



US 20120127875A1

(19) **United States**

(12) **Patent Application Publication**
Zhao

(10) **Pub. No.: US 2012/0127875 A1**

(43) **Pub. Date: May 24, 2012**

(54) **METHOD AND DEVICE FOR CALCULATING K-SHORTEST PATHS**

Publication Classification

(51) **Int. Cl.**
H04B 10/08 (2006.01)

(52) **U.S. Cl.** **370/252**

(57) **ABSTRACT**

(75) Inventor: **Wei Zhao**, Shenzhen (CN)

(73) Assignee: **ZTE CORPORATION**, Shenzhen, Guangdong (CN)

(21) Appl. No.: **13/258,281**

(22) PCT Filed: **Apr. 20, 2010**

(86) PCT No.: **PCT/CN2010/071944**

§ 371 (c)(1),
(2), (4) Date: **Feb. 1, 2012**

The disclosure discloses a method and a device for calculating K-shortest paths, belonging to the field of a path calculation technology. The method comprises: calculating a shortest path tree of a whole network topological graph of Traffic Engineering (TE) links in current optical network; sorting, on the calculated shortest path tree, necessarily passed links or nodes in an optical path to be built; executing segmented calculation according to the sorted necessarily passed links and nodes, and assembling the K-shortest paths with paths obtained by the segmented calculation. The manner of sorting the necessarily passed nodes and links on the calculated shortest path tree effectively avoids the path assembly failure probability which is caused by the randomness of the necessarily passed nodes and links, further speeds up the quick determination of path accessibility, effectively improves the path assembly success rate and the path calculation reliability.

(30) **Foreign Application Priority Data**

Aug. 6, 2009 (CN) 200910090368.0

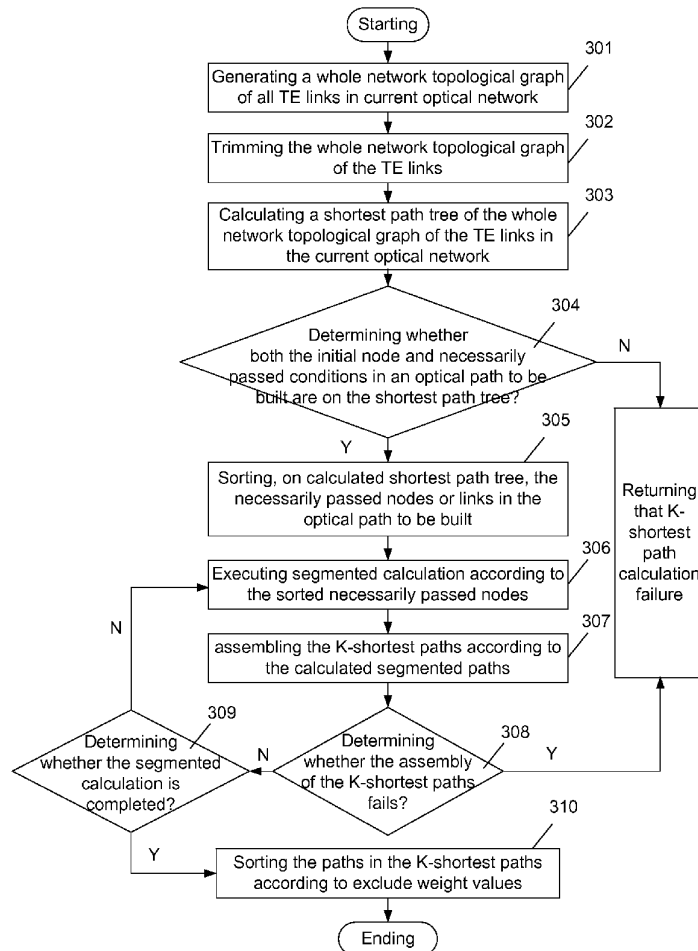


Fig. 1

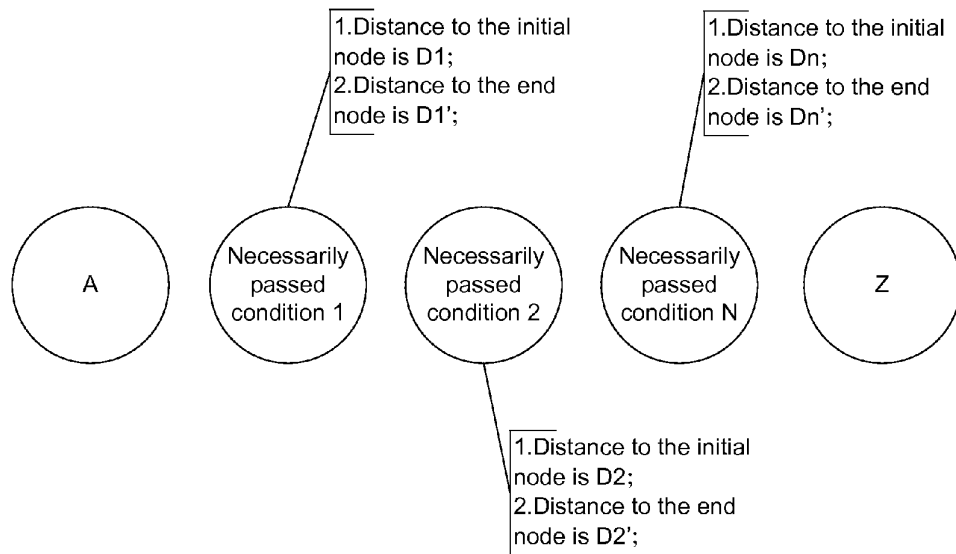


Fig. 2

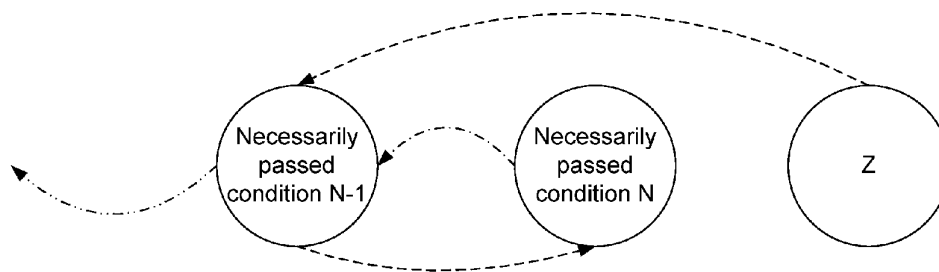


Fig. 3

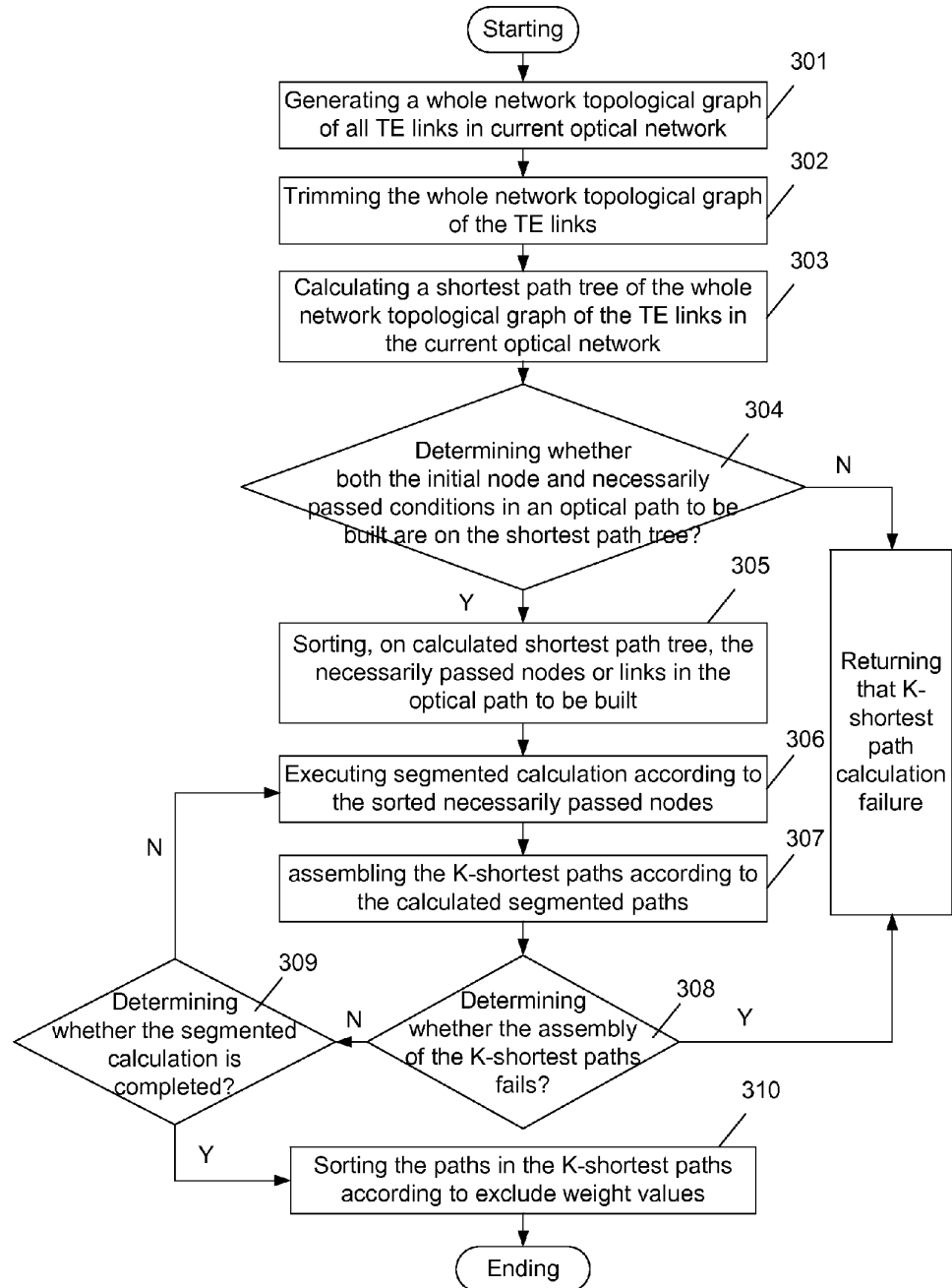


Fig. 4

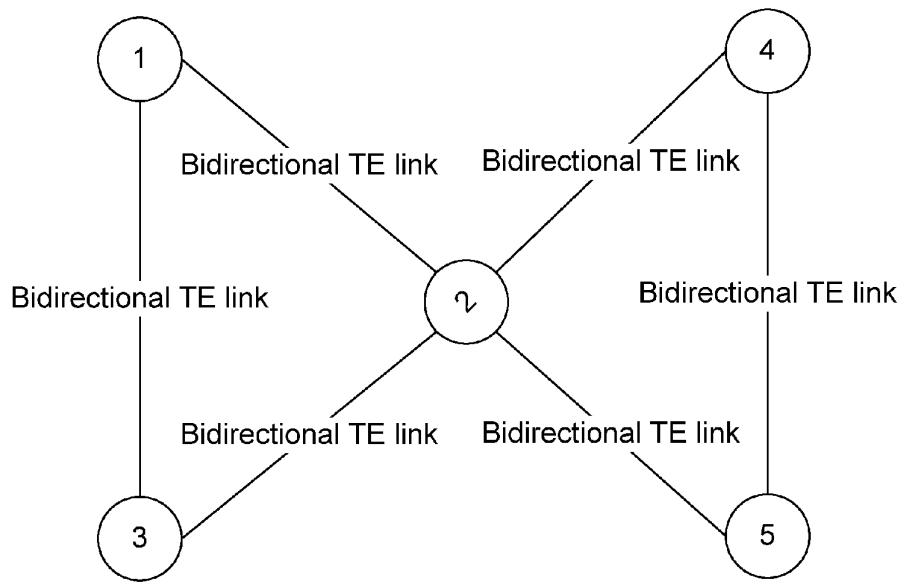


Fig. 5

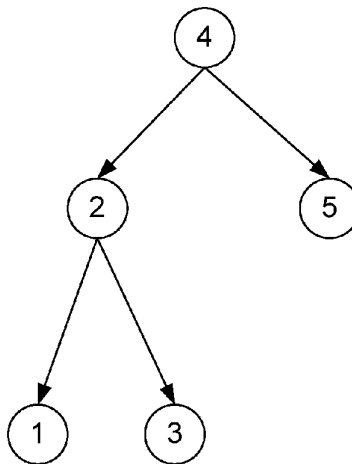
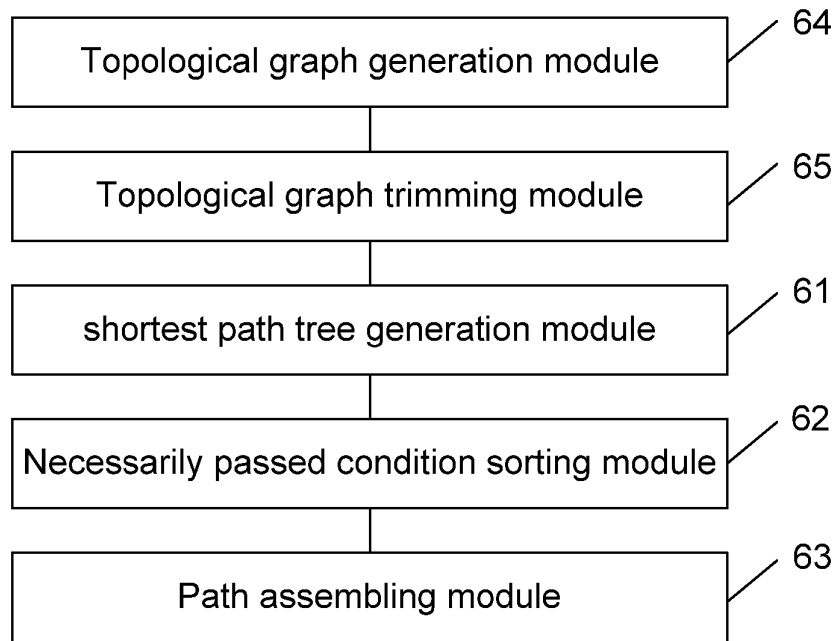


Fig.6



METHOD AND DEVICE FOR CALCULATING K-SHORTEST PATHS

TECHNICAL FIELD

[0001] The disclosure relates to the field of path calculation technology, particularly to a method and a device for calculating K-shortest paths in an optical network.

BACKGROUND

[0002] Path analysis is an important part of optical network analysis, and network analysis comprises optimal path analysis, K-shortest path analysis and optimal traversing path analysis. In Wavelength Division Multiplexing (WDM) optical network, path calculation and wavelength assignment are two sub-problems of Routing and Wavelength Assignment (RWA). The path calculation usually provides K shortest paths for the wavelength assignment, so as to improve the success rate of the wavelength assignment.

[0003] At present, most of the K-shortest path algorithms are based on Dijkstra algorithm. When the input condition of an K-shortest path algorithm is necessarily passing through is a plurality of links and nodes, a segmented calculation method is adopted usually; then the results of the segmented calculation are assembled to form complete K-shortest paths. When links or nodes are found to be overlapped with those of the path (except the nodes participating in the segmentation) calculated in the last time during the assembling process, the assembling process of the path fails; when all the assembled paths fail during the assembling process, then the current calculation for the K-shortest path fails (stopping the path calculation). Finally the wavelength assignment will be provided with no path and wavelength resources, thus the wavelength assignment fails.

[0004] In a segmented calculation, the distance from initial node to first necessarily passed condition or the distance from last necessarily passed condition to end node is calculated firstly according to configuration. As the necessarily passed conditions are given by a connection management module, and a plurality of necessarily passed conditions may be provided and not sorted according to a certain rule, which will result in the distances between the necessarily passed conditions and the initial or end nodes being determined randomly; such distances may be determined according to hop counts or link cost; and such necessarily passed conditions may be understood as necessarily passed nodes or links.

[0005] In FIG. 1, wherein node A is an initial node, node Z is an end node; necessarily passed condition 1, necessarily passed condition 2, . . . and necessarily passed condition N exist between nodes A and Z. If the distance between necessarily passed condition N and node Z is larger than the distance between necessarily passed condition N-1 and node Z, then a schematic diagram of a path shown in FIG. 2 may be obtained according to the above described algorithm; wherein the path is from the end node Z to necessarily passed condition N-1 firstly, then from necessarily passed condition N-1 to necessarily passed condition N, then from necessarily passed condition N back to necessarily passed condition N-1 and finally from necessarily passed condition N-1 to other necessarily passed conditions. According to the path assembly rule, the path assembly fails when a repeated link or node exists during assembling.

[0006] The necessarily passed conditions are not sorted according to a certain rule in the prior art, likely to cause the

existence of repeated links and nodes, thereby reducing the success rate of the K-shortest path assembly.

SUMMARY

[0007] In order to solve the problem above, the disclosure provides a method and a device for calculating K-shortest paths, which effectively improves the K-shortest paths assembly success rate and the reliability of the path calculation.

[0008] In order to solve the problem above, the technical solution of the disclosure is realized as follows.

[0009] A method for calculating K-shortest paths, wherein the method comprises:

[0010] Step A: calculating a shortest path tree of a whole network topological graph of Traffic Engineering (TE) links in current optical network;

[0011] Step B: sorting, on the calculated shortest path tree, necessarily passed links or nodes in an optical path to be built; and

[0012] Step C: executing segmented calculation according to the sorted necessarily passed links and nodes, and assembling K-shortest paths with paths obtained by the segmented calculation.

[0013] The Step A may specifically comprise:

[0014] calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network by taking an end node of the optical path to be built as a root.

[0015] The step of calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network may specifically comprise:

[0016] calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network by using a Shortest Path First (SPF) algorithm.

[0017] In Step B, the step of sorting the necessarily passed links or nodes in the optical path to be built may be as follows:

[0018] sorting based on distance between the necessarily passed links or nodes and the end node of the optical path to be built.

[0019] Before the Step A the method may further comprise:

[0020] generating the whole network topological graph of all the TE links in the current optical network; and

[0021] trimming the whole network topological graph of the TE links according to a preset limiting condition.

[0022] Before executing the Step B the method may further comprise:

[0023] determining whether an initial node, the end node and the necessarily passed nodes in the optical path to be built are all on the shortest path tree; when all the nodes are on the shortest path tree, executing the Step B; otherwise, returning that K-shortest path calculation failure.

[0024] After executing the Step C the method may further comprise:

[0025] determining whether the assembly of the K-shortest path fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, returning to execute Step C.

[0026] Before returning back to the Step C the method may further comprise:

[0027] determining whether the segmented calculation is completed; when the segmented calculation is not completed, returning back to Step C; otherwise, calculating exclude

weight value of each path in the K-shortest path, then sorting the paths in the K-shortest paths according to the calculated exclude weight values.

[0028] A device for calculating K-shortest paths, wherein the device comprises:

[0029] a shortest path tree generation module, which is configured to calculate a shortest path tree of a whole network topological graph of TE links in current optical network;

[0030] a necessarily passed condition sorting module, which is configured to sort, on the is calculated shortest path tree, necessarily passed links and nodes in an optical path to be built; and

[0031] a path assembling module, which is configured to execute segmented calculation according to the sorted necessarily passed links or nodes, and assembling K-shortest path with the paths obtained by the segmented calculation.

[0032] The calculation device may further comprise:

[0033] a topological graph generation module, which is configured to generate the whole network topological graph of all the TE links in the current optical network; and

[0034] a topological graph trimming module, which is configured to trim the whole network topological graph of the TE links according to a preset limiting condition.

[0035] At least one technology in above technical program has the advantages of effectively avoiding the rate of path assembly failure caused by randomness of nodes and links necessarily to be passed through ordering nodes and links necessarily to be passed on calculated shortest path tree; speeding up quick determination of path accessibility through judging whether the initial node and the end node are on shortest path tree; effectively improving the K-shortest path assembly success rate and is beneficial to improve path calculation reliability; and effectively eliminating the links and nodes passed through by network fault and improving the optical network reliability through sorting loose exclude condition of necessarily passed links and nodes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0036] FIG. 1 shows a schematic diagram of the necessarily passed conditions between the initial node and the end node in the prior art;

[0037] FIG. 2 shows a schematic diagram of the K-shortest path calculated through the prior art;

[0038] FIG. 3 shows a flowchart of a method for calculating K-shortest paths in this embodiment;

[0039] FIG. 4 shows a typical network in this embodiment;

[0040] FIG. 5 shows a shortest path tree calculated based on the typical networking in FIG. 4;

[0041] FIG. 6 shows a structural diagram of a device for calculating K-shortest paths in this embodiment.

DETAILED DESCRIPTION

[0042] In order to clear the object, technical solution and advantages of the embodiment of the disclosure, the embodiments of the disclosure will be described below with reference to the drawings and embodiments in detail. And the drawings and embodiments of the disclosure are only used for explaining the disclosure, not intended to limit it.

[0043] FIG. 3 shows a flowchart of a method for calculating K-shortest paths in the current embodiment. Specific steps are as follows:

[0044] Step 301: generating a whole network topological graph of all TE links in current optical network;

[0045] the whole network topological graph of the TE links is obtained from a path protocol (Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (ISIS)) TE expand flooding information, and stored in a TE Database (TED).

[0046] Step 302: trimming the whole network topological graph of the TE links, then executing Step 303.

[0047] That is, the whole network topological graph of the TE links is trimmed according to a preset limiting condition. The limiting condition comprises an avoiding limiting condition and an unilateral limiting condition, wherein

[0048] 1) the avoiding limiting condition refers to an avoiding node needed to be excluded during the K-shortest path calculation process; wherein the avoiding node is a node necessarily to be avoided in the shortest path satisfying a user constraint condition.

[0049] 2) the unilateral limiting condition refers to unilateral TE links in the whole network is topological graph except the TE links.

[0050] After trimming the whole network topological graph of the TE links by using the avoiding limiting condition, because an unilateral TE link may exist between two nodes, as for bidirectional communication service, the unilateral TE link cannot participate in K path calculation; thus, the whole network topological graph of the TE links can be trimmed through the unilateral limiting condition, excluding the unilateral TE link in the whole network topological graph of the TE links. The limiting conditions can be preset by a network administrator.

[0051] Step 303: calculating the shortest path tree of the whole network topological graph of the TE links in current optical network, then executing Step 304.

[0052] In this embodiment, by taking the end node of optical path to be built as a root, the shortest path tree of the whole network topological graph of the TE links can be calculated through the SPF algorithm; wherein the SPF algorithm is a basic algorithm of OSPF path calculation. The whole network topological graph of the TE links and end node (node Z) of the optical path to be built are input firstly in calculation, and the calculation result is as follows: a tree taking node Z as the root, wherein the tree represents a set of nodes and links. Certainly, the shortest path tree can be calculated through other algorithms, and the embodiment does not intend to limit the specific manner of adopted algorithm.

[0053] Step 304: determining whether the initial node, end node and necessarily passed nodes in the optical path to be built are on the shortest path tree; when such nodes are on the shortest path tree, executing Step 305; otherwise, returning that K-shortest path calculation failure.

[0054] The shortest path tree calculated in Step 303 can represent the set of a series of links and nodes, and the set forms the shortest path tree. In this step, just inquiring whether the initial node, end node and necessarily passed nodes (the necessarily passed links can be transferred into two necessarily passed nodes) are in the set, so as to determine whether the initial node, end node and necessarily passed node are on the is shortest path tree; when such nodes are all in the set, executing Step 305; otherwise, returning that the K-shortest path calculation failure.

[0055] The necessarily passed node mentioned above refers to a node which necessarily to be passed through in the shortest path satisfying user constraint condition.

[0056] There are two reasons for that the initial node and the end node are not on the shortest path tree in the embodiment:

[0057] 1) two or more unconnected topological graphs exist, then the initial node and end node are on different topological graphs respectively;

[0058] 2) the calculation object of the SPF algorithm is a bidirectional graph (side is bidirectional, or called as an undirected graph), so before the SPF algorithm, revising the original topological graph of the TE links to remove the unilateral TE link; so that the initial node and end node may be in different topological graphs, or the node may be isolated node (without any side of the TE link) and existing in no topological graph.

[0059] The initial node or end node are determined to be not on the shortest path tree when the above two situations happen, but not limited to the two situations. By determining whether the initial node and the end node are on the shortest path tree, the determination speed of path accessibility (an unconnected graph) is improved effectively.

[0060] Step 305: sorting, on the calculated shortest path tree, the necessarily passed links or nodes in the optical path to be built.

[0061] For example, sorting the necessarily passed nodes according to a preset sorting strategy. Wherein the sorting strategy can sort according to the distance from the necessarily passed nodes to the end node of the optical path to be built, or sort according to the distance from the necessarily passed nodes to the initial node of the optical path to be built. By sorting the necessarily passed nodes according to the shortest path tree, segmented path calculation can be applied to the sorted result, and can effectively avoids repeated links or nodes existing in path, thereby improving the K-shortest path assembly efficiency.

[0062] Step 306: executing segmented path calculation according to the sorted is necessarily passed nodes, then executing Step 307.

[0063] In this step, K-shortest calculation algorithm can be called to calculate the path in segmentation. In order to obtain the K-shortest paths, the constraint condition of the K-shortest algorithm comprises information of all the other necessarily passed nodes as performing each segmented calculation, thereby ensuring that repeated nodes and links do not exist in the calculated result.

[0064] Step 307: assembling the K-shortest paths according to the calculated segmented path, then executing Step 308.

[0065] The assembly of the calculated segmented paths can be performed in an existing path assembling manner, and the number of all the current path successfully assembled paths is recorded.

[0066] Step 308: determining whether the assembly of the K-shortest paths fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, executing Step 309.

[0067] In this step, determining whether the assembly of the K-shortest paths fails through determining whether repeated links exist; when the repeated links exist, then the path assembly fails. Of course, also can determine whether the assembly of K-shortest paths fails through determining whether repeated nodes exist; when the repeated nodes exist, then the path assembly fails.

[0068] Step 309: determining whether the segmented calculation is completed; when the segmented calculation is completed, executing Step 310; otherwise, returning back to Step 306.

[0069] That is to repeat Steps 306-308 until the segmented calculation and assembly are completed.

[0070] Because the final assembled paths calculated through the segmented calculation may be not sorted according to the minimum hop counts or minimum link cost, thus it could be determined whether a secondary sorting is needed according to the number of the current paths assembled successfully. When the number is 1, returning that the is K-shortest path calculation failure; when the number is more than 1, applying a secondary sorting to the necessarily passed nodes or links.

[0071] The minimum hop counts mean that the desired calculated path passes through the least number of the nodes in the possibly existed paths.

[0072] The cost of the minimum link cost refers to the attribute of the TE links. The cost has relations with the bandwidth; the larger the bandwidth is, the smaller the TE link cost is. When calculating the path, it expects to obtain a path which passes through links with the minimum sum of the cost, which means to pass a path with large bandwidth.

[0073] Step 310: sorting the paths in the K-shortest paths according to exclude weight values.

[0074] That is, the paths in the K-shortest paths are sorted according to the minimum hop counts or minimum link cost firstly; then the exclude weight value of each path in the K-shortest paths is calculated according to a preset excluding strategy; then the paths in the K-shortest paths are resorted according to the exclude weight values. The exclude weight value can be understood as fault probability of links or nodes, but not limited to it.

[0075] The preset excluding strategy may be a loose exclude condition, which specifically means that: when the K-shortest paths are obtained, a group of the exclude weight values are obtained with a certain weighting coefficient (the exclude weight value of the link is large, the exclude weight value of the node is small) according to the times of the loose excluded links and nodes appear in the paths; then the paths are sorted in order according to the exclude weight values; thus the paths with the exclude weight values are used preferentially, thereby avoiding failed links easily.

[0076] The link fault probability in an actual network is far more higher than that of the node; thus a path avoiding links is selected preferentially, which means the link exclude weight value is far more larger than the node exclude weight value. This is because that if both the links and the nodes are needed to be avoided in the path calculation, it is needed to decide which are selected first and which are selected later. In this embodiment, the links are selected to be avoided firstly, because the link fault probability is far more larger than that of the node. The exclude weight value of each node or link can be set according to actual situation.

[0077] The calculated K-shortest path can be used in wavelength assignment after the calculation of Steps 301-310.

[0078] Based on the above technical solution, the manner of sorting the necessarily passed links or nodes on the calculated shortest path tree effectively avoids the path assembly failure probability, which is caused by the randomness of the necessarily passed links or nodes, further speeds up quick determination of path accessibility, effectively improves the path assembly success rate and is beneficial to improve path

calculation reliability. The manner of sorting the loose exclude conditions effectively is beneficial to exclude the links and nodes passed through by the network fault and effectively improve the reliability of optical network.

[0079] In order to describe the processing conditions of the embodiment in detail, a typical network is tested below. The topological graph of the test sample is shown in FIG. 4, wherein the costs of all the TE links are the same, and the costs referring to the available bandwidth capacity of the optical fiber between the nodes.

[0080] Step 1: constructing a network, configuring internal optical fiber connection and configuring transmission interface optical fiber connection; for convenience, this test sample configuring no unilateral side, each node is in full configuration, no unconnected graph is generated, and each node is started respectively.

[0081] Step 2: establishing service from node 1 to node 4 and the constraint condition is that node 5 and node 3 must be passed through; then node 5 and node 3 are called necessarily passed node 5 and necessarily passed node 3 respectively.

[0082] That is, when an optical path from node 1 to node 4 is to be established, node 1 is node A (i.e. the initial node), similarly, node 4 is node Z (i.e. the end node), and node 5 and node 3 are necessarily passed nodes.

[0083] Step 3: obtaining a shortest path tree as shown in FIG. 5 by calling SPF algorithm.

[0084] Step 4: examining whether all the necessarily passed nodes are on the shortest path tree, which means that node 5 and node 3 are on the shortest path tree; sorting the necessarily passed nodes, and finding node 5 is closest to node Z (node 4) and node 3 is the secondary one.

[0085] Step 5: calculating K-shortest paths in segmentation.

[0086] Namely, calculating from node 5 to node Z (excluding node 3 and node A) to obtain K-shortest paths: 5-4 and 5-2-4;

[0087] furthermore, calculating from node 3 to node 5 (excluding node Z and node A) to obtain a K-shortest path: 3-2-5;

[0088] assembling paths to obtain a K-shortest path: 3-2-5-4; then the formerly calculated K-shortest path 5-2-4 is invalid (there is overlapped node 2);

[0089] then continuing to calculate from node A to node 3 (excluding node 5 and node Z) to obtain a K-shortest path: 1-3;

[0090] assembling paths to obtain a K-shortest path: 1-3-2-5-4; then K-shortest path calculation is completed.

[0091] it can be seen from the embodiment that, by using the shortest path tree, it can be determined at early stage of the K-shortest path calculation whether a path between the initial node A and end node Z can be calculated. As for a large network, such K-shortest path algorithm is beneficial to improve the path assembly success rate of a path inquiry request with several necessarily passed conditions; and the loose exclude algorithm is beneficial to exclude fault links and nodes, thereby improving the path protection probability of optical network.

[0092] In order to realize above method embodiments, other embodiments of the disclosure also provide a device for calculating the K-shortest paths. In addition, what is needed to be described firstly is that the following embodiments are used for realizing the above method embodiment, so that the modules in the device are set for realizing the steps of the above method; but, the disclosure is not limited to the following embodiments, any devices and modules capable of real-

izing the method shall fall within the protection scope of the disclosure. In the following description, the content same to the method is omitted in order to save the length of this specification.

[0093] FIG. 6 shows a structural diagram of a device for calculating K-shortest paths in this embodiment. Based on the figure, the calculation device comprises:

[0094] a shortest path tree generation module 61, which is configured to calculate a shortest path tree of a whole network topological graph of TE links in current optical network;

[0095] a necessarily passed condition sorting module 62, which is configured to sort, on calculated shortest path tree, necessarily passed links and nodes in an optical path to be built;

[0096] a path assembling module 63, which is configured to execute segmented calculation and path assembly according to the sorted necessarily passed links or nodes.

[0097] In another embodiment of the disclosure, the calculation device further comprises:

[0098] a topological graph generation module 64, which is configured to generate a whole network topological graph of all TE links in current optical network;

[0099] a topological graph trimming module 65, which is configured to trim the whole network topological graph of the TE links according to a preset limiting condition.

[0100] The above is only the preferred embodiments of the disclosure. It should be noted that, for those skilled in the art, several improvements and modifications can be done without separating from the principle of the disclosure, such improvements and modifications shall also fall within the protection scope of the disclosure.

1. A method for calculating K-shortest paths, comprising:
 - Step A: calculating a shortest path tree of a whole network topological graph of Traffic Engineering (TE) links in current optical network;
 - Step B: sorting, on the calculated shortest path tree, necessarily passed links or nodes in an optical path to be built; and
 - Step C: executing segmented calculation according to the sorted necessarily passed links and nodes, and assembling K-shortest paths with paths obtained by the segmented calculation.
2. The calculation method according to claim 1, wherein the Step A specifically comprises:
 - calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network by taking an end node of the optical path to be built as a root.
3. The calculation method according to claim 2, wherein the step of calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network specifically comprises:
 - calculating the shortest path tree of the whole network topological graph of the TE links in the current optical network by using a Shortest Path First (SPF) algorithm.
4. The calculation method according to claim 1, wherein, in Step B, the step of sorting the necessarily passed links or nodes in the optical path to be built is as follows:
 - sorting based on distance between the necessarily passed links or nodes and the end node of the optical path to be built.
5. The calculation method according to claim 1, before the Step A the method further comprising:

- generating the whole network topological graph of all the TE links in the current optical network; and
trimming the whole network topological graph of the TE links according to a preset limiting condition.
6. The calculation method according to claim 1, before executing the Step B the method further comprising:
determining whether an initial node, the end node and the necessarily passed nodes in the optical path to be built are all on the shortest path tree; when all the nodes are on the shortest path tree, executing the Step B; otherwise, returning that K-shortest path calculation failure.
7. The calculation method according to claim 1, after executing the Step C the method further comprising:
determining whether the assembly of the K-shortest path fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, returning to execute Step C.
8. The calculation method according to claim 7, before returning back to the Step C the method further comprising:
determining whether the segmented calculation is completed; when the segmented calculation is not completed, returning back to Step C; otherwise, calculating exclude weight value of each path in the K-shortest path, then sorting the paths in the K-shortest paths according to the calculated exclude weight values.
9. A device for calculating K-shortest paths, comprising:
a shortest path tree generation module, which is configured to calculate a shortest path tree of a whole network topological graph of TE links in current optical network;
a necessarily passed condition sorting module, which is configured to sort, on the calculated shortest path tree, necessarily passed links and nodes in an optical path to be built; and
a path assembling module, which is configured to execute segmented calculation according to the sorted necessarily passed links or nodes, and assembling K-shortest path with the paths obtained by the segmented calculation.
10. The calculation device according to claim 9, further comprising:
a topological graph generation module, which is configured to generate the whole network topological graph of all the TE links in the current optical network; and
a topological graph trimming module, which is configured to trim the whole network topological graph of the TE links according to a preset limiting condition.
11. The calculation method according to claim 2, before the Step A the method further comprising:
generating the whole network topological graph of all the TE links in the current optical network; and
trimming the whole network topological graph of the TE links according to a preset limiting condition.
12. The calculation method according to claim 3, before the Step A the method further comprising:
generating the whole network topological graph of all the TE links in the current optical network; and
trimming the whole network topological graph of the TE links according to a preset limiting condition.
13. The calculation method according to claim 4, before the Step A the method further comprising:
generating the whole network topological graph of all the TE links in the current optical network; and
trimming the whole network topological graph of the TE links according to a preset limiting condition.
14. The calculation method according to claim 2, before executing the Step B the method further comprising:
determining whether an initial node, the end node and the necessarily passed nodes in the optical path to be built are all on the shortest path tree; when all the nodes are on the shortest path tree, executing the Step B; otherwise, returning that K-shortest path calculation failure.
15. The calculation method according to claim 3, before executing the Step B the method further comprising:
determining whether an initial node, the end node and the necessarily passed nodes in the optical path to be built are all on the shortest path tree; when all the nodes are on the shortest path tree, executing the Step B; otherwise, returning that K-shortest path calculation failure.
16. The calculation method according to claim 4, before executing the Step B the method further comprising:
determining whether an initial node, the end node and the necessarily passed nodes in the optical path to be built are all on the shortest path tree; when all the nodes are on the shortest path tree, executing the Step B; otherwise, returning that K-shortest path calculation failure.
17. The calculation method according to claim 2, after executing the Step C the method further comprising:
determining whether the assembly of the K-shortest path fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, returning to execute Step C.
18. The calculation method according to claim 3, after executing the Step C the method further comprising:
determining whether the assembly of the K-shortest path fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, returning to execute Step C.
19. The calculation method according to claim 4, after executing the Step C the method further comprising:
determining whether the assembly of the K-shortest path fails; when the assembly fails, returning that the K-shortest path calculation failure; otherwise, returning to execute Step C.
20. The calculation method according to claim 17, before returning back to the Step C the method further comprising:
determining whether the segmented calculation is completed; when the segmented calculation is not completed, returning back to Step C; otherwise, calculating exclude weight value of each path in the K-shortest path, then sorting the paths in the K-shortest paths according to the calculated exclude weight values.
21. The calculation method according to claim 18, before returning back to the Step C the method further comprising:
determining whether the segmented calculation is completed; when the segmented calculation is not completed, returning back to Step C; otherwise, calculating exclude weight value of each path in the K-shortest path, then sorting the paths in the K-shortest paths according to the calculated exclude weight values.
22. The calculation method according to claim 19, before returning back to the Step C the method further comprising:
determining whether the segmented calculation is completed; when the segmented calculation is not completed, returning back to Step C; otherwise, calculating exclude weight value of each path in the K-shortest path, then sorting the paths in the K-shortest paths according to the calculated exclude weight values.

* * * * *