k* shortest paths between a source node and a destination node satisfying a given constraint (e.g., as described in “Finding the *k* Shortest Paths”, by David Eppstein, SIAM Journal on Computing, Volume 28, pp 652-673).

A further method for calculating the paths based on paired cost is set out below with respect to Figure 5, in which a set of diverse paths are found.

Once the path or paths for one of the demand pair components in a demand pair is calculated, the path or paths for the other demand pair component in the demand pair are identical, but in the opposite direction. The paths for the demand pair are co-routed (i.e., identical) but in opposite directions.

The usage of the paired cost is motivated by the fact that the cost of a path pair is calculated as the sum of the costs of the demand pair components. **Figure 4** shows an example of one possible network, with four nodes labelled A to D. The costs for each link are indicated in the drawing, and it will be seen that the link cost in one direction is not always the same as the link cost between the same nodes in the opposite direction (although the latency will in general be the same). Node A has links with each of nodes B, C and D; node B has links with each of nodes A, C and D; nodes C and D each have links only with nodes A and B (and not with each other). Suppose there is a demand pair for connectivity between nodes A and B. If calculated separately, the shortest paths between A and B in the two directions would follow different routes. Traffic flowing from node A to node B would be routed via node C (at a cost of 1 + 1 = 2). Traffic flowing from node B to node A would be routed via node D (at a cost of 1 + 1 = 2). However, by using the paired costs instead of the unidirectional cost (e.g., AC cost would be 4 + 1 = 5) and selecting the lowest cost path, the direct path pair (AB) would be chosen (cost 8), and any other path pair would have a higher cost (e.g., ACB with cost 1 + 1 + 4 + 4 = 10). Thus, the shortest path may be determined while still satisfying the requirement of bidirectional co-routing.

The output of step 302 is a set of one or more paths for each connectivity demand, which satisfy the constraint(s) associated with that demand.

In step 304, the paths calculated in step 302 are established in the network (e.g., as described above with respect to step 212). For example, the paths may be provided to the nodes of the network to be included in routing tables implemented in those nodes. Thus the method may comprise transmitting an indication of the paths in the set of paths to a source node for each demand, for example by means of a configuration protocol (e.g. Network Configuration Protocol (NETCONF) or Path Computation Element Protocol (PCEP)) and from the source node to the nodes in the paths via a signaling protocol (e.g., Resource Reservation Protocol – Traffic Engineering (RSVP-TE)) or embedded in the data packet overhead (segment routing).

In step 306, the management node assigns weights, for each demand, for each of the paths in the set (see also step 212 described above). The weight assigned to path $\boldsymbol{p_k}$ is indicated by $\boldsymbol{w_k}$. The sum of all weights for a demand may be constrained as equal to one, such that the traffic for a given demand is distributed over the set of paths found for that demand. Where a single path is defined for a demand (e.g., the shortest path), the weight for that path is defined as 1.

Numerous techniques are known for assigning weights. One example is described in more detail in Figure 6, below. Further examples may be found in PCT application nos PCT/EP2017/084207 and PCT/EP2018/080040.

In step 308, traffic for each demand is allocated to the paths in accordance with the weights calculated in step 306. In the simplest case, where only a single path is defined for a demand, all traffic for that demand is routed via that path. Where more than one path is defined for a demand, the task of allocating traffic to the various paths is more complicated as the traffic is typically comprised of a plurality of indivisible flows, with each flow allocated to a single path. For example, each flow might relate to a particular service or application. Further, where that flow forms part of a paired demand, traffic for the other demand pair component (and the corresponding traffic flow) should follow the same path in the opposite direction. Further detail regarding this aspect is described below with respect to Figure 7.

The allocation of traffic flows to paths may be output by the management node to the transport network 204 and/or the client network 202 for implementation. For example, the management node may output a mapping between flow identifiers and corresponding paths, to the first node (e.g. the ingress node) or the client node associated with the demand.

Figure 5 is a flowchart of a method for calculating paths with diversity between a first (or source) node and a second (or destination) node, according to embodiments of the disclosure. The method assumes that a plurality of possible paths between the first node and the second node are available. The method is one example of a path-finding process, by which the paths for a demand may be calculated in step 302 above.

In step 501 the method comprises determining a first path from the plurality of possible paths, wherein the first path comprises a path with a smallest distance between the source node and the destination node in the plurality of paths. The “distance” here refers to the penalty or cost metric of each path, and comprises a sum of the cost metrics associated with each node and link in the path. Where a demand is a paired demand, the distance refers to the sum of the paired cost for each node and link in the path.

In step 502, the method comprises adding the first path to a set of paths. The set of paths may be defined as H, and may be updated as each path is selected. At this step the number of paths in the set, j, is set to 1.

In some examples, at step 503, the method checks whether enough paths have been found. For example, if k paths are needed, the method checks whether the number of paths in the set, j, is less than k. If enough paths have been found, the method passes to step 507 in which the method comprises configuring transmitting of the data from the source node to the destination node using paths in the set of paths. For example, where the method is implemented in step 302 of the method described above, the method proceeds to step 304 to establish the paths and to step 306 to determine weights for the set of paths.

If j is less than k the method passes to step 504. In step 504, the method comprises determining a next path based on a cost of the next path, wherein the cost of the next path comprises a first component indicating a distance between the source node and the destination node in the next path, and a second component representative of common links between nodes in the next path and links between nodes in paths already in the set of paths, for example representative of a number of common links between nodes in the next path and links between nodes in paths already in the set of paths. . The next path may comprise a $(j+1)^{th}$ path, where j is an integer value representing a number of paths in the set of paths on determination of the next path.

In some examples, the method may then pass to step 505 in which the method checks whether the selected next path is already in the set of paths. If the next path is already in the set of paths, the method passes to step 507. If the next path is not already in the set of paths, the method passes to step 506.

In other words, the method may comprise performing the step of determining k paths. However, in some examples, if the (k – n)th path determined is equal to a path in the set of paths, only the k – (n + 1) paths already found are added to the set of paths. However, if all k paths are distinct, the method comprises adding the k paths to the set of paths.

In step 506 the method comprises adding the next path to the set of paths. The number of paths in the set of paths may then be increased by 1, e.g j = j+1. The method may then return to step 503. In this way, the method repeats until a sufficient number of paths (i.e., k) is found.

For example, if the method is searching for k paths, and j of the paths have already been found (i.e. $j < k$). Path p_{j+1}must now be calculated for example, with the algorithm described below. The algorithm described here is a modified Dijkstra algorithm, however, it will be appreciated that a different shortest path algorithm may be modified (e.g., Bellman-Ford) in a similar manner.

In this example, the modified Dijkstra algorithm receives as input: the network graph; the first and second nodes; and the set H of the previously computed paths. It will be appreciated that when the algorithm first starts, the set H may comprise no paths. However, in some examples, paths may be already included in the set H, for example, if a particular path has been chosen as it passes via a node which also requires the transmitted data. The algorithm may then be performed to find each path to then add to the set of paths. In some examples therefore, the algorithm may be performed k times.

In some examples, for each node in the network graph the algorithm maintains the first component, herein noted as $c(p_{j+1})$, as the distance from the source node of the path p_{j+1} calculated so far and the predecessor of the node in p_{j+1}. In some examples, the first component $c(p_{j+1})$, is calculated as

$$c(p_{j+1}) = \sum_{l} \beta_l,$$

where l are links in the (j+1)th path and β_l is a link weighting factor associated with a link l. β_l may in some examples be set as higher for some links in the path, for example to avoid heavy traffic passing through that particular link.

In this example, the algorithm also maintains on each node, a vector of ***j*** numbers representing intersection costs $c_i(p_{j+1}, p_i)$ between the (j+1)th path, p_{j+1} and each of the previously found j paths, p_i in the set of paths H.

The value of the intersection cost path p_{j+1} and paths, p_i may be calculated as

$$c_i(p_{j+1}, p_i) = \sum_m \beta_m,$$

where m are links between nodes in the (j+1)th path p_{j+1} which are also in the one of the paths in the set of paths p_i and β_m is a link weighting factor associated with a link m.

In other words, for each link m in the path p_{j+1}, the weighting of the link β_m is added to $c_i(p_{j+1}, p_i)$ if the link m is in p_i, otherwise $c_i(p_{j+1}, p_i)$ remains unchanged.

A similarity between a path p_{j+1} and one previously computed path p_i may be defined as the ratio between the cost of the intersection between the path $p_{j+1}$and the path p_i and the first component of path p_i**:**

$$Sim(p_{j+1}, p_i) = \frac{c_i(p_{j+1}, p_i)}{c(p_i)}. \quad (1)$$

However, it will be appreciated that there may be several ways to define the similarity between two paths. A straightforward one would be cost of the intersection divided by cost of union. However in the definition of equation (1) the cost of union is replaced with the first component of only the already computed path p_i such that the first component of the path p_{j+1} is left out of the calculation. This avoids encouraging the selection of $p_{j+1}$from utilizing more network resources.

A similarity between the path p_{j+1} and the set H of paths that has been previously computed may be defined as the Euclidean norm of the similarity between p_{j+1} and all the paths in H:

$$Sim(p_{j+1}, H) = \sqrt{\sum_i (Sim(p_{j+1}, p_i))^2}\ , p_i \in H\ . \quad (2)$$

In some examples, the second component comprises the similarity between the path p_{j+1} and the set H of paths that has been previously computed.

In some examples, step 504 of Figure 5 therefore comprises calculating the first component as:

$$c(p_{j+1}) = \sum_l \beta_l,$$

as described above, and calculating the second component as:

$$\lambda Sim(p_{j+1}, H).$$

The cost of the path p_{j+1}may therefore be calculated as:

$$c(p_{j+1}) + \lambda Sim(p_{j+1}, H)\,,$$

where λ is a weighting factor.

In some examples, the weighting parameter λ may be set based on a relative importance of network resources and diversity between paths in the telecommunications network. In other words, in some examples, it may be desirable for selected paths to be as diverse as possible, at the expense of using extra network resources. In these examples, the value of λ may be set as a higher value, for example 50 to 100. However, in circumstances where it is less important for the selected paths to be diverse, the value of λ may be set as a lower value for example 1 to 5. It will be appreciated that any suitable value of λ may be used.

In some examples therefore the step 504 comprises determining the $(j+1)^{th}$ path such that the cost of the $(j+1)^{th}$ path is lower than costs of all other paths in the plurality of possible paths.

It will be noted that the method show in Figure 5 is one example of a method of finding suitable paths with diversity. Other methods are also possible, such as the breadth-first and depth-first searches mentioned above, Dijkstra algorithm, Bellman–Ford algorithm, Yen's algorithm (which calculates k shortest paths in non-decreasing cost order), Eppstein's algorithm, and Suurballe's algorithm. In each of these methods, where the demand belongs to a demand pair, the paired cost of each link and node is used for the calculation of path cost, rather than the unidirectional cost metric.

Figure 6 is a flowchart illustrating in more detail a method of adjusting the weights for distributing traffic over a plurality of paths of a network according to embodiments of the disclosure. The method may be employed by the management node in step 306, described above with respect to Figure 3.

The method begins in step 600, in which the management node measures the traffic on each of the links in the network. The traffic in each link may be reported directly to the management node by nodes of the network. Alternatively, the traffic in each link may be calculated based on the measured traffic associated with each demand (e.g., as reported by client nodes associated with the demand), and a set of initial weights assigned to each of the paths for each demand. The set of initial weights may be determined using any suitable method, including simply setting the weights at identical, nominal values. The traffic flowing in each link is the aggregate of the traffic flowing in each of the paths that traverse that link, and thus may comprise traffic from multiple demands. The measured traffic may be an average value, calculated over a defined time window.

As noted above, the weight assigned to path $\boldsymbol{p_k}$ is indicated by $\boldsymbol{w_k}$. A connectivity demand $\boldsymbol{d_{ij}}$ sends a measured traffic $\boldsymbol{u(d_{ij})}$. As a consequence, a path $\boldsymbol{p_k}$ satisfying demand $\boldsymbol{d_{ij}}$ carries traffic $\boldsymbol{u(p_k) = w_k \cdot u(d_{ij})}$.

The traffic on a link l_{ij} can be calculated as the sum of the usages of all paths crossing it plus the background traffic, $u(l_{ij}) = \sum_{l_{ij} \in p_k} u(p_k) + bt_{ij}$. This traffic is also directly measured as stated above.

The utilization or load of a link $v(l_{ij})$ is a ratio between the traffic $u(l_{ij})$ flowing on that link and the capacity $c(l_{ij})$ of the link. As such, it is the amount of traffic on the link divided by the capacity of the link. Thus a link which is operating at full capacity (i.e. the amount of traffic flowing on the link is equal to the capacity of the link) has a utilization of 1 or unity. According to embodiments of the disclosure, the utilization can take values which are greater than 1, which is representative of a link that is overloaded, i.e. the traffic exceeds a full capacity of the link. Traffic which exceeds the capacity of the link may be held in a queue prior to transmission over the link, e.g. in a buffer.

To calculate the optimal weights, we first of all define the problem in such a way that it is well understood what is the desired status of the network that the calculation aims to achieve.

Embodiments of the disclosure define a criticality of each link and use the criticality, instead of the utilization $v(l_{ij})$ of each link, to determine and adjust or determine the weight assigned to each path flowing through the network. Thus, in step 602, the management node calculates the criticality of each link l_{ij} as a function of the utilization $v(l_{ij})$ of the link. Traffic routing is carried out on the basis of the calculated weight of each path of the network.

The criticality is a measure of the status of the link in dependence on the traffic flowing through that link. As said above, there is traffic crossing a link. While the traffic remains below the link capacity, the status of the link can be considered to be good. In some aspects, if the link has a low load, e.g. the link is at half capacity, the link can be considered as having a status which is slightly worse than if the link is completely empty, because there is a lower opportunity for future traffic accommodation.
When the traffic load approaches the link capacity, however, a negative impact on the ability of the link to carry the traffic is determined, and reflected in the determined criticality value. For example, as the traffic load approaches the link capacity, the queues begin to fill and/or initial congestion is experienced. Therefore, in a region close to the capacity of the link (e.g., between 0.8 and 1, or 80% and 100% utilization), the criticality

is determined to rise sharply. Thus, the criticality reflects the impact of the traffic on a link/node, and not merely the amount of the traffic on the link/node.

In principle, the capacity of the link should not be exceeded, but the presence of queues (or buffers) allows some extra traffic to be absorbed for limited periods of time. Therefore, the criticality may also be defined for values of the utilization which are greater than unity, i.e. above nominal link/node capacity or over full capacity. In some aspects, the criticality has a minimum value at a first value of the utilization (e.g., a low value of utilization) and a maximum value at a second value of the utilization (e.g., a high value of utilization, greater than the first value). The second value of the utilization corresponds to a utilization which is over a full capacity of the link. As such, the criticality does not have a maximum value at a nominal maximum capacity of the link, but instead the criticality has a maximum value at a utilization which is above a nominal maximum (i.e. full) capacity of the link.

In one embodiment, the criticality may be mathematically defined using an S-shaped or sigmoid function. The sigmoid function as a shape is well studied and those skilled in the art will appreciate that there are several functions with a sigmoid shape. The logistic function is one of the most common, defined as follows:

$$s = \frac{1}{1 + exp\ (b \cdot (a - v))}$$

In this example, the criticality function *s* has two configurable parameters: the slope ***b*** and the intercept ***a***. ***v*** is the link utilization, defined above. The slope determines how steep the curve is in proximity of the intercept, and the intercept determines where the gradient of the function is highest (e.g., the value of the utilization at which the criticality is equal to 0.5 in this specific example). The presence of one or more configurable parameters in the criticality function means that the determined criticality value can be tuned or optimized for a particular network or network type. Thus, for example, in an initial configuration phase, one or more configurable parameters can be determined which provide for the determined criticality values to accurately reflect the impact of traffic on the operation of the network.

Those skilled in the art will appreciate that the criticality may not be defined as a logistic function. For example, alternative sigmoid functions may include $s = \frac{v}{\sqrt{(1+v^2)}}$.

The criticality may also not be defined as a conventional S-shaped or sigmoid function. For example, the example functions given above and with respect to Figure 5 are all smoothly varying and differentiable. The criticality function may be defined differently, for example as a series of straight line curves for different regions of the utilization. In another example, values of the criticality may be defined in a look-up table, based on corresponding values of the utilization. The values of the criticality may be defined based on the utilization so as to approximate the functions described above.

It will be further noted that if the same function were used regardless of link capacity, links with smaller capacity would not be discouraged over links with larger capacity in cases of equal utilization. However, the smaller link has less spare capacity than the larger link. In order to counteract this effect, the criticality may be normalized based on the capacity of the link.

One aim of the method of Figure 6 may be to minimize the sum of the link criticalities in the network.

In step 604, the management node determines the path criticality for each path, for each demand. The path criticality may be defined as a sum of the criticalities of each link which the path crosses. Thus the output of step 604 is a set of path criticalities for each demand, the set of path criticalities comprising a path criticality for each path defined for the demand.

In step 606, the management node determines the improvability of each demand. The improvability relates to the amount by which the path criticalities of the paths for each demand can be improved or equalized, so that no path is more critical than another path for a given demand. In one embodiment, the demand improvability for a particular demand is defined as the difference in criticality between the path for the demand having non-zero weight and maximum criticality, and the path for the demand having minimum criticality. Thus the output of step 606 is an improvability value for each demand.

The method then proceeds to evaluate stopping criteria and performance improvements, which are relevant only when the method is performed iteratively. For the purposes of this description, we assume that the stopping criteria are not met for a first iteration of the method, and thus the description of the method now moves to step 616, in which the demand with the maximum improvability is selected. In step 618, the weights for the paths of that selected demand are adjusted so as to decrease the improvability for that demand. One method of achieving that improvement is to reduce the weight for the path having the highest criticality (and non-zero weight) by a certain amount or step size, and to increase the weight for the path having the lowest criticality by the same amount or step size. The amount by which the weights are adjusted may vary as a function of the traffic for that demand. For example, the amount varies inversely proportionally to the traffic for the demand. The amount may be defined as:

$$\boldsymbol{min(w(cp(d_{max})), sensitivity * impr / u(d_{max}))}$$

So the amount is the minimum of the weight for the most critical path (**cp($\mathbf{d_{max}}$)**), and the improvability for the demand divided by the traffic for the demand, multiplied by a constant ***sensitivity***. The minimum function is present so that the weight is not reduced below zero. $\boldsymbol{d_{max}}$ is the demand having highest improvability.

As the improvability is defined as the difference in criticality between these two paths, the improvability of the selected demand should decrease.

In the illustrated embodiment, the method is iterative, and thus the method moves from step 618 back to step 600, in which the traffic flowing in each link is measured again. In iterations which are subsequent to the first iteration (as here), this step may involve the calculation of the traffic flowing in each link based on the previously measured values and the changes to the weights made in step 618, rather than re-measurement of the traffic flowing in all links of the network.

In steps 602, 604 and 606, the criticality of the links of the network, the path criticalities, and the improvabilities of the demands are re-calculated.

In step 608 (description of which was omitted above for the first iteration), the management node determines whether a stopping criterion has been reached. For example, one stopping criterion may be a maximum number of iterations. In another embodiment, the stopping criterion may comprise a determination that the highest improvability for all demands in the network is below a threshold In the event that the stopping criterion is reached, the method ends in step 610. The weights are output from the management node to the network, such that traffic for each demand can be routed through the network in accordance with the weights.

If the stopping criterion has not been reached, the method proceeds to step 612, in which the management node determines whether the adjustment to the weights made in step 618 resulted in an increase to the highest value of improvability of the demands in the network. If the maximum improvability has increased, the method proceeds to step 614, in which the step size (the amount by which the weights are adjusted in step 618) is attenuated. For example, the constant ***sensitivity***, defined above, may be attenuated. The attenuation may be implemented by multiplying the constant by an attenuation factor, which is a positive number less than 1 (such that the step size cannot be negative). The attenuated step size is then used in step 618 to adjust the weights of the demand having highest improvability. If the maximum improvability has not increased (as determined in step 612), the step size is not attenuated, and the method proceeds directly to steps 616 and 618.

It will be noted that, in embodiments where the minimum value of the criticality is non-zero but the gradient of the criticality is zero (i.e. flat) as the utilization approaches zero, when the load in the network is low the path criticality for each path will correspond to the non-zero value multiplied by the number of links that the path traverses. Thus longer paths will have higher criticality than shorter paths. In such embodiments, the method may alter the weights so as to increase the weights for shorter paths and lower the weights for longer paths. At low network load, this approaches a shortest path allocation.

In embodiments where the criticality varies with a non-zero gradient as the utilization approaches zero, even when the load in the network is low, the criticality for each path is dependent on the utilization of each link that the path traverses. In this embodiment, the weights for shorter paths may again be increased while the weights for longer paths may be decreased; however, a greater proportion of traffic may be allocated to longer paths than in the case where the gradient of the criticality is zero as the utilization approaches zero. The value of the gradient of the criticality as the utilization approaches zero may be configured or tuned so as to achieve a desired performance at low network load.

It will be noted that the method show in Figure 6 is one example of a method of finding weights for the various paths. Other methods are also possible, such as those described in PCT application nos PCT/EP2017/084207 and PCT/EP2018/080040.

Figure 7 is a flowchart of a method of allocating traffic to paths according to embodiments of the disclosure. The method may be performed as part of step 308 described above. The method is particularly applicable to the allocation of traffic for a demand pair to one or more paths through the network.

The method shown in Figure 7 begins with the assumption that one or more paths have been calculated and established for each demand in the demand pair, e.g., as described above with respect to steps 302 and 304, and also Figure 5. The paths for one demand of the pair are equal and opposite to the paths for the other demand of the pair. Further, in some embodiments, particularly where more than one path is defined for each path, weights have been determined for the allocation of traffic amongst those multiple paths.

The traffic for each demand may be comprised of a plurality of indivisible flows, such that each flow should be allocated to a single path. For example, each flow might relate to a particular service or application. Further, where that flow forms part of a paired demand, traffic for the other demand pair component (and the corresponding traffic flow) should follow the same path in the opposite direction. A flow pair is thus defined as a pair of unidirectional flows in opposite directions, belonging to the respective demands of a demand pair (defined above). Two components of the same flow pair should be allocated on the component of the same path pair to guarantee the same differential delay.

The method further begins with the assumption that no traffic flows for the demand have been allocated to the paths. In step 700, a traffic flow f of the demand is selected. In one embodiment, the largest traffic flow is selected in step 700 for allocation to a path; this embodiment is motivated by the insight that allocating a large traffic flow is more difficult than allocating a small traffic flow. In other embodiments, the traffic flow belonging to a flow pair with the highest combined traffic (i.e., flowing in both directions) is selected in step 700.

In step 702, the selected traffic flow is mapped to the path (e.g., of those defined in step 302) having the most unused capacity. Here the unused capacity is defined as the difference between the desired usage of the path (e.g., the bandwidth of the demand multiplied by the weight for that path), and the actual usage of the path (e.g., the bandwidth from the demand, currently allocated to the path). In the first iteration of the method shown in Figure 7, the actual usage of the path will be zero. Note that, as the method progresses, the unused capacity may become negative as the actual usage exceeds the desired usage of a path.

In step 704, the other traffic flow belonging to the same traffic flow pair as the flow selected and allocated in steps 700 and 702 respectively, is automatically allocated to the co-routed path in the opposite direction. In this way, the traffic for each traffic flow pair is allocated to co-routed paths in opposite directions, ensuring compliance with the need for identical latency within a traffic flow pair.

In step 706, the usage of the paths is updated to reflect the allocated traffic flow pair, and the method flows back to step 700 for selection and allocation of another traffic flow.

Each iteration of the method therefore allocates a flow pair to a path pair, one component per direction. This strategy is simple and fast.

The disclosure also provides machine-readable mediums (e.g., computer programs, computer program products, etc) and apparatus for performing the methods described above.

Figure 8 is a schematic diagram of a network management node 800 according to embodiments of the disclosure.

The network management node 800 may be configured to implement the method described above with respect to Figure 3, and possibly also those methods described with respect to Figures 5, 6 and/or 7, for example. The network management node 800 may be operable as the management node 206 shown in Figure 2, for example.

The network management node may be suitable for configuring a communication network comprising a plurality of nodes. Pairs of nodes are connected by respective links, with each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow.

The network management node 800 comprises processing circuitry (such as one or more processors) 802 and a non-transitory machine-readable medium 804 (such as memory). The memory may store instructions which, when executed by the processing circuitry 802, cause the network management node to: receive a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node; execute a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions; establish the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and allocate the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

The network management node 800 may further comprise one or more communications interfaces 806, for transmitting signals to and/or receiving signals from other network nodes of the wireless communications network. The interfaces 806 may comprise circuitry for the transmission and/or reception of electrical, optical or wireless signals.

The interface(s) 806, processing circuitry 802 and machine-readable medium 804 may be connected together in any suitable manner. In the illustrated embodiment, the components are coupled together directly, in series. In alternative embodiments, the components may be coupled to each other via a system bus or other communication line.

Thus the disclosure provides methods, apparatus and machine-readable mediums for calculating, establishing and allocating traffic to paths through a network. According to embodiments of the disclosure, a path-finding process seeks find paths by minimizing a sum of a cost metric for each link in a calculated path in both forward and backward directions. That is, when calculating a path for a demand belonging to a demand pair, specifying traffic flowing in opposite directions, the path-finding process utilizes the cost metric for each link in both directions when determining the path for just one of the demands in the demand pair. In this way, optimal paths may be chosen through the network for paired traffic demands and paired traffic flows belonging to those demands.

Further, traffic is allocated to the paths in a manner which accounts for the need for paired traffic flows to have the same latency, flowing in opposite directions, despite the potential for multiple paths to be defined for each traffic demand.

It should be understood—especially by those having ordinary skill in the art with the benefit of this disclosure—that that the various operations described herein, particularly in connection with the figures, may be implemented by other circuitry or other hardware components. The order in which each operation of a given method is performed may be changed, and various elements of the systems illustrated herein may be added, reordered, combined, omitted, modified, etc. It is intended that this disclosure embrace all such modifications and changes and, accordingly, the above description should be regarded in an illustrative rather than a restrictive sense.

It should be noted that the above-mentioned embodiments illustrate rather than limit the concepts disclosed herein, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended following statements and claims. The word “comprising” does not exclude the presence of elements or steps other than those listed in a statement or claim, “a” or “an” does not exclude a plurality, and a single processor or other unit may fulfil the functions of several units recited in the statements. Any reference signs in the statements and claims shall not be construed so as to limit their scope.

Similarly, although this disclosure makes reference to specific embodiments, certain modifications and changes can be made to those embodiments without departing from the scope and coverage of this disclosure. Moreover, any benefits, advantages, or solutions to problems that are described herein with regard to specific embodiments are not intended to be construed as a critical, required, or essential feature or element.

Further embodiments likewise, with the benefit of this disclosure, will be apparent to those having ordinary skill in the art, and such embodiments should be deemed as being encompassed herein.

CLAIMS

1. A method for configuring a communication network (200), the communication network comprising a plurality of nodes, with pairs of nodes being connected by respective links, each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow, the method comprising:

   receiving (300) a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node;

   executing (302) a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions;

   establishing (304) the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and

   allocating (308) the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

2. The method according to claim 1, wherein the path-finding process comprises determining (501) at least a first calculated path from the first node to the second node having the smallest sum of the cost metric for each link in the first calculated path in both the first and second directions.

3. The method according to claim 2, wherein the path-finding process comprises determining (504) one or more second calculated paths from the first node to the second node having next smallest sums of the cost metric for each link in the second calculated paths in both the first and second directions.

4. The method according to claim 2, wherein the path-finding process comprises determining (504) one or more second calculated paths from the first node to the second node, wherein a cost of each second calculated path comprises a first component corresponding to the sum of the cost metric for each link in the second calculated path in both the first and second directions, and a second component

representative of common links between nodes in the second calculated path and links between nodes in the first calculated path, and wherein the second calculated path is determined responsive to the cost of the second calculated path being a lowest cost compared to costs of other paths in a plurality of possible paths.

5. The method according to any one of the preceding claims, wherein traffic for the demand comprises a plurality of traffic flow pairs, each traffic flow pair comprising a first traffic flow belonging to the first traffic and a second traffic flow belonging to the second traffic, wherein allocating the first traffic and the second traffic comprises iteratively performing the following steps:
   a. allocating (702) a traffic flow for the demand to a calculated or co-routed path based on an amount of unused capacity in the calculated or co-routed paths; and
   b. automatically allocating (704) the other traffic flow in the traffic flow pair to a corresponding path in the opposite direction.

6. The method according to claim 5, wherein the traffic flow allocated in step a. is the largest unallocated traffic flow.

7. The method according to claim 5, wherein the traffic flow allocated in step a. belongs to the largest unallocated traffic flow pair.

8. The method according to any one of claims 5 to 7, wherein the traffic flow in step a. is allocated to the calculated or co-routed path having the greatest amount of unused capacity.

9. The method according to any one of claims 5 to 8, further comprising executing (306) an optimization process to determine respective first weights for each of the one or more calculated paths and respective second weights for each of the one or more co-routed paths, and wherein the unused capacity of a path is equal to a difference between an amount of traffic allocated to the path and a desired amount of traffic allocated to the path based on the first or second weights.

10. The method according to any one of the preceding claims, wherein the cost metrics associated with a link are defined by an operator of the communication network.

11. The method according to any one of the preceding claims, wherein each path has the same latency as its respective co-routed path.

12. The method according to any one of the preceding claims, wherein the communication network comprises a backhaul network.

13. The method according to any one of the preceding claims, wherein the method is performed by a software-defined network controller (206) for the communication network.

14. A network management node (206, 800) for configuring a communication network, the communication network comprising a plurality of nodes, with pairs of nodes being connected by respective links, each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow, the network management node comprising processing circuitry (802) and a non-transitory computer-readable medium (804) storing instructions which, when executed by the processing circuitry, cause the network management node to:

receive (300) a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node;

execute (302) a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions;

establish (304) the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and

allocate (308) the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

15. The network management node according to claim 14, wherein the path-finding process comprises determining at least a first calculated path from the first node to the second node having the smallest sum of the cost metric for each link in the first calculated path in both the first and second directions.

16. The network management node according to claim 15, wherein the path-finding process comprises determining one or more second calculated paths from the first node to the second node having next smallest sums of the cost metric for each link in the second calculated paths in both the first and second directions.

17. The network management node according to claim 15, wherein the path-finding process comprises determining one or more second calculated paths from the first node to the second node, wherein a cost of each second calculated path comprises a first component corresponding to the sum of the cost metric for each link in the second calculated path in both the first and second directions, and a second component representative of common links between nodes in the second calculated path and links between nodes in the first calculated path, and wherein the second calculated path is determined responsive to the cost of the second calculated path being a lowest cost compared to costs of other paths in a plurality of possible paths.

18. The network management node according to any one of claims 14 to 17, wherein traffic for the demand comprises a plurality of traffic flow pairs, each traffic flow pair comprising a first traffic flow belonging to the first traffic and a second traffic flow belonging to the second traffic, wherein allocating the first traffic and the second traffic comprises iteratively performing the following steps:
    a. allocating (702) a traffic flow for the demand to a calculated or co-routed path based on an amount of unused capacity in the calculated or co-routed paths; and
    b. automatically allocating (704) the other traffic flow in the traffic flow pair to a corresponding path in the opposite direction.

19. The network management node according to claim 18, wherein the traffic flow allocated in step a. is the largest unallocated traffic flow.

20. The network management node according to claim 18, wherein the traffic flow allocated in step a. belongs to the largest unallocated traffic flow pair.

21. The network management node according to any one of claims 18 to 20, wherein the traffic flow in step a. is allocated to the calculated or co-routed path having the greatest amount of unused capacity.

22. The network management node according to any one of claims 18 to 21, further comprising executing an optimization process to determine respective first weights for each of the one or more calculated paths and respective second weights for each of the one or more co-routed paths, and wherein the unused capacity of a path is equal to a difference between an amount of traffic allocated to the path and a desired amount of traffic allocated to the path based on the first or second weights.

23. The network management node according to any one of claims 14 to 22, wherein the cost metrics associated with a link are defined by an operator of the communication network.

24. The network management node according to any one of claims 14 to 23, wherein each path has the same latency as its respective co-routed path.

25. The network management node according to any one of claims 14 to 24, wherein the communication network comprises a backhaul network.

26. The network management node according to any one of claims 14 to 25, wherein the network management node is a software-defined network controller for the communication network.

27. A computer program product comprising a non-transitory machine-readable medium (804) storing instructions which, when executed by processing circuitry of a network management node, the network management node for configuring a communication network comprising a plurality of nodes, with pairs of nodes being connected by respective links, each link being associated with a cost metric having a first value in a first direction of traffic flow and a second value in a second, opposite direction of traffic flow, cause the network management node to:

receive (300) a demand for connectivity between a first node and a second node of the plurality of nodes, the demand specifying first traffic for the demand flowing in a direction from the first node to the second node, and second traffic for the demand flowing in an opposite direction from the second node to the first node;

execute (302) a path-finding process to determine one or more calculated paths from the first node to the second node, wherein the path-finding process seeks to minimize a sum of the cost metric for each link in the calculated path in both the first and second directions;

establish (304) the one or more calculated paths in the communication network for the first traffic, and establishing respective co-routed paths in the opposite direction for the second traffic; and

allocate (308) the first traffic for the demand to the one or more calculated paths and the second traffic for the demand to the one or more co-routed paths.

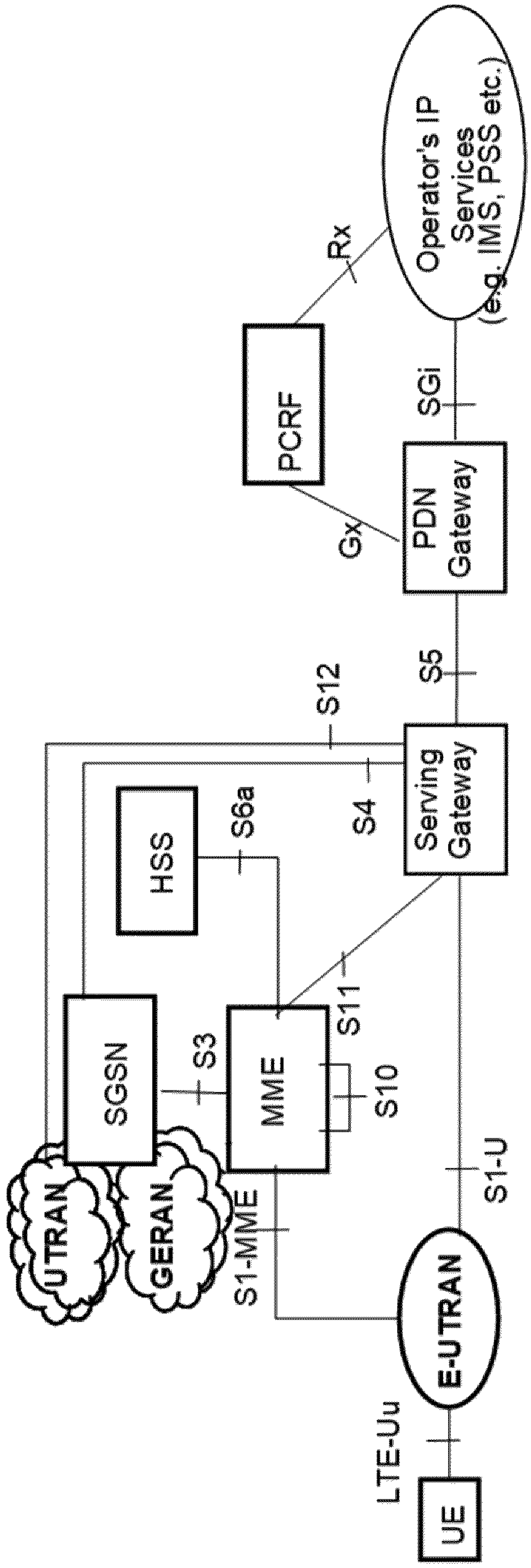
UE
LTE-Uu
E-UTRAN
S1-MME
S1-U
UTRAN
GERAN
SGSN
MME
S3
S10
S11
HSS
S6a
S4
S12
Serving Gateway
S5
PDN Gateway
Gx
PCRF
SGi
Rx
Operator's IP Services (e.g. IMS, PSS etc.)

Fig. 1

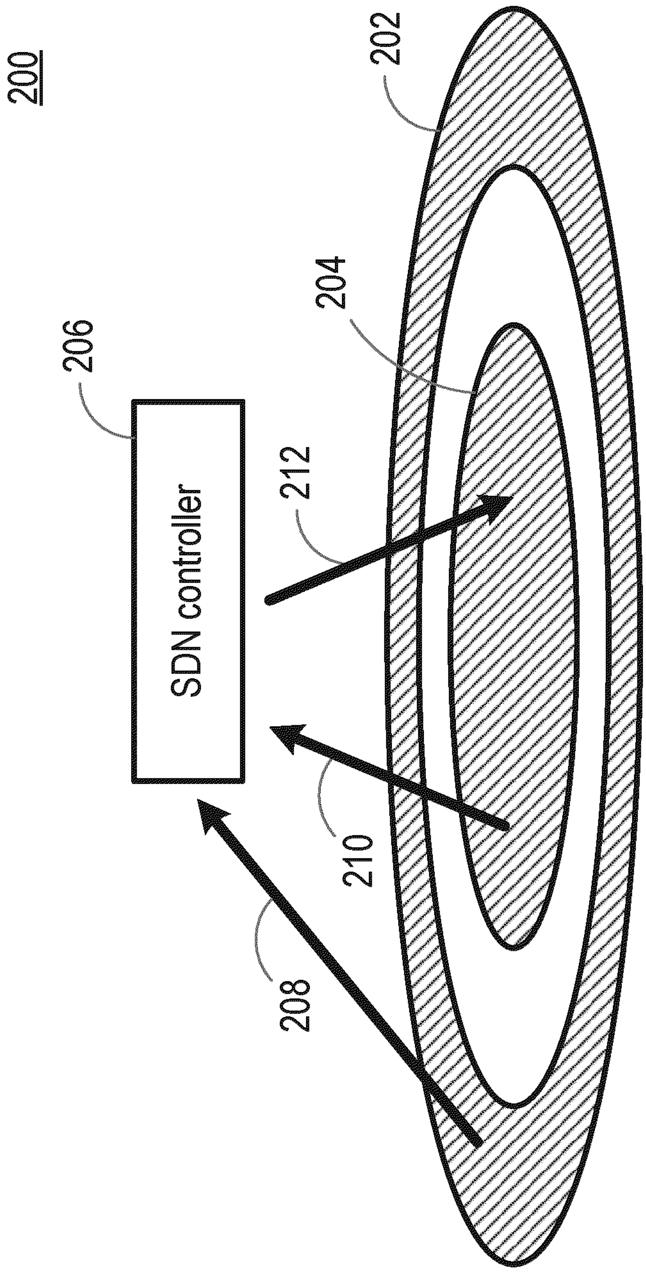

Fig. 2

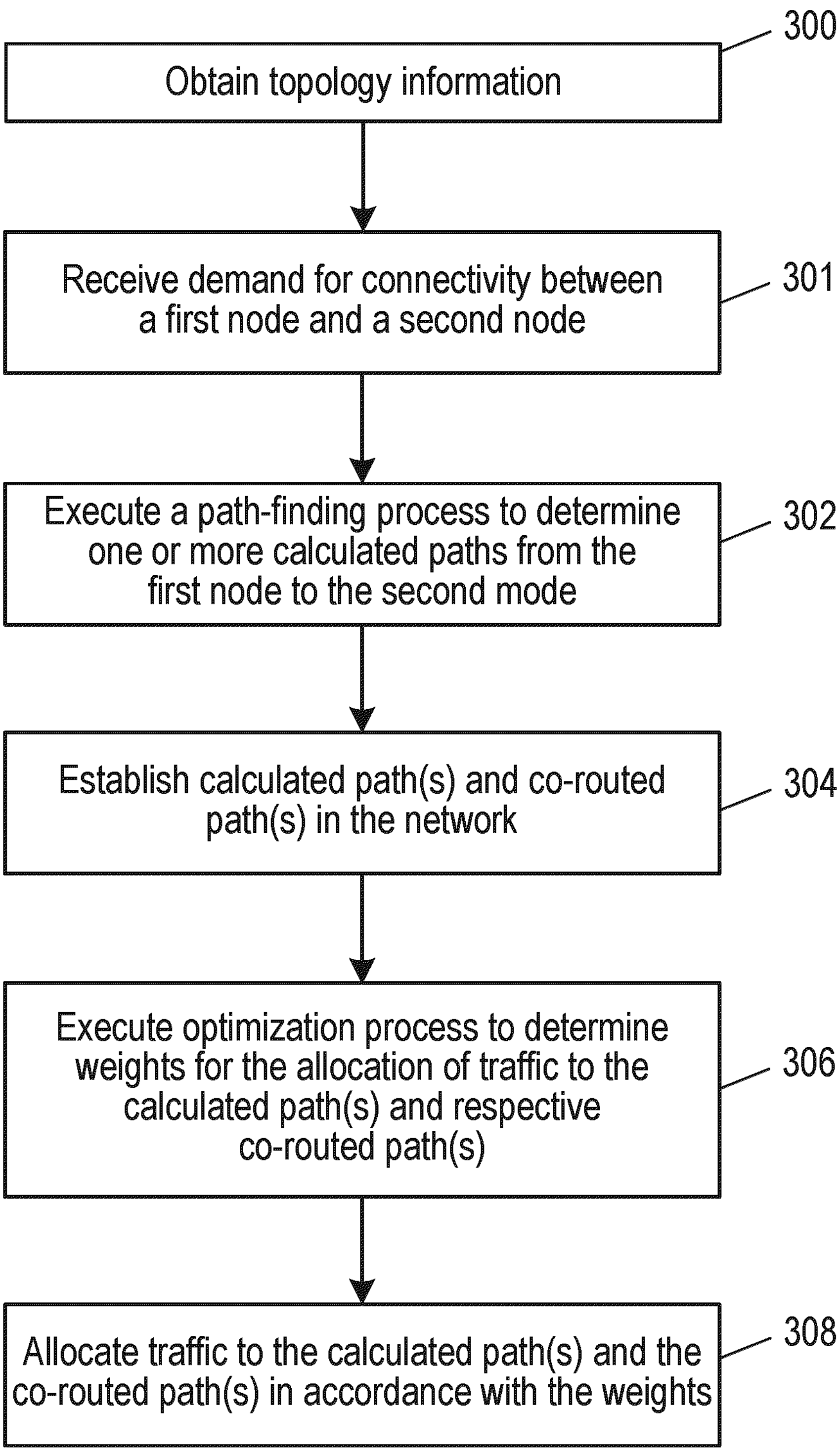
300
Obtain topology information
301
Receive demand for connectivity between a first node and a second node
302
Execute a path-finding process to determine one or more calculated paths from the first node to the second mode
304
Establish calculated path(s) and co-routed path(s) in the network
306
Execute optimization process to determine weights for the allocation of traffic to the calculated path(s) and respective co-routed path(s)
308
Allocate traffic to the calculated path(s) and the co-routed path(s) in accordance with the weights

Fig. 3

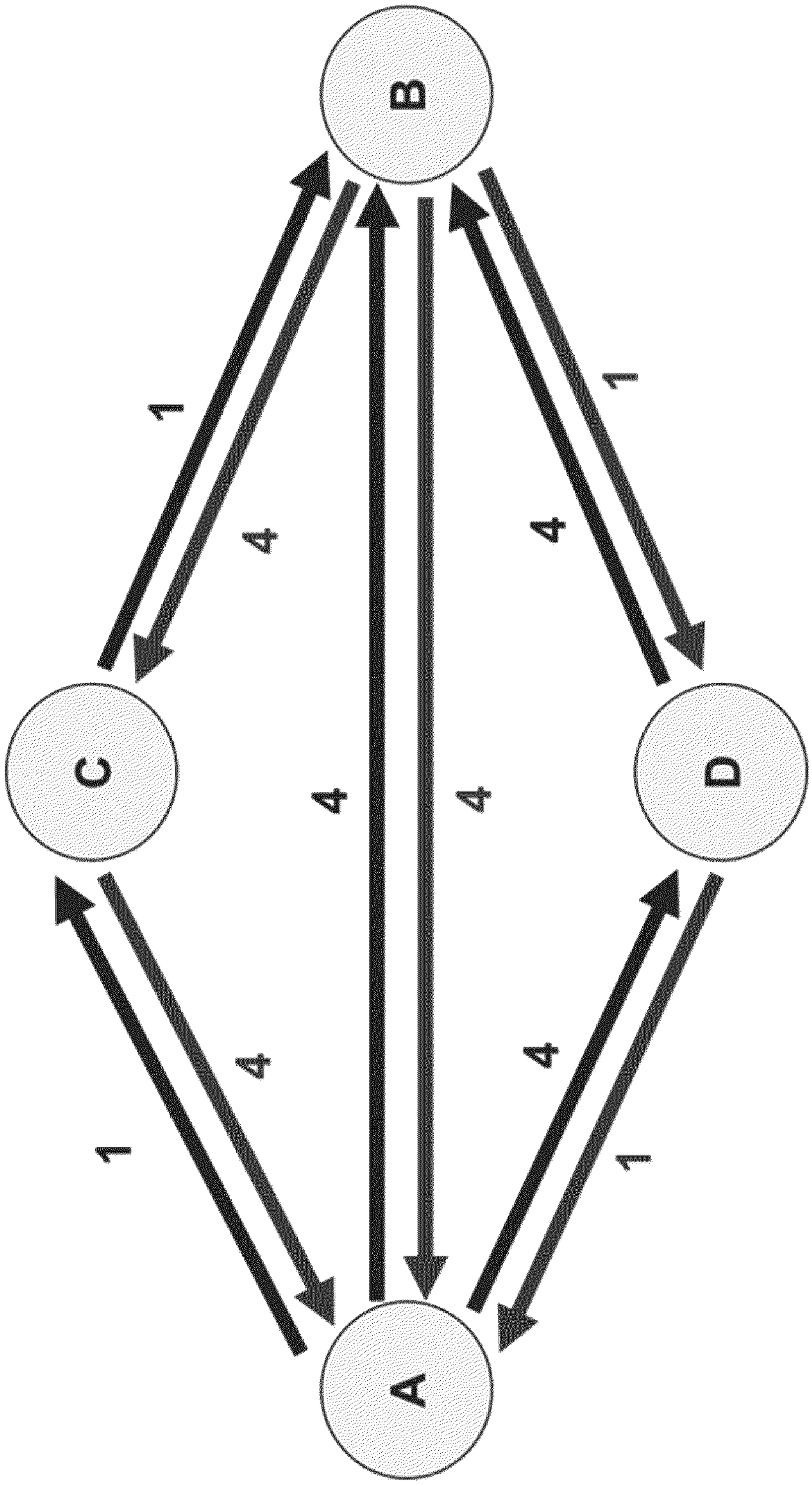

Fig. 4

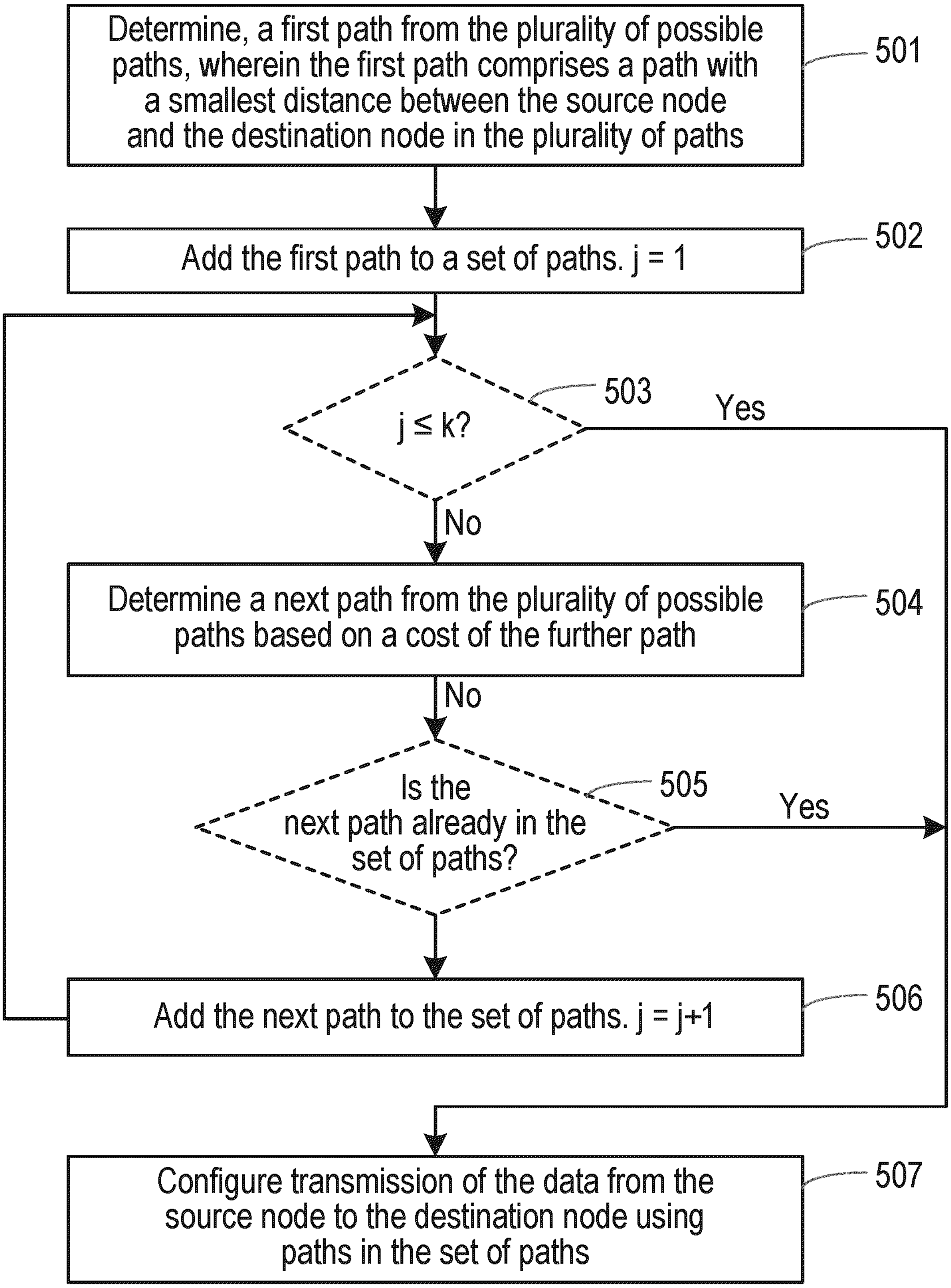
Determine, a first path from the plurality of possible paths, wherein the first path comprises a path with a smallest distance between the source node and the destination node in the plurality of paths
501
Add the first path to a set of paths. j = 1
502
j ≤ k?
503
Yes
No
Determine a next path from the plurality of possible paths based on a cost of the further path
504
No
Is the next path already in the set of paths?
505
Yes
Add the next path to the set of paths. j = j+1
506
Configure transmission of the data from the source node to the destination node using paths in the set of paths
507

Fig. 5

6/8

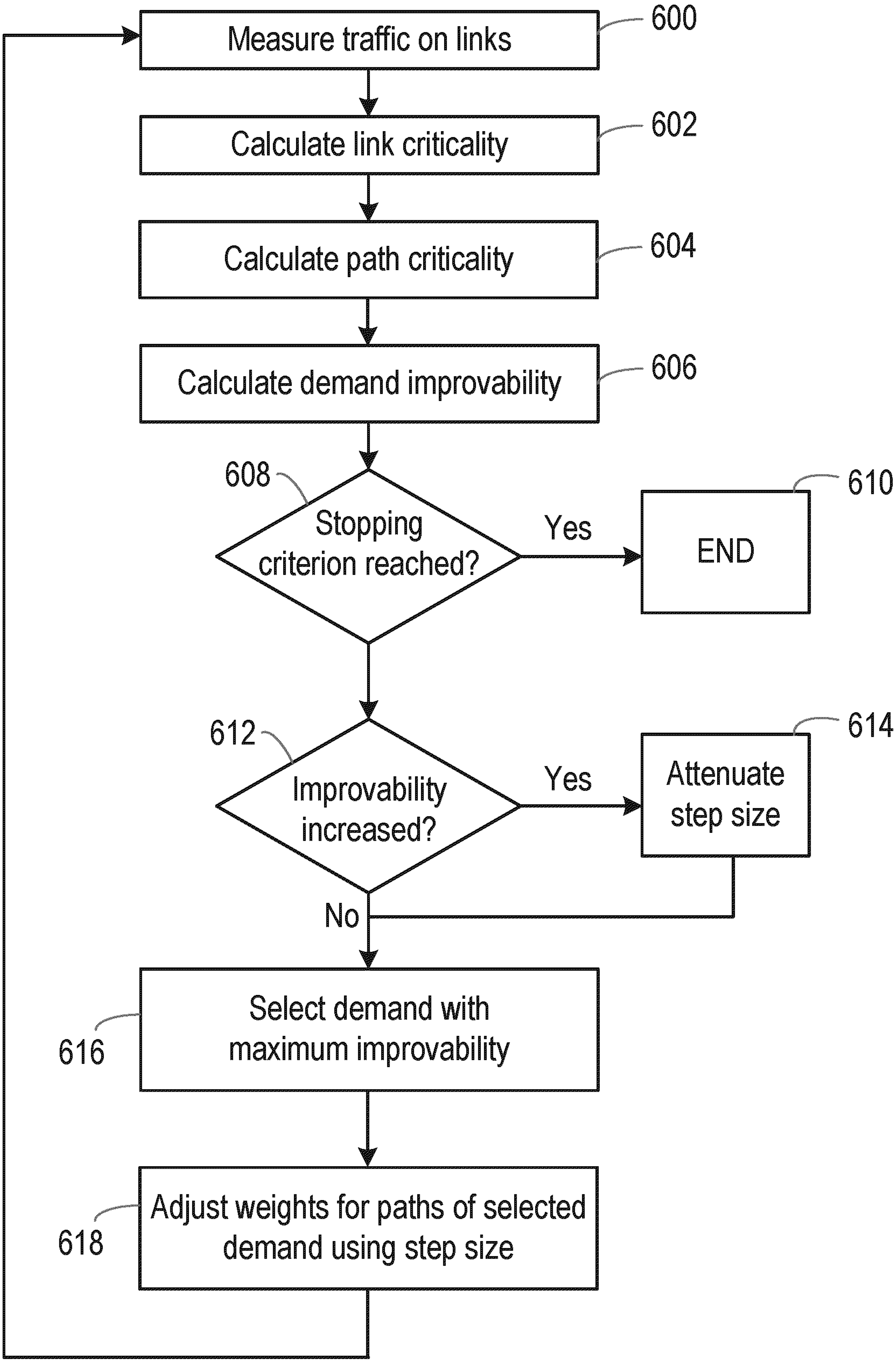
Measure traffic on links
600
Calculate link criticality
602
Calculate path criticality
604
Calculate demand improvability
606
608
Stopping criterion reached?
Yes
END
610
612
Improvability increased?
Yes
Attenuate step size
614
No
Select demand with maximum improvability
616
Adjust weights for paths of selected demand using step size
618

Fig. 6

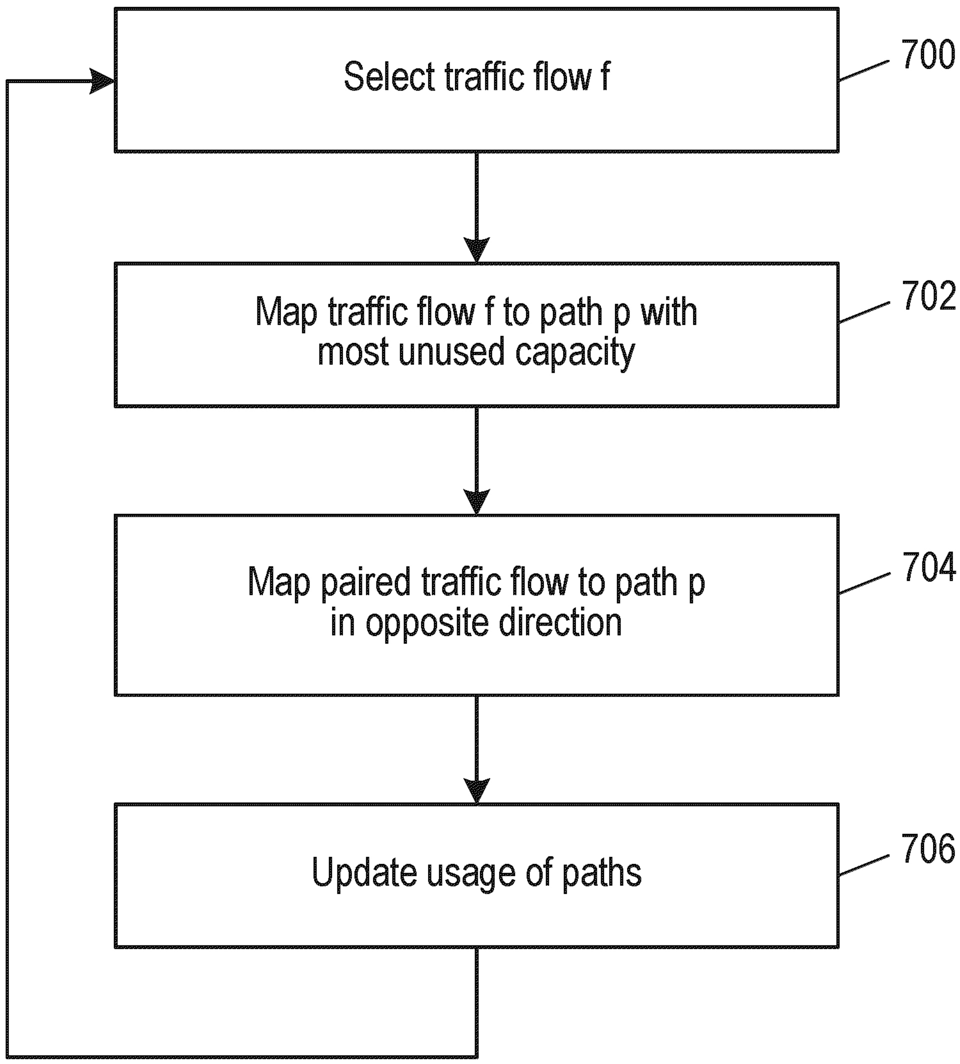
Select traffic flow f
700
Map traffic flow f to path p with most unused capacity
702
Map paired traffic flow to path p in opposite direction
704
Update usage of paths
706

Fig. 7

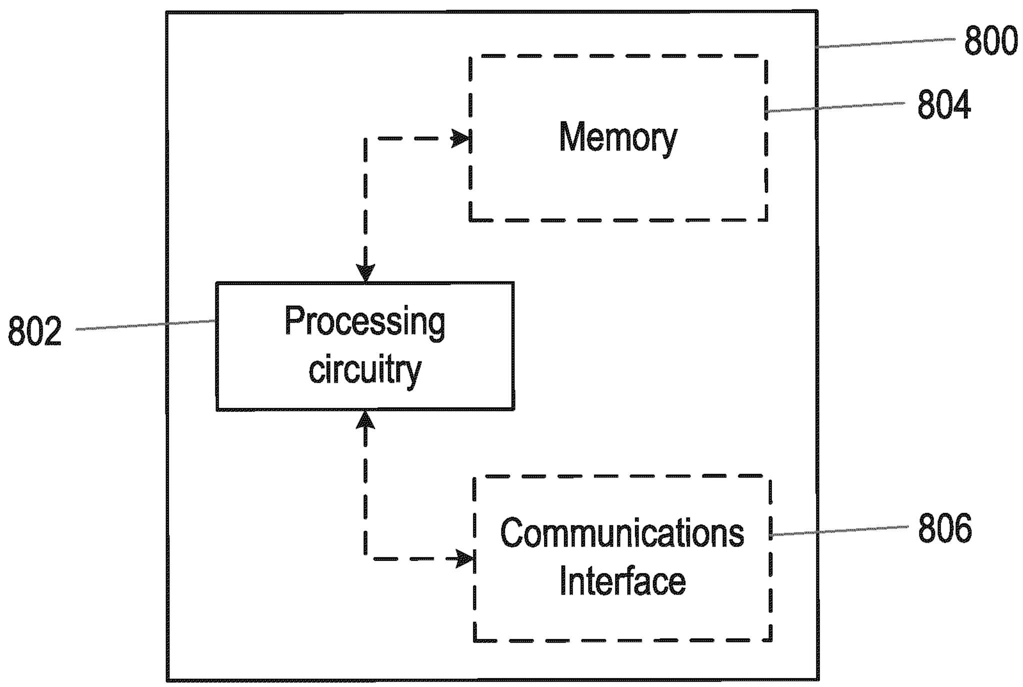
800
804
Memory
802
Processing circuitry
806
Communications Interface

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2021/064888

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L45/121 H04L45/12 H04L45/64
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LIEN-WU CHEN ET AL: "Exploiting Spectral Reuse in Routing, Resource Allocation, and Scheduling for IEEE 802.16 Mesh Networks", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE, USA, vol. 58, no. 1, 1 January 2009 (2009-01-01), pages 301-313, XP011226373, ISSN: 0018-9545, DOI: 10.1109/TVT.2008.923685	1-11, 14-24,27
Y	page 307, right-hand column algorithm 2; page 308 -----	12,13, 25,26
Y	US 2020/154335 A1 (CUI ZHI [US] ET AL) 14 May 2020 (2020-05-14) paragraphs [0036], [0037]; figures 2, 3 -----	12,13, 25,26

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 February 2022	28/02/2022

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Tsuchiya, Kuni

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2021/064888

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2020154335 A1	14-05-2020	US 2016366632 A1	15-12-2016
		US 2020154335 A1	14-05-2020