ABSTRACT

Embodiments herein provide a system and a method for optimizing a network topology of a communication network is disclosed. The method includes computing an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on at least one parameter associated with each network element. Further, the method includes determining whether the entropy is less than a maximum entropy value. Furthermore, the method includes performing at least one action on the at least one network element in the communication network to optimize the network topology in response to determining that the entropy is less than the maximum entropy value.

FIGS. 3 and 4

STATEMENT OF CLAIMS

We claim:

1. A method for optimizing a network topology of a communication network, the method comprising:

computing an entropy of said network topology based on a probability distribution of said communication network, wherein said probability distribution is derived based on at least one parameter associated with each said network element;

determining whether said entropy is less than a maximum entropy value; and

performing at least one action on at least one network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value.

2. The method of claim 1, wherein performing at least one action on at least one said network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value comprises:

dynamically modifying a weight associated with at least one said network element to maximize said entropy of said communication network, wherein said weight associated with said at least one network element is dynamically modified based on said at least one parameter of said network element.

3. The method of claim 2, wherein said dynamically modifying said weight associated with at least one said network element comprises increasing said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

4. The method of claim 2, wherein said dynamically modifying said weight associated with at least one said network element comprises decreasing said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

5. The method of claim 1, wherein performing at least one action on at least one said network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value comprises dynamically modifying a number of said network elements in said network topology to optimize said network topology.

6. The method of claim 1, wherein performing at least one action on at least one said network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value comprises dynamically modifying a direction of communication between said network elements in said network topology to optimize said network topology.

7. The method of claim 1, wherein deriving said probability distribution of said communication network based on said at least one parameter associated with each said network element comprises:

determining a probability value of each said network element based on said at least one parameter; and

determining said probability distribution of said communication network based on said probability value of each said network element.

8. The method of claim 7, wherein deriving said probability value based on at least one parameter of said network element comprises:

determining a Shortest Path Betweenness Centrality (SPBC) for each network element based on a number of shortest paths passing through said network element; and

deriving said probability value for each network element based on said SPBC for each network element.

9. The method of claim 1, wherein said parameter comprises at least one of a number of shortest paths, a number of packets, a number of sessions, link utilization data, link capacity data, link residual capacity data, a number of flows, and congestion.

10. The method of claim 1, wherein said network topology is a centralized network topology when said entropy is less than said maximum entropy value.

11. The method of claim 1, wherein said network topology is a de-

centralized network topology when said entropy is equal to said maximum entropy value.

12. A method for optimizing a network topology of a communication network, the method comprising:

computing an entropy of said network topology based on a probability distribution of said communication network, wherein said probability distribution is derived based on a probability value of each network element, and wherein said probability value is derived based on a number of shortest paths passing through said network element;

determining whether said entropy is less than a maximum entropy value;

dynamically modifying a weight associated with at least one said network element to maximize said entropy of said communication network in response to determining that said entropy is less than said maximum entropy value.

13. The method of claim 12, wherein said dynamically modifying said weight associated with at least one said network element comprises increasing said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

14. The method of claim 12, wherein said dynamically modifying said weight associated with at least one said network element comprises

decreasing said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

15. The method of claim 12, wherein deriving said probability value based on at least one parameter of said network element comprises:

determining a Shortest Path Betweenness Centrality (SPBC) for each network element based on a number of shortest paths passing through said network element; and

deriving said probability value for each network element based on said SPBC for each network element.

16. The method of claim 12, wherein said parameter comprises at least one of a number of shortest paths, a number of packets, a number of sessions, link utilization data, link capacity data, link residual capacity data, a number of flows, and congestion.

17. The method of claim 12, wherein said network topology is a centralized network topology when said entropy is less than said maximum entropy value.

18. The method of claim 12, wherein said network topology is a de-centralized network topology when said entropy is equal to said maximum entropy value.

19. A system for optimizing a network topology of a communication network, the system comprises:

a controller module configured to:

compute an entropy of said network topology based on a probability distribution of said communication network, wherein said probability distribution is derived based on at least one parameter associated with each said network element;

determine whether said entropy is less than a maximum entropy value; and

perform at least one action on at least one network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value.

20. The system of claim 19, wherein said controller module is configured to dynamically modify a weight associated with at least one said network element to maximize said entropy of said communication network, wherein said weight associated with said at least one network element is dynamically modified based on said at least one parameter of said network element.

21. The system of claim 20, wherein said controller module is configured to increase said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

22. The system of claim 20, wherein said controller module is configured to

decrease said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

23. The system of claim 19, wherein said controller module is configured to dynamically modify a number of said network elements in said network topology to optimize said network topology.

24. The system of claim 19, wherein said controller module is configured to dynamically modify a direction of communication between said network elements in said network topology to optimize said network topology.

25. The system of claim 19, wherein said controller module is configured to:

determine a probability value of each said network element based on said at least one parameter; and

determine said probability distribution of said communication network based on said probability value of each said network element.

26. The system of claim 25, wherein said controller module is configured to:

determine a Shortest Path Betweenness Centrality (SPBC) for each network element based on a number of shortest paths passing through said network element; and

derive said probability value for each network element based on said SPBC for each network element.

27. The system of claim 19, wherein said parameter comprises at least one of a number of shortest paths, a number of packets, a number of sessions, link utilization data, link capacity data, link residual capacity data, a number of flows, and congestion.

28. The system of claim 19, wherein said network topology is a centralized network topology when said entropy is less than said maximum entropy value.

29. The system of claim 19, wherein said network topology is a de-centralized network topology when said entropy is equal to said maximum entropy value.

30. A system for optimizing a network topology of a communication network, the system comprises:

a controller module configured to:

compute an entropy of said network topology based on probability distribution of said communication network, wherein said probability distribution is derived based on a probability value of each network element, and wherein said probability value is derived based on a number of shortest paths passing through said network element;

determine whether said entropy is less than a maximum entropy value;

dynamically modify a weight associated with at least one said network element to maximize said entropy of said communication network in response to determining that said entropy is less than said maximum entropy value.

31. The system of claim 30, wherein said controller module is configured to increase said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

32. The system of claim 30, wherein said controller module is configured to decrease said weight associated with at least one said network element to a degree to maximize said entropy of said communication network, wherein said degree is derived using an optimal shortest-path routing technique.

33. The system of claim 30, wherein said controller module is configured to:

determine a Shortest Path Betweenness Centrality (SPBC) for each network element based on a number of shortest paths passing through said network element; and

derive said probability value for each network element based on said SPBC for each network element.

34. The system of claim 30, wherein said parameter comprises at least one of a number of shortest paths, a number of packets, a number of sessions, link utilization data, link capacity data, link residual capacity data, a

number of flows, and congestion.

35. The system of claim 30, wherein said network topology is a centralized network topology when said entropy is less than said maximum entropy value.

36. The system of claim 30, wherein said network topology is a de-centralized network topology when said entropy is equal to said maximum entropy value.

37. A computer program product comprising computer executable program code recorded on a computer readable non-transitory storage medium, said computer executable program code when executed, causing the actions including:

computing an entropy of said network topology based on a probability distribution of said communication network, wherein said probability distribution is derived based on at least one parameter associated with each said network element;

determining whether said entropy is less than a maximum entropy value; and

performing at least one action on at least one network element in said communication network to optimize said network topology in response to determining that said entropy is less than said maximum entropy value.

38. A computer program product comprising computer executable program code recorded on a computer readable non-transitory storage medium, said

computer executable program code when executed, causing the actions including:

computing an entropy of said network topology based on a probability distribution of said communication network, wherein said probability distribution is derived based on a probability value of each network element, and wherein said probability value is derived based on a number of shortest paths passing through said network element;

determining whether said entropy is less than a maximum entropy value; and

dynamically modifying a weight associated with at least one said network element to maximize said entropy of said communication network in response to determining that said entropy is less than said maximum entropy value.

Dated: 5th Day of November, 2014

Signature

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Total No Of Sheets: 10

Application Number:

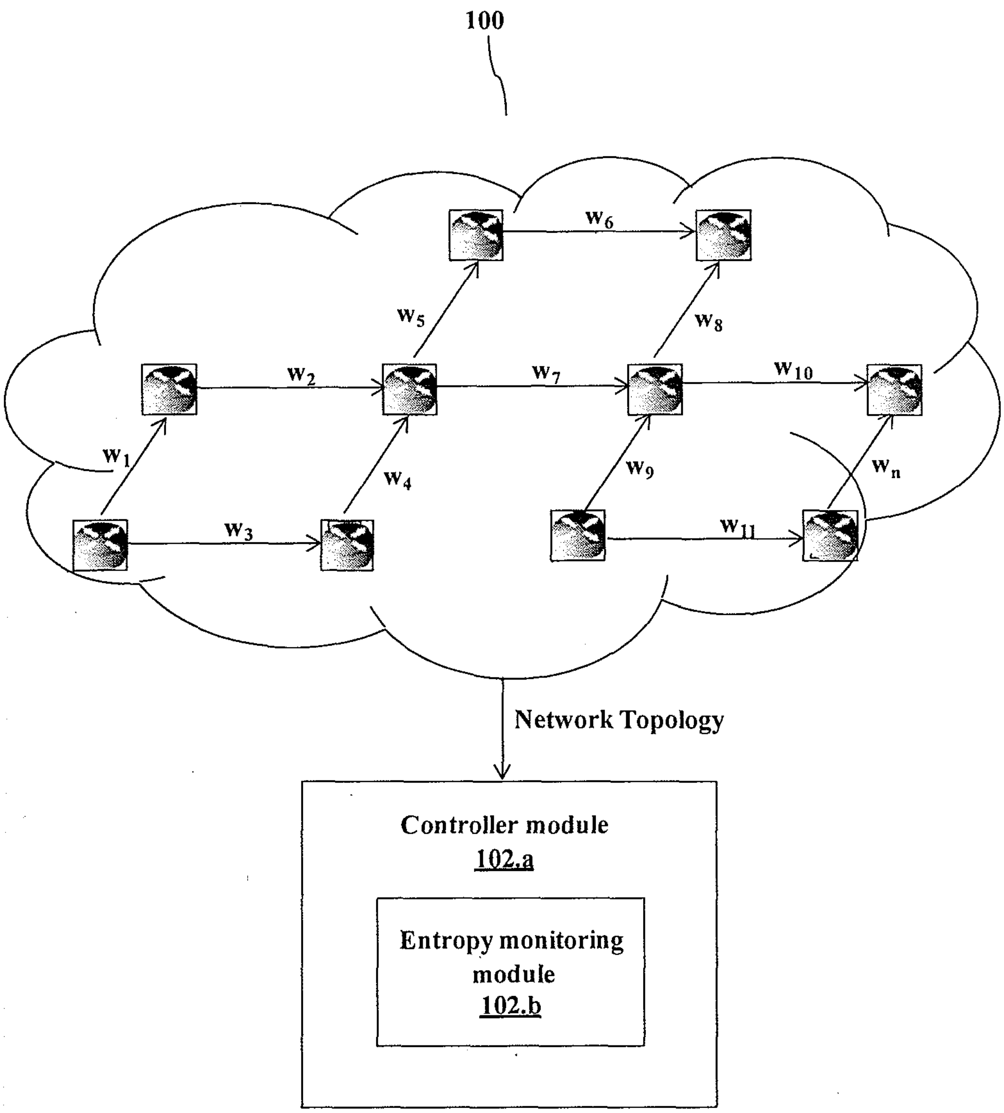


FIG. 1a

Arun Kishore Narasani
Patent Agent

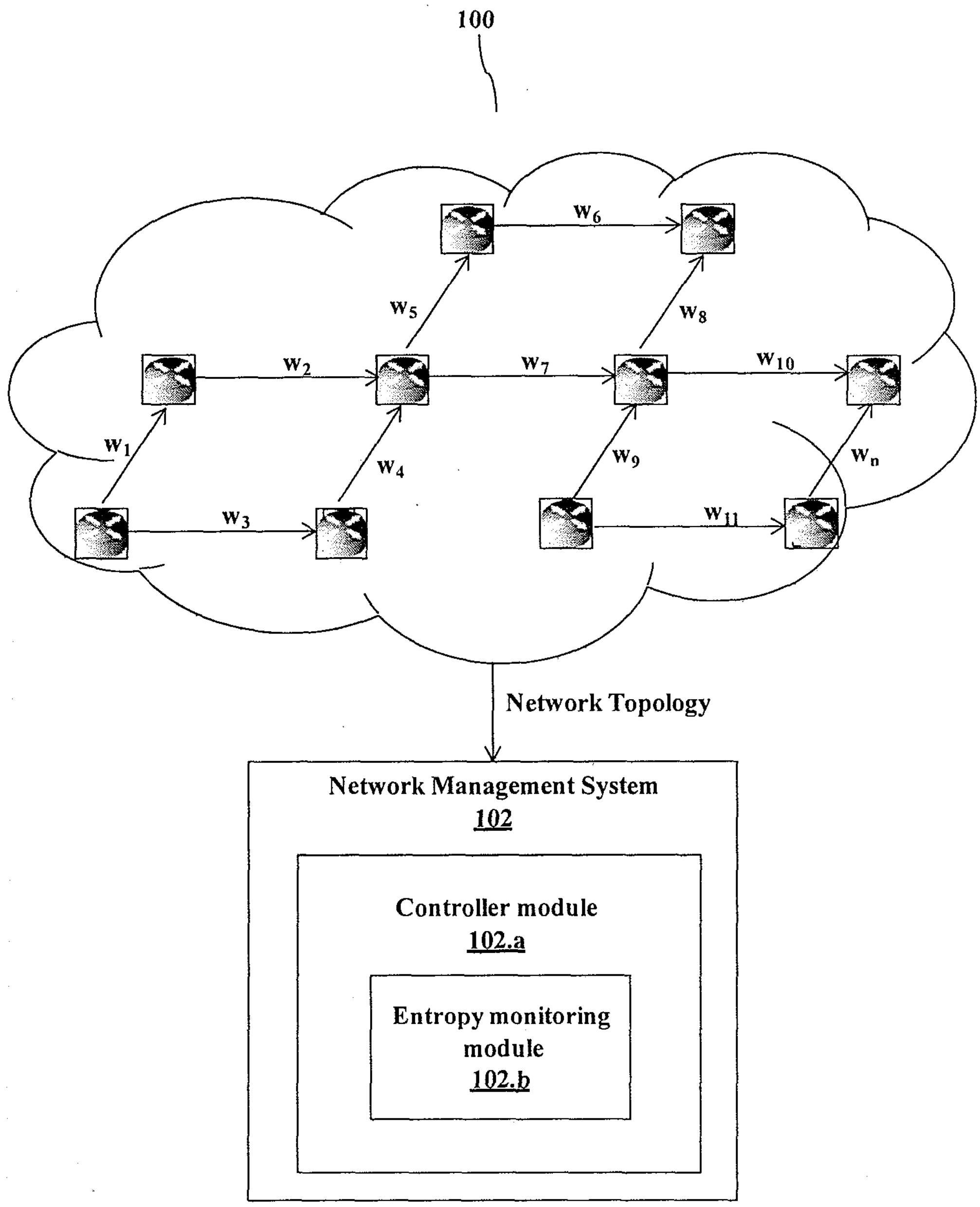


FIG. 1b

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Total No Of Sheets: 10

Application Number: Sheet No.: 3/10

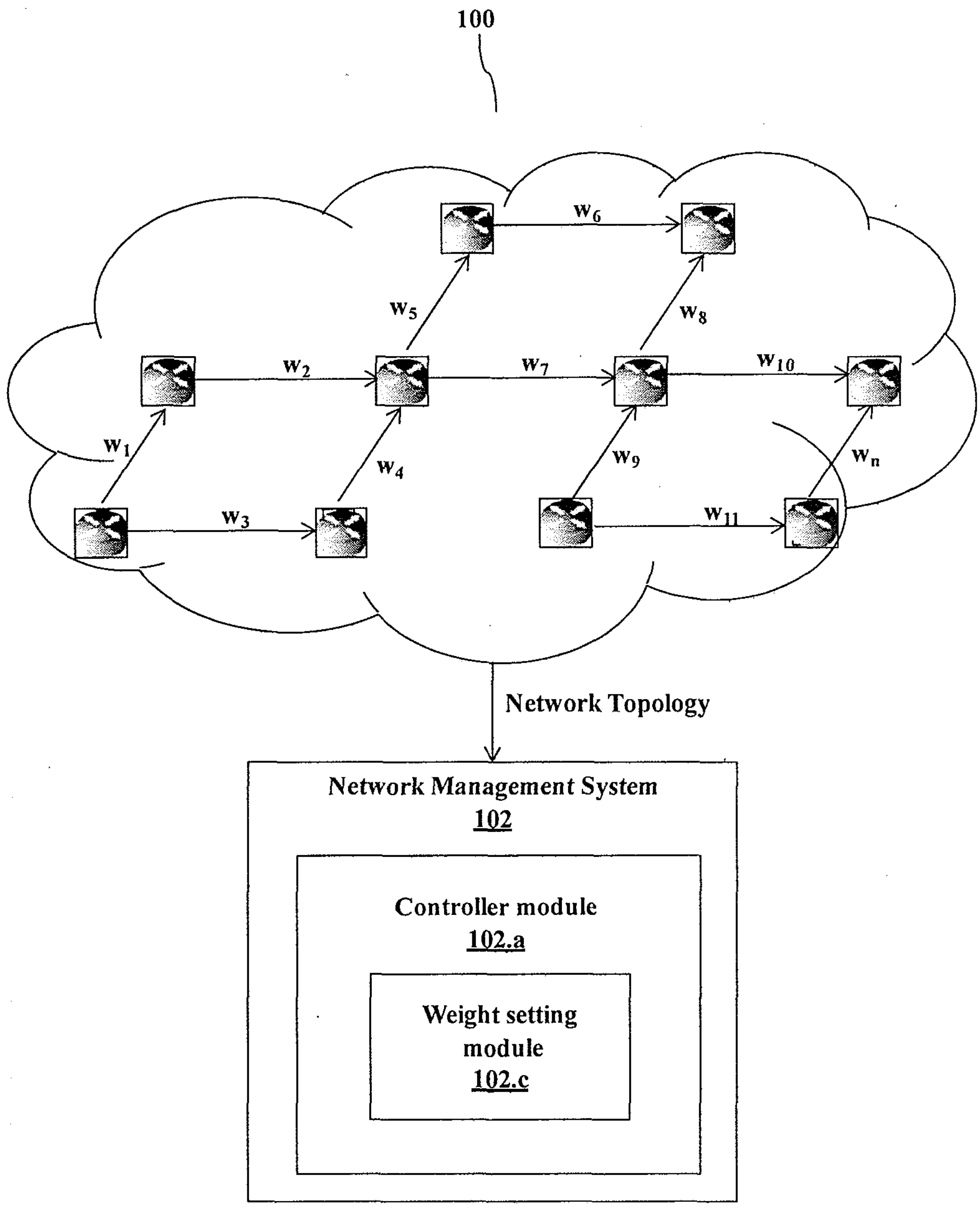


FIG. 1c

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Total No Of Sheets: 10

Application Number: Sheet No.: 4/10

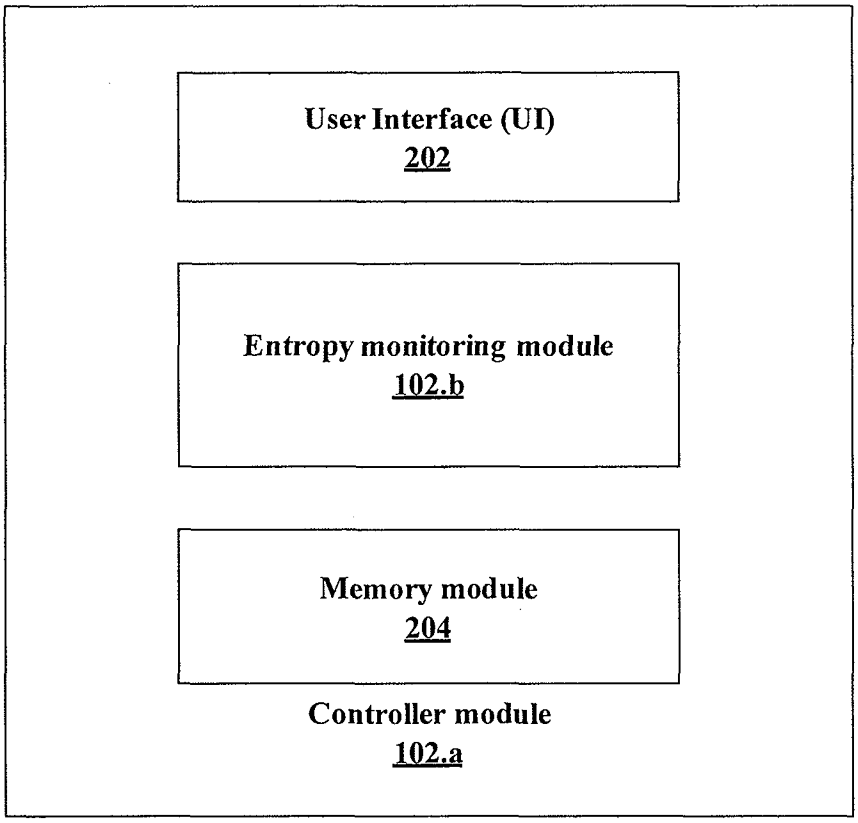


FIG. 2a

Arun Kishore Narasani
Patent Agent

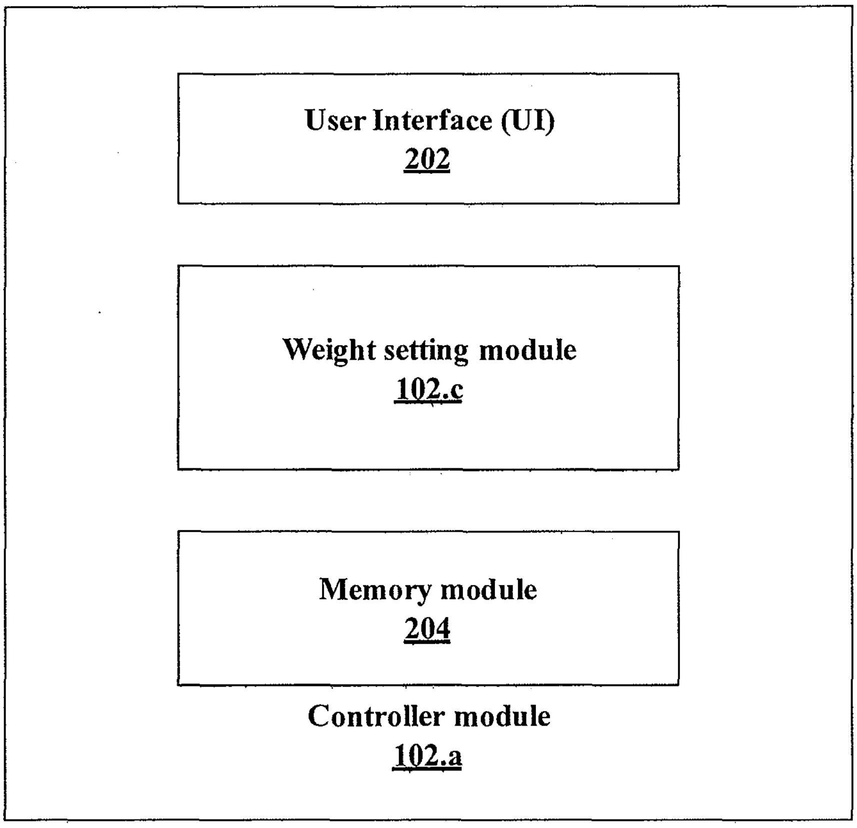


FIG. 2b

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Total No Of Sheets: 10

Application Number:

Sheet No.: 6/10

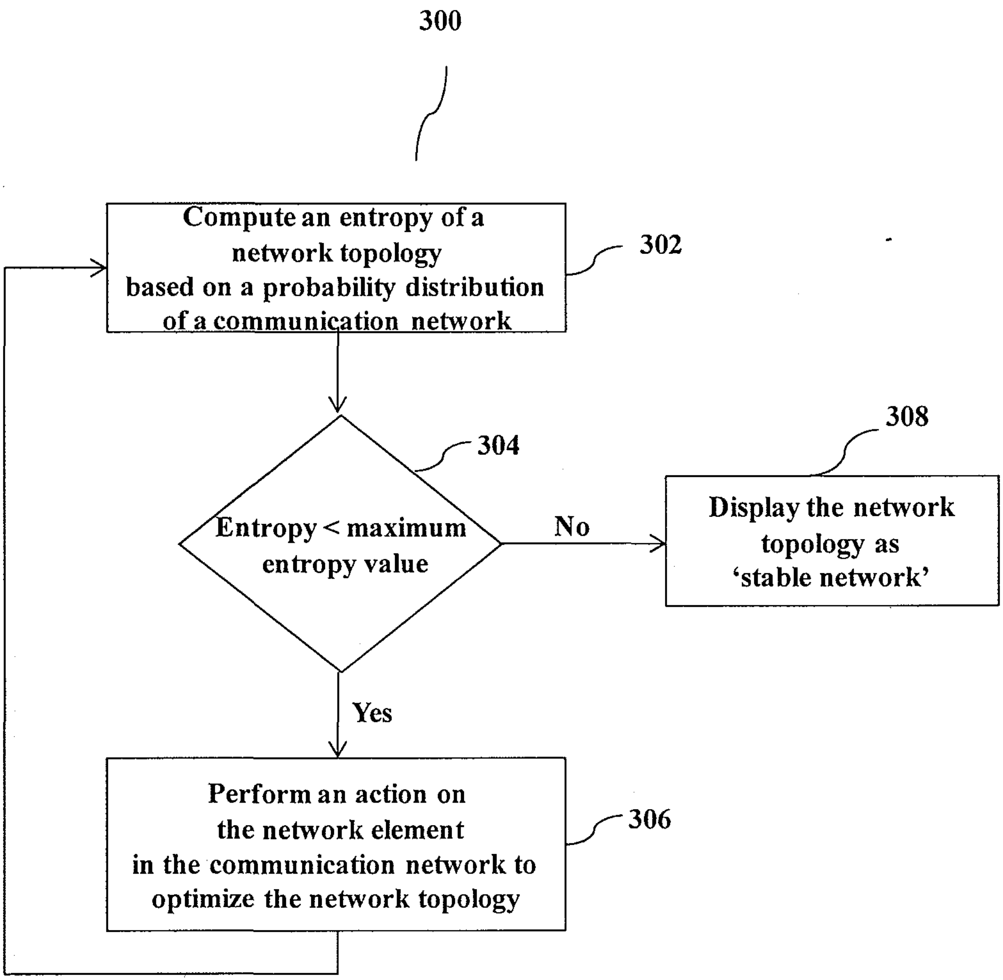


FIG. 3

Arun Kishore Narasani
Patent Agent

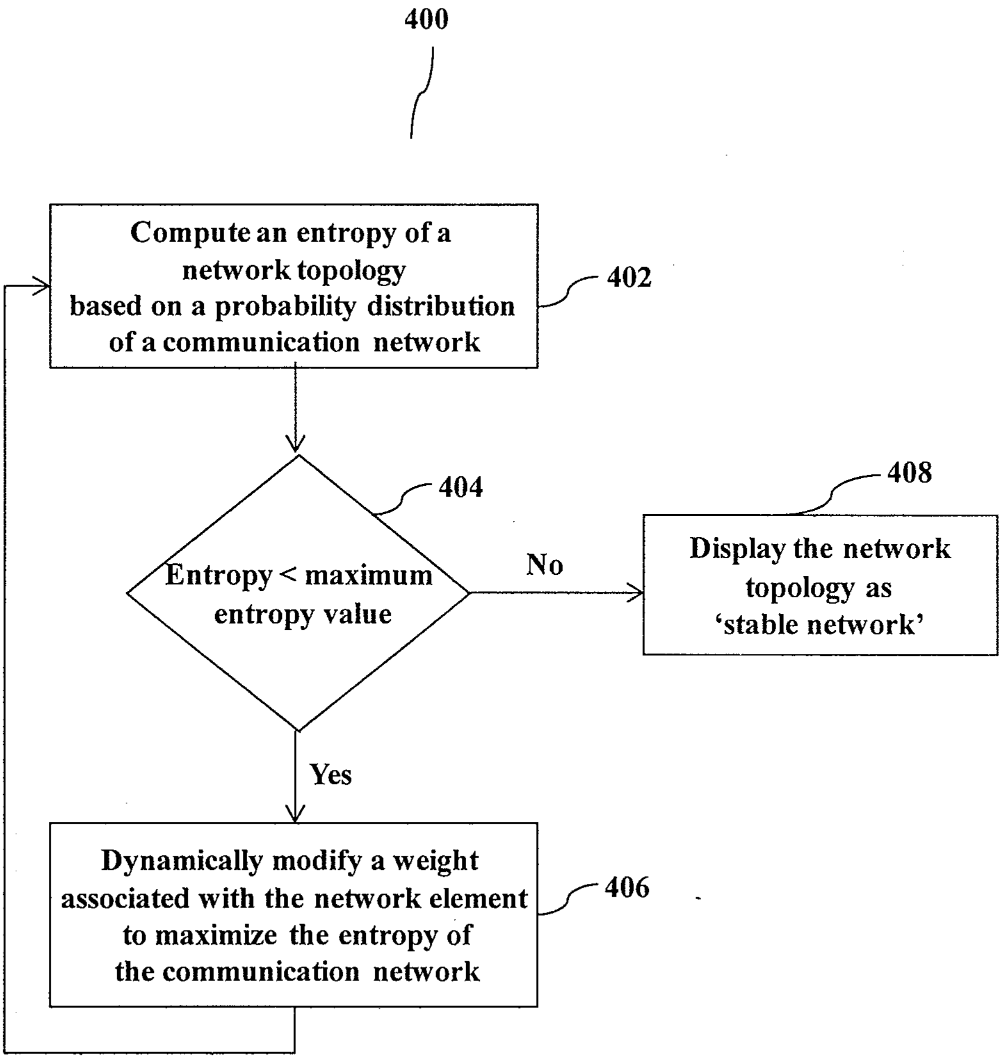


FIG. 4

Arun Kishore Narasani
Patent Agent

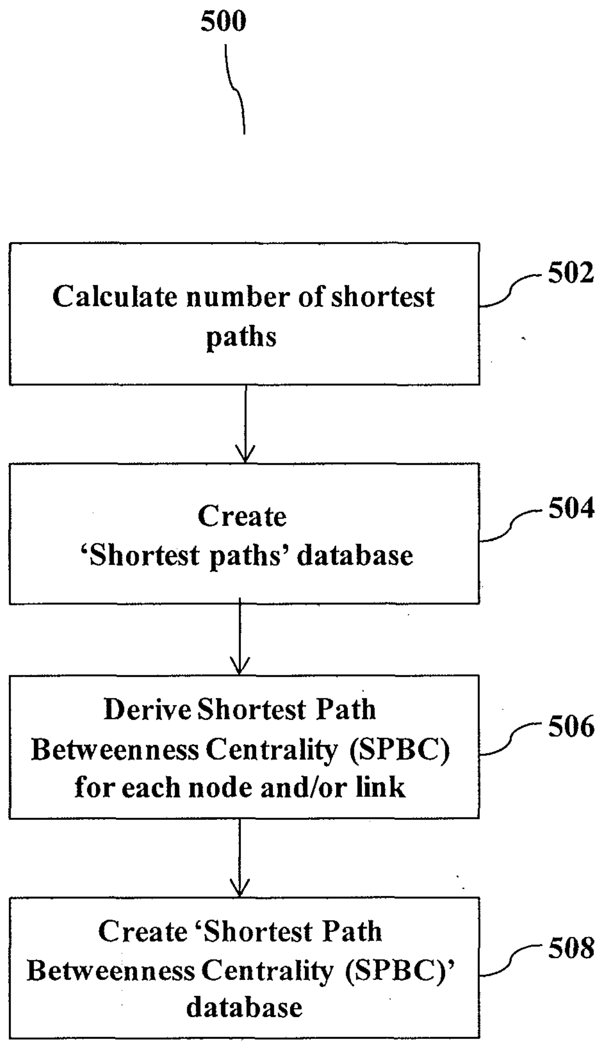


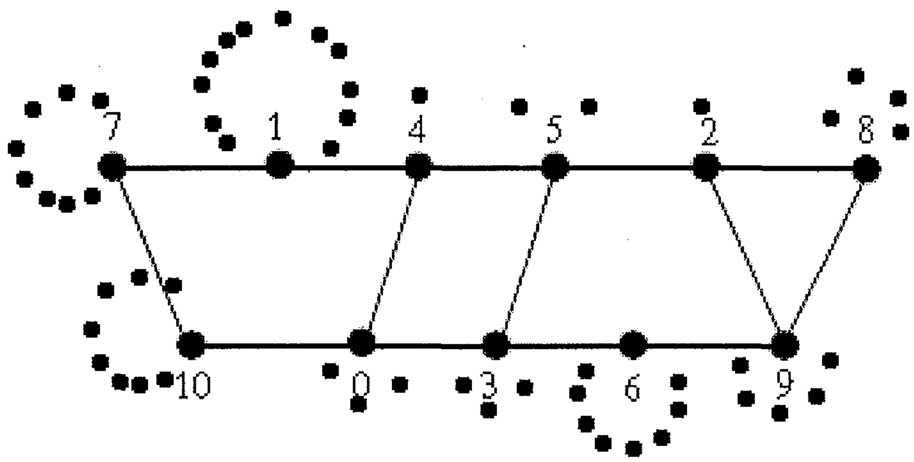
FIG. 5

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Application Number:

Total No Of Sheets: 10
Sheet No.: 9/10



Network 1

FIG. 6

Arun Kishore Narasani
Patent Agent

Name of the Applicant: Indian Institute of Technology Madras & HCL Technologies

Total No Of Sheets: 10

Application Number:

Sheet No.: 10/10

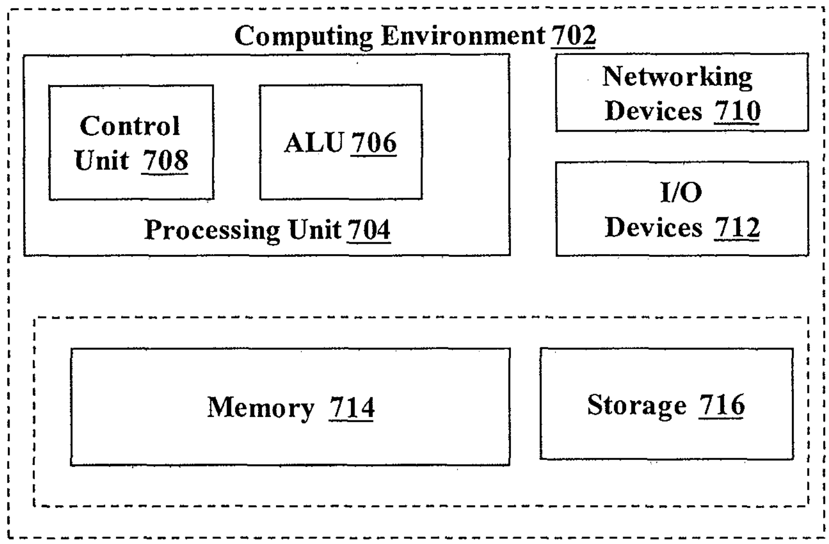


FIG. 7

Arun Kishore Narasani
Patent Agent

FIELD OF INVENTION

[001] The embodiments disclosed herein relate to a communication network and more particularly to optimize a network topology of a communication network. The present application is based on, and claims priority from, both the Indian Application Number, **5398/CHE/2013** filed on **22nd November, 2013** and **1514/CHE/2014** dated **21st March, 2014**, the disclosure of which is hereby incorporated by reference herein.

BACKGROUND OF INVENTION

[002] Network complexity provides a quantitative framework to measure the information content of a given topology of the network. Network analysis, on the other hand, provides formal ways of interpreting the network complexity measurement. In the context of network resiliency and connectivity, it is important to identify the critical nodes or links in a network. Intuitively, the significance of a node or link is determined based on its structural position with respect to the other network elements in the network topology.

[003] In the context of communication networks, the network topology is represented as a weighted directed graph. The weight associated with each link usually represents the cost of utilizing the link. Depending on the goals of network design, the weight could represent other characteristics of the link such as delay, capacity, estimated queuing times, and so on. Shortest Path First protocols such as Open Shortest Path First

(OSPF) or Intermediate System-Intermediate System (IS-IS) route traffic along shortest cost paths to the destination, where the path cost is determined by the link weights. The weights of the links can be set by the network operator to meet desired objectives. The inverse shortest path problem is to determine the link weights such that the given set of paths becomes shortest distance paths. The OSPF weight setting problem is an instance of inverse shortest path problem. In OSPF weight setting problem, the weights of the links is determined such that the paths between origin-destination (OD) pairs are made shortest distance paths that optimize the given traffic engineering objective function.

[004] Centrality distribution is a probability distribution associated with the network, and a new way of measuring the network topology complexity. A probability value is associated with each node or link based on some specific parameters of the node or link. For example, the parameters can be, but are not limited to the link capacity, link utilization, a number of flows, and node congestion. Intuitively, centrality distribution provides an insight into the importance of each node or link, and how they are ranked from the entire network context.

[005] In context of network resilience, a link or node failure can have varying impact on the entire network. It implies that the failure of a link with minimum centrality can imbalance the centrality distribution to a greater extent. When a link goes down, the shortest paths that pass through

the link are diverted through other links. This results in increase in the path distance between the OD pairs, and also routing re-convergence. In such cases, the original weight setting for optimal shortest path routing cannot be efficient for a network with link failures. Hence, there remains a need of a dynamic and robust system and method for optimizing the network topology by maximizing entropy of a select centrality distribution over the entire network in various conditions.

[006] The above information is presented as background information only to help the reader to understand the present invention. Applicants have made no determination and make no assertion as to whether any of the above might be applicable as Prior Art with regard to the present application.

OBJECT OF INVENTION

[007] The principal object of the embodiments herein is to provide a system and method for optimizing a network topology of a communication network.

[008] Another object of the embodiments herein is to provide a mechanism to compute an entropy of the network topology based on a probability distribution of the communication network.

[009] Another object of the embodiments herein is to provide a mechanism to determine whether the entropy of the network topology is less than a maximum entropy value.

[0010] Another object of the embodiments herein is to provide a mechanism to perform at least one action on at least one network element in the communication network to optimize the network topology.

[0011] Another object of the embodiments herein is to provide a mechanism to dynamically modify a weight associated with network elements to maximize the entropy of the communication network.

SUMMARY

[0012] Accordingly the embodiment herein provides method for optimizing a network topology of a communication network. The method includes computing an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on at least one parameter associated with each network element. Further, the method includes determining whether the entropy is less than a maximum entropy value. Further, the method includes performing at least one action on the at least one network element in the communication network to optimize the network topology in response to determining that the entropy is less than the maximum entropy value.

[0013] Accordingly the embodiments herein provide a method for optimizing a network topology of a communication network. The method includes computing an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on a number of shortest paths passing through the network element. Further, the method includes determining whether the entropy is less than a maximum entropy value. Furthermore, the method includes dynamically modifying a weight associated with the at least one network element to maximize the entropy of the communication network in response to determining that the entropy is less than the maximum entropy value.

[0014] Accordingly the embodiments herein provide a system for optimizing a network topology of a communication network. The system includes a controller module. The controller module is configured to compute an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on at least one parameter associated with each network element. Further, the controller module is configured to determine whether the entropy is less than a maximum entropy value. Furthermore, the controller module is configured to perform at least one action on the at least one network element in the communication network to optimize the network topology in response to determining that the entropy is less than the maximum entropy value.

[0015] Accordingly the embodiments herein provide a system for optimizing a network topology of a communication network. The system includes a controller module. The controller module is configured to compute an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on a number of shortest paths passing through the network element. Further, the controller module is configured to determine whether the entropy is less than a maximum entropy value. Furthermore, the controller module is configured to dynamically modify a weight associated with at least one network element to maximize the entropy of the communication network in response to determining that the entropy is less than the maximum entropy value.

[0016] Accordingly the embodiments herein provide a computer program product comprising computer executable program code recorded on a computer readable non-transitory storage medium. The computer executable program code when executed causes the actions including computing an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on at least one parameter associated with the network element. The actions include determining whether the entropy is less than a maximum entropy value. Further, the actions include performing at least one action on at least one network element in the communication network to optimize the network topology in response to determining that the entropy is less than the maximum entropy value.

[0017] Accordingly the embodiments herein provide a computer program product comprising computer executable program code recorded on a computer readable non-transitory storage medium. The computer executable program code when executed causes the actions including computing an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on a number of shortest paths passing through the network element. The actions include determining whether the entropy is less than a maximum entropy value. Further, the actions include dynamically modifying a weight associated with at least one the network element to maximize the entropy of the communication network in response to determining that the entropy is less than the maximum entropy value.

[0018] These and other aspects of the embodiments herein will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. It should be understood, however, that the following descriptions, while indicating preferred embodiments and numerous specific details thereof, are given by way of illustration and not of limitation. Many changes and modifications can be made within the scope of the embodiments herein without departing from the spirit thereof, and the embodiments herein include all such modifications.

BRIEF DESCRIPTION OF FIGURES

[0019] This invention is illustrated in the accompanying drawings, throughout which like reference letters indicate corresponding parts in the various figures. The embodiments herein will be better understood from the following description with reference to the drawings, in which:

[0020] FIG. 1aillustrates an example block diagram of a system to optimize a network topology of a communication network, in accordance to the embodiments as described herein;

[0021] FIG. 1b illustrates another example block diagram of a system to optimize the network topology of the communication network, in accordance to the embodiments as described herein;

[0022] FIG. 1c illustrates yet another example block diagram of a system to optimize the network topology of the communication network, in accordance to the embodiments as described herein;

[0023] FIG. 2a illustrates a block diagram that shows various components of a controller module, in accordance to the embodiments as described herein;

[0024] FIG. 2b illustrates another block diagram that shows various components of the controller module, in accordance to the embodiments as described herein;

[0025] FIG. 3 is a flow diagram illustrating a method for optimizing the network topology of the communication network, according to the embodiments as described herein;

[0026] FIG. 4 is a flow diagram illustrating a method for optimizing the network topology of the communication network, according to the embodiments as described herein;

[0027] FIG. 5 is a flow diagram illustrating a method for deriving a probability value associated with a network element of the communication network by using Shortest Path Betweenness Centrality (SPBC), according to the embodiments as described herein;

[0028] FIG. 6 illustrates an example network topology for which entropy is maximized, according to the embodiments as described herein; and

[0029] FIG. 7 illustrates a computing environment implementing the system and method for optimizing the network topology of the communication network, in accordance with an embodiment disclosed herein.

DETAILED DESCRIPTION OF INVENTION

[0030] The embodiments herein and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that are illustrated in the accompanying drawings and detailed in the following description. Descriptions of well-known components and processing techniques are omitted so as to not unnecessarily obscure the embodiments herein. Also, the various embodiments described herein are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments. The term "or" as used herein, refers to a non-exclusive or, unless otherwise indicated. The examples used herein are intended merely to facilitate an understanding of ways in which the embodiments herein can be practiced and to further enable those skilled in the art to practice the embodiments herein. Accordingly, the examples should not be construed as limiting the scope of the embodiments herein.

[0031] Prior to describing the present invention in detail, it is useful to provide definitions for key terms and concepts used herein. Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs.

[0032] Network Topology: A 'network topology' can be defined as a 'network design' or a 'network graph' which contains a set of nodes (or vertices) that are joined together in pairs by links (or edges).

[0033] Weighted network topology: A network topology is said to be 'weighted' if specific weights are assigned to each of its edges or links. Similarly, a network topology is said to be 'unweighted' if no weights are assigned to its edges or links.

[0034] Distance of path: For a weighted network topology, 'distance of path' between any two vertices or nodes can be given as sum of the weights associated with each of the edge or link that is present in the path selected.

[0035] Shortest path: 'Shortest path' between any two vertices or nodes in a network graph refers to the path in which the value of 'distance of path' is less when compared with the values of 'distance of path' of all other alternative paths that are present between the two selected nodes of the network topology. The shortest path in a network is also called as a 'geodesic'. Further, while routing traffic in a network, 'shortest paths' are chosen as they have very less 'distance of path'.

[0036] Centralized network: A network is said to be 'centralized' if at least one of its nodes or links is 'extremely critical' to the operation of the network. Extremely critical nodes or links can refer to the nodes or links which have got more number of shortest paths or more convergence/divergence paths through them. As these nodes or links are highly critical, they play very important role in the entire network. Due to these critical nodes, centralized networks are not preferred while implementing critical networks, because the whole network can be affected even if a single critical link or critical node fails.

[0037] Decentralized network: A network in which load is equally distributed among all the links/nodes or which does not have any critical nodes/links is known as a 'decentralized network'. Further, the decentralized networks are mostly preferred in real life network implementations, as the network performance cannot be affected even if a link or node fails.

[0038] Degree centrality: The 'degree centrality' refers to the count of number of links or edges incident upon a given node. Degree centrality further implies that the node with higher degree has more convergence or divergence of paths, and hence critical to the entire network.

[0039] Entropy function: An entropy function of a network measures how probability distribution of the given network is distributed. It quantifies the evenness or unevenness of the probability distribution present in the network topology. A network is said to be 'centralized' if its entropy value is low. Similarly, the network is said to be a 'decentralized network', if its entropy value is high.

[0040] Shannon Entropy: Let α be a random variable (indicating 'node' of a network topology) with a finite range between $a_1, \ldots\ldots a_n$. Let p_i be the probability of the event $\alpha = a_i$. Then, the Shannon entropy of the node α can be defined as

$$H_n(\alpha) = -\sum_{i=1}^{n} Pi \log pi \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The maximum Entropy value of H_n is:

$$\hat{H}n(\alpha) = -\sum_{i=1}^{n} 1/n \log(1/n) = \log n \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where, n represents total number of random variables.

[0041] When Shannon's entropy is extended to the edges or links present in a network topology, the entropy of the network can be defined as:

$$H_f(G) = -\sum_{(u,v)\in E} p(u,v) \log p(u,v) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

And the probability value of each link can be defined as

$$p(u,v) = \frac{f(u,v)}{\sum_{(x,y)\in E} f(x,y)} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where (u, v) represents a link in the network and (x, y) represents total number of links present in the network. The above 'entropy definitions (equation 1 and equation 3) can also be applied to any sub-graph that is induced by the set of paths that connect the given source-destination pair (say *(s, t)*).

[0042] Relative Entropy: The 'relative entropy' of a network topology can be defined as ratio of entropy value of the network with its maximum entropy value.

$$h_G = \frac{H(\Delta_G)}{\hat{H}(\Delta_G)} = \frac{H(\Delta_G)}{\log(\|E\|)}, 0 \leq h_G \leq 1 \text{.........................} (5)$$

Where, h_G represents relative entropy of the network topology

$H(\Delta_G)$ represents entropy of the network topology

$\hat{H}(\Delta_G)$ represents maximum entropy of the network topology.

The 'relative entropy' for an induced sub-graph (known as 'diversity index') can be given as

$$h_{st} = \frac{H(\Delta_{st})}{\log(\|E\|)}, \text{ and } 0 \leq h_{st} \leq 1 \text{.............................} (6)$$

Where, h_{st} represents the relative entropy of a sub-graph with source node s and destination node d.

[0043] Centrality distribution: 'Centrality distribution' of a given network defines whether a network is centralized or decentralized. This centrality distribution can be measured by using an 'information theoretic approach'. In this approach, a 'probability value' associated with the nodes or links of the network topology is determined based on its structural properties. Further, a suitable 'entropy' function (such as but not limited to 'Shannon's entropy function') can be applied on the determined probability values in order to measure the entropy of the network topology. Depending on the measured entropy value, the centrality distribution of the given network can be decided.

[0044] To determine the entropy of the given network, 'probability value of each node or link' is to be measured. The 'probability value of each node or link' refers to 'the degree to which each node or link acts as an intermediary in communication between every pair of nodes' present in the given network topology and it can be measured by using 'Betweenness centrality' of each node or link present in the network topology. More precisely, the 'Betweenness centrality' of a node or link in a network topology is determined by its occurrence in the shortest paths between the pairs of nodes associated with the network topology.

[0045] Shortest Path Betweenness Centrality: For a given node (say 'v') in a network topology, the Shortest Path Betweenness Centrality (SPBC) can be given as :

$$\eta_{\bullet,\bullet}(v) = \frac{\sigma_{\bullet,\bullet}(v)}{\sigma_{\bullet,\bullet}} \qquad (7)$$

where, '$\sigma_{\bullet,\bullet}$' represents the total number of shortest paths between every pair of source (s) and destination (t) nodes.

$\sigma_{\bullet,\bullet}(v)$ represents the number of geodesics or shortest paths that present between every pair of nodes (s, t) that pass through 'v'. Similarly, SPBC for a link (say (u, v)) with respect to a pair of nodes (say (s, t)) can be given as:

$$\eta_{\bullet,\bullet}(u,v) = \frac{\sigma_{\bullet,\bullet}(u,v)}{\sigma_{\bullet,\bullet}} \quad \text{.................................... (8)}$$

whereas, $\sigma_{\bullet,\bullet}$ represents total number of shortest paths between every pair of source (s) and destination (t) nodes.

$\sigma_{\bullet,\bullet}(u,v)$ represents the number of geodesics or shortest paths that are present between every pair of nodes *(s, t)* that pass through the *(u, v)*.

[0046] The embodiments herein provide a method and system for optimizing a network topology of a communication network. In an embodiment, an entropy of the network topology is computed based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on at least one parameter associated with each network element. Further, the system determines whether the entropy is less than a maximum entropy value. Further, the system performs at least one action on at least one network element in the communication network to optimize the network topology. The action is performed in response to determining that the entropy is less than the maximum entropy value.

[0047] In another embodiment, an entropy of the network topology is computed based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on a number of shortest paths passing through the network element. Further, the system determines whether the entropy is less than a maximum entropy value. Further, the system dynamically modifies a weight associated with at least one network element to maximize the entropy of the communication network. The weight associated with at least one network element is modified in response to determining that the entropy is less than the maximum entropy value.

[0048] Referring now to the drawings and more particularly to FIGS. 1 to 7 where similar reference characters denote corresponding features consistently throughout the figures, there are shown preferred embodiments.

[0049] Referring to FIG. 1a, an example block diagram of a system 100 to optimize a network topology of a communication network is illustrated, in accordance to the embodiments as described herein. Th e system 100includes a communication network. The system 100 optimizes a network topology of the communication network. Further, the system 100 includes a controller module 102.a. Furthermore, the controller module 102.a includes an entropy monitoring module 102.b. The topology of the network (i.e. network topology) can be fed as input to the controller module 102.a.

[0050] The controller module 102.a can be any Software Defined Network (SDN) controller or Network Management System (NMS) controller that provide(s) required network management functions to the network topology. The entropy monitoring module 102.b is a monitoring framework that provides assistance to the controller module 102.a in optimizing the network topology of the communication network. In an embodiment, the entropy monitoring module 102.b can be configured to perform at least one action on at least one network element in the communication network to optimize the network topology.

[0051] In another embodiment, a user can directly provide the network topology to be optimized, as input to the entropy monitoring module 102.b, using a suitable interface. Similar interface(s) can be used by the entropy monitoring module 102.b to communicate the entropy to the user.

[0052] FIG. 1b illustrates another example block diagram of a system 100 to optimize the network topology of the communication network, in accordance to the embodiments as described herein. The system 100 includes a Network management system (NMS) 102 connected to a communication network. The network topology of the communication network is optimized. Further, the NMS 102includes a controller module 102.a. Further, the controller module 102.a includes an entropy monitoring module 102.b. The topology of the network (i.e. network topology) can be fed as input to the controller module 102.a.

[0053] The controller module 102.a can be any Software Defined Network (SDN) controller or Network Management System (NMS) controller that provide(s) required network management functions to the network topology. The entropy monitoring module 102.b is a monitoring framework that provides assistance to the controller module 102.a in optimizing the network topology of the communication network. In an embodiment, the entropy monitoring module 102.b can be configured to perform at least one action on at least one network element in the communication network to optimize the network topology.

[0054] In another embodiment, a user can directly provide the network topology to be optimized, as input to the entropy monitoring module 102.b, using a suitable interface. Similar interface(s) can be used by the entropy monitoring module 102.b to communicate the entropy to the user.

[0055] FIG. 2aillustrates a block diagram that shows various components of the controller module 102.a, in accordance to the embodiments as described herein. In an embodiment, the controller module 102.acan be configured to include a User Interface (UI) 202, the entropy monitoring module 102.b, and a memory module 204.

[0056] The UI 202can provide the necessary input interface such as, but not limited to, touch screen, keyboard, and mouse to the user to input the network topology to the entropy monitoring module 102.b. The UI

202can also provide necessary output interfaces such as, but not limited to display screen in order to communicate the entropy to the user. The UI 202can also possess interface (s) to communicate and fetch required input (s) directly from the controller module 102.a or any other network component which can provide the input data.

[0057] In an embodiment, the UI 202can be configured to proactively fetch the input network topology from the user or from the controller module 102.a.

[0058] The entropy monitoring module 102.b can be configured to fetch the input network topology from the user through the UI 202. Further, the entropy monitoring module 102.bcan be configured to perform at least one action on at least one network element in the communication network to optimize the network topology. The action is performed on the network element in response to determining that an entropy of the network topology is less than a maximum entropy value. The entropy monitoring module 102.b can be configured to fetch the maximum entropy value from the memory module 202.

[0059] In order to optimize the network topology, the controller module 102.a can be configured to compute an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on at least one parameter associated with each network element. Further, the controller module 102.a. can be configured to determine whether the entropy is less than the maximum entropy value. The entropy monitoring module 102.b. can be configured to perform at least one action on at least one network element in the communication network to optimize the network topology in response to determining that the entropy is less than the maximum entropy value.

[0060] FIG. 2b illustrates another block diagram that shows various components of the controller module, in accordance to the embodiments as described herein. In an embodiment, the controller module 102.aincludes of the User Interface (UI) 202, the weight setting module 102.c, and the memory module 204.

[0061] The UI 202can provide the necessary input interface such as, but not limited to touch screen, keyboard, and mouse to the user to input the network topology to the weight setting module 102.c. The UI 202can also provide necessary output interfaces such as (but not limited to) display screen in order to communicate the entropy to the user. The UI 202can also possess interface (s) to communicate and fetch required input (s) directly from the controller module 102.a or any other network component which can provide the input data.

[0062] In an embodiment, the UI 202can be configured to proactively fetch the input network topology from the user or from the controller module 102.a. The input network topology can also include weights of the network elements of the network topology.

[0063] The weight setting module 102.c can be configured to fetch the network topology from the user through the UI 201. Further, weight setting module 102.c can be configured to determine weight settings that maximize the entropy of shortest path betweenness-centrality distribution. In an embodiment, the weight settings can be done considering an optimal shortest-path routing algorithm. In an embodiment, any suitable known algorithm can be used for setting the weights to the links present in the shortest paths. Furthermore, the weight setting module 102.c can be configured to determine weight settings such that failure of a link with minimum centrality cannot imbalance the centrality distribution to a greater extent.

[0064] In order to optimize the network topology, the controller module 102.a can be configured to compute an entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The probability value is derived based on a number of shortest paths passing through the network element. The controller module 102.a can be configured to determine whether the entropy is less than a maximum entropy value. The weight setting module 102.c can be configured to dynamically modify the weight associated with at least one network element to maximize the entropy of the communication network in response to determining that the entropy is less than the maximum entropy value.

[0065] FIG.3isa flow diagram illustrating a method 300 for optimizing the network topology of the communication network, according to the embodiments as described herein. A decentralized network topology is considered to be a preferred network topology as the criticality is more or less equally distributed between all nodes or links present in the network. This makes the network more resilient in the face of node or link failures. Hence, before implementing the network topology, the centrality distribution of the communication network should be measured. The centrality distribution of the communication network can be represented by computing the entropy of the network topology.

[0066] At step 302, the method 300 includes computing the entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The network element can be a node, or a link joining two nodes. In one embodiment, the method 300 allows the controller module 102.a. to derive the probability value based on at least one parameter associated with each network element. The parameter includes at least one of a number of shortest paths, a number of packets, a number of sessions, link utilization data, link capacity data, link residual capacity data, a number of flows, and congestion.

[0067] In an embodiment, the probability value can be derived by determining a Shortest Path Betweenness Centrality (SPBC) for each network element based on a number of shortest paths passing through the network element. The probability value for each network element is derived based on the SPBC for each network element.

[0068] Further, at step 304, the method 300 includes determining whether the entropy is less than a maximum entropy value. The maximum entropy value for the network topology can be one. Further, at step 306, the method 300 includes performing at least one action on at least one network element in the communication network to optimize the network topology, when the entropy is less than the maximum entropy value.

[0069] In an embodiment, performing the at least one action can include dynamically modifying a weight associated with at least one network element to maximize the entropy of the communication network. The weight associated with the network element is dynamically modified based on at least one parameter of the network element. The method 300 allows the entropy monitoring module 102.b to dynamically modify the weight by increasing the weight associated with the network element to a degree to maximize the entropy of the communication network. The degree of increasing the weight is derived using an optimal shortest-path routing technique.

[0070] In another embodiment, the weight can be dynamically modified by decreasing the weight associated with the network element to a degree to maximize the entropy. The degree is derived using an optimal shortest-path routing technique. In another embodiment, the at least one action can be communicating to the user to take a corrective action to maximize the entropy of the network topology.

[0071] In another embodiment, performing the at least one action can include dynamically modifying a number of the network elements in the network topology to optimize the network topology. In another embodiment, performing the at least one action can include modifying a direction of communication between the network elements in the network topology to optimize the network topology.

[0072] Further, at step 308, the method 300 includes displaying the network topology as 'stable network', when the entropy of the network topology is not less than the maximum entropy value. After step 306, the method 300 repeats from step 302 and continues to perform at least one action on at least one network element in the communication network until the entropy of the network topology is optimized.

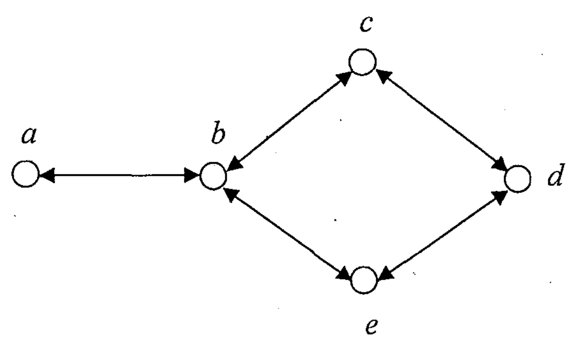
[0073] The various actions, acts, blocks, steps, and the like in the method 300 can be performed in the order presented, in a different order or simultaneously. Further, in some embodiments, some actions, acts, blocks, steps, and the like can be omitted, added, modified, skipped, and the like without departing from the scope of the invention.

[0074] FIG. 4 is a flow diagram illustrating a method 400 for optimizing the network topology of the communication network, according to the embodiments as described herein. At s tep 402, the method 40 0 includes computing the entropy of the network topology based on a probability distribution of the communication network. The probability distribution is derived based on a probability value of each network element. The network element can be a node, or a link joining two nodes. In one embodiment, the method 400 allows the controller module 102.a. to derive the probability value of each network element based on a number of shortest paths passing through the network element.

[0075] In an embodiment, the probability value for each node or link can be derived by applying a technique of SPBC to each node or link present in the given network topology. In this technique, initially shortest paths that are associated with each of the node or link present in the network are identified. Later, based on the shortest paths, the probability value of each node or link present in the network is derived.

[0076] FIG. 5 is a flow diagram illustrating a method500for deriving the probability value associated with the network element of the communication network by using Shortest Path Betweenness Centrality

(SPBC), according to the embodiments as described herein. The probability values of each node or link of the communication network can be derived by applying SPBC to each node and link present in the network topology. At step 502, the method500includes computing the number of shortest paths passing through each node or link. For example, let us consider the following network topology with each bidirectional edges and let us also assume that each edge has 'unit weight'. For routing purpose, let us consider that the network topology has no self-loops, and paths connecting any pair of vertices or nodes are loop-free.



Network 2

[0077] In the above network topology, let a, b, c, d, e represents the nodes while (a, b), (b, c), (c, d), (b, e), (e, d) represent links of the network topology. Now, the 'shortest paths' of the above network topology can be calculated as follows:

[0078] Let us consider a node (say 'a'). Further, the number of shortest paths that this node has with all other nodes in the given network can be computed. This can be done as follows: First, consider node b. The number of paths that are present in between a, and can be calculated. Later, the shortest path among them can be identified by calculating 'distance of path' for each of the path identified. Here, there is only one path that exists between the nodes a, and b. So the number of shortest paths can be considered as 1.

[0079] The same procedure can be repeated for all the other nodes that are present in the network topology. Further, consider node c. There is again only one path which connects the nodes a, and c. The path connecting a, and c can be shortest path and the number of shortest paths between them is 1. Next, let us consider node d. There are two paths (i.e., a-b-c, a-b-e) which connects the nodes a, and d. The shortest path among these two paths can be found by calculating 'distance of path' between them. As the each link in the network has 'unit weight', the distance of path for a-b-c is 2 (i.e., 1+1) and for a-b-e the distance of path is 2 (i.e., 1+1). So, the two paths which exists between a, and d are short. Hence, in this case the number of shortest paths is 2. Next, if we consider node e, there exists only one path between the nodes a, and e, which can be considered as the shortest path.

[0080] Further, the number of shortest paths that exists from other nodes to all other nodes present in the network topology can be computed by using the similar procedure. At step 504, the method 500 includes creating a 'Shortest paths database' which contains details of calculated number of shortest paths. In one embodiment, the method 500 allows the controller module 102.a. to store the 'Shortest paths database' in the memory module 204. The following table shows an example database which shows the number of shortest paths that exists from each node to all other nodes of the above network topology.

Nodes	*A*	*b*	*c*	*d*	*e*
a	-	1	1	2	1
b	1	-	1	2	1
c	1	1	-	1	2
d	2	2	1	-	1
e	1	1	2	1	-

Table 1

[0081] After creating the 'Shortest Paths database' in the memory module 204, the total number of shortest paths (i.e., $\sigma_{\bullet,\bullet}$) that exists in the given network topology can be calculated. This can be calculated by adding all the shortest paths from each node to all other nodes present in the network topology. For example, the total number of shortest paths for the above network topology (Network 2) is 26 (5+5+5+6+5).

i.e., $\sigma_{\bullet,\bullet}=26$

[0082] Further, at step 506, the method 500 includes deriving the SPBC of each node and or link present in the network topology. The SPBC value for each node present in the network topology can be calculated by using equation (7) and the SPBC value for each link in the network topology can be calculated by using equation (8).

[0083] For example, if we consider node '*a*' of *Network 2,* its SPBC value can be calculated as follows:

Initially, calculate the number of shortest paths between every pair of nodes that passes through the node *a* ($\sigma_{\bullet,\bullet}(v)$). These can be listed as follows:

1) *a-b*
2) *a-b-c*
3) *a-b-c-d*
4) *a-b-e-d*
5) *a-b-e*
6) *b-a*
7) *c-b-a*
8) *d-c-b-a*
9) *d-e-b-a*
10) *e-b-a*

[0084] So, the number of shortest paths between every pair of nodes that passes through node 'a' is 10.

The SPBC value for the node *a* is given as (using equation 5)

$$\eta_{\bullet,\bullet}(v) = \frac{\sigma_{\bullet,\bullet}(v)}{\sigma_{\bullet,\bullet}} = \frac{10}{26}$$

[0085] Similarly, the SPBC values can be calculated for each other nodes or links present in the network topology.

[0086] Further, at step 508, the method 500 includes creating a 'SPBC database' of the network topology. The method 500 allows the controller module 102.a. to store the SPBC database in the memory module 204.

[0087] The following table shows an example of the SPBC database for each node present in the above network (i.e., Network 2).

Nodes	*a*	*b*	*c*	*d*	*e*
$\eta_{\bullet,\bullet}(v)$	10/26	20/26	14/26	12/26	14/26

Table 2

[0088] Similarly, the following table shows an example database for SPBC each link present in the above network (i.e. *Network 2*).

Links	*(a, b)*	*(b, c)*	*(c, d)*	*(d, e)*	*(e, b)*
$\eta_{\bullet,\bullet}(u,v)$	10/26	10/26	8/26	8/26	10/26

Table 3

Thus, the probability values can be calculated for all the nodes or links present in the given network topology using the concept of SPBC.

[0089] The various actions, acts, blocks, steps, and the like in the method 500 can be performed in the order presented, in a different order or simultaneously. Further, in some embodiments, some actions, acts, blocks, steps, and the like can be omitted, added, modified, skipped, and the like without departing from the scope of the invention.

[0090] FIG. 6 illustrates an example network topology (Network 1) for which entropy is maximized, according to the embodiments as described herein. Let us consider that the example network topology is a Tactical Communication System (TCS) network which consists of communication nodes connected in a partial mesh topology over an operational area. For routing purpose, let us consider that the network topology has no self-loops, and paths connecting any pair of vertices or nodes are loop-free. The nodes in the network can be classified as trunk nodes and access nodes. The trunk nodes are connected by high capacity point-to-point radio links, and form backbone of the network. The access nodes are connected to the trunk nodes, and are capable of providing connectivity to several users in the network.

[0091] Let 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 be the nodes of the example network topology. The number of shortest paths across each node and the probability P (u, v) value for each link present in the above network topology are as illustrated in Table 4below:

(u, v)	$\sigma_{\bullet,\bullet}(u,v)$	***P(u, v)***	***(u, v)***	$\sigma_{\bullet,\bullet}(u,v)$	***P(u, v)***
(0, 3)	32	0.041209	(0, 4)	24	0.032967
(0, 10)	50	0.068681	(1, 4)	15	0.020604
(1, 7)	18	0.024725	(2, 5)	31	0.042582

(2, 8)	18	0.024725	(2, 9)	21	0.028846
(3, 0)	42	0.057692	(3, 5)	21	0.028846
(3, 6)	34	0.046703	(4, 0)	27	0.037088
(4, 1)	63	0.086538	(4, 5)	27	0.037088
(5, 2)	31	0.042582	(5, 3)	30	0.041209
(5, 4)	46	0.063187	(6, 3)	21	0.028846
(6, 9)	22	0.030220	(7, 1)	18	0.024725
(7, 10)	15	0.020604	(8, 2)	12	0.016484
(8, 9)	17	0.023352	(9, 2)	9	0.012363
(9, 6)	34	0.046703	(9, 8)	13	0.017857
(10, 0)	17	0.023352	(10, 7)	22	0.030220

Table 4

[0092] The following table shows the total number of access nodes that are using the trunk nodes of the above network topology.

Trunk nodes	Number of Access nodes using corresponding trunk node
0	3
1	12

2	1
3	3
4	1
5	2
6	8
7	8
8	4
9	5
10	8

Table 5

[0093] Now, the entropy for the example network topology is calculated by substituting the probability values in equation (3) which is:

$$H_{28}(\Delta_G) = 4.662206$$

[0094] In another embodiment, the entropy can be a relative entropy. The relative entropy of the example network topology can be computed by substituting the values of 'entropy' and 'maximum entropy value' of the example network topology in equation (5). Further, the maximum entropy value of the network can be calculated by using equation (2). The maximum entropy value of any communication network is '1'. In an embodiment, the maximum entropy value is pre-configured in the memory module 203.

[0095] If we consider the Network 1, the 'relative entropy' can be calculated as follows:

The 'maximum entropy' value of *Network 1* is given by

$$\hat{H}(\Delta_G) = \log(\|E\|) = \log(28) = 4.8073549221$$

Now, the relative entropy of the *network 1* is calculated as

$$h_G = \frac{H(\Delta_G)}{\hat{H}(\Delta_G)} = \frac{H(\Delta_G)}{\log(\|E\|)} = \frac{4.662206}{4.807354} = 0.974946$$

[0096] The network topologies which are having higher relative entropy are considered to be highly reliable and more decentralized. Hence, to check whether the communication network is centralized or decentralized, the 'relative entropy' of the network should be compared with the maximum entropy value.

[0097] Referring to the FIG. 4, at step 404, the method 400includes determining whether the entropy is less than the maximum entropy value. Further, at step 408, the method 400 includes displaying a message indicating that the network topology is 'stable network', when the entropy is equal to or less than the maximum entropy value.

[0098] If we consider the above example (Network 1), since the measured 'relative entropy' value (0.969807) is much closer to the maximum entropy value (i.e., 1), the above network can be considered as partially decentralized network but not a perfect decentralized network.

[0099] Further, at step 406, the method 400 includes dynamically modifying a weight associated with at least one the network element to maximize the entropy of the communication network, when the entropy is less than the maximum entropy value. In one embodiment, the method 400 allows the weight setting module 202.c to dynamically modify the weight by increasing the weight associated with the network element to a degree to maximize the entropy of the communication network. The degree of increasing the weight is derived using an optimal shortest-path routing technique.

[00100] In another embodiment, the weight can be dynamically modified by decreasing the weight associated with the network element to a degree to maximize the entropy. The degree is derived using an optimal shortest-path routing technique.

[00101] In an embodiment, the optimal shortest-path routing technique can be a Centrality Entropy Maximization' (CEM) algorithm. The below algorithm describing an example way for obtaining the CEM in the NMS is as follows:

1: Input $G = (V, E, W_0)$ {Network with initial weight Vector}

2: Output W {New Weight Assignment Vector}

3: $P_0 \leftarrow$ Initial Entropy of SPBC

4: $P \leftarrow \{\}$

5: for each link $(u, v) \in E$ do

6: $S \leftarrow$ Set of OD pairs that have SPs pass through (u, v)

7: $k \leftarrow |S|$

8: D ←Path distance of k OD pairs

9: w← $w_{u,v}$

10: $w_{u,v}$ ← ∞

11: D! ← Path distance of k OD pairs

12: di ← D!i - Di i = 1, 2,..., k

13: δ1 ← first smallest di

14: δ2 ← second smallest di

15: $w_{u,v}$ ← w + (δ1+ δ1)/2 {Increment the weight}

16: P ← P U H (ΔG) {Neighborhood of P_0}

17: end for

18: P_{max} ← Max (P) {Improved Entropy over P_0}

19: if no improvement over P_0 then return W

20: else W ← modified weight vector corresponding to P_{max}

21: P_0← P_{max}

22: repeat from step 4

Max $H(\Delta G) = -\sum_{(u,v)\in E} p(u,v)\log p(u,v)$

Subject to constraints $p(u,v) \geq 0$, $\sum_{(u,v)\in E} p(u,v) = 1$, and $w_{u,v} \geq 0$

[00102] In an embodiment, the optimal shortest-path routing algorithm (CEM) is applied to find weight settings that maximize the entropy of the network topology. The entropy of the network topology can be termed as the entropy of the shortest path betweenness-centrality

distribution. The working details of the CEM algorithm implemented to find the weight settings are described below:

[00103] In an embodiment, let W_0 denote the initial weight assignment vector. Let P_0 denote the value of entropy in equation (5) obtained from the weight assignment vector W_0. Here neighborhood of P_0 is defined as {P}, where {P} is a set of points with cardinality |E|. Each member in {P} is a value of entropy function in the above equation such that only a minimum number of paths are changed with respect to P_0 as a consequence of an increase in a single weight of W_0. From the neighborhood solution set {P}, the maximum value is chosen called P_{max} such that $P_{max} > P_0$, and also $P_{max} \geq P_i$, $P_i \in P$. Let W denote the corresponding weight assignment vector. Now, W_0 is set to W, and P_0 is set to P_{max}. This procedure is repeated with the newly discovered neighborhood until such P_{max} is found.

[00104] In an embodiment, consider a link (u, v) $\in$ E. Let $\sigma 0_{\bullet,\bullet}(u,v)$ represent the number of shortest paths that pass through the link (u, v) for the weight assignment vector W_0. The idea is to divert a few paths from this link by increasing the weight $w_{u,v}$ appropriately. This will in turn affect the value of $\sigma_{\bullet,\bullet}(u,v)$, and in turn the entropy.

[00105] Let k denote the number of origin-destination (OD) pairs that have shortest paths pass through (u, v). Let Si denote the set of shortest

paths between i^{th} OD pair that pass through (u, v), $1 \leq i \leq k$. The choice of incremental value to $w_{u,v}$ is determined such that the number of shortest paths diverted from (u, v) is minimum. The link (u, v) is removed by setting $w_{u,v}$ to ∞. The shortest paths are recomputed for these k OD pairs. In effect, all paths that go through the link (u, v) are diverted. Let d_i denote the difference between the path distances after and before removing the link (u, v) for an OD pair i, $1 \leq i \leq k$. This difference list is sorted, and the following observations are made on the sorted list.

[00106] If any d_i is 0, it means that i_{th} OD pair has at least one shortest path of same distance that does not pass through the link (u, v). In this case, increment the value of $w_{u,v}$ by the mid-point between 0 and the first smallest non-zero difference, and re-compute the shortest paths between these OD pairs. This eventually reduces the number of shortest paths connecting an OD pair, without modifying the distances of other OD pairs that have shortest paths passing through the link (u, v). In this case, the path distance of any OD pair is not affected; however, the number of shortest paths is minimized between the OD pairs that have zero difference in their path distance after and before removing the link.

[00107] If there is no zero difference, then increment the value of $w_{u,v}$ by the mid-point between the first smallest difference and the second smallest difference. This can be used to ensure only those paths are diverted that have the smallest difference in their distance after and before removing

the link. In this case, some OD pairs will have increase in their path distance. So, while distributing the paths between OD pairs to improve the entropy, the path lengths of some OD pairs are also increased.

[00108] This condition conflicts with the design goal of reducing the hop count or path distance between OD pairs. These design goals can be balanced, by having an additional constraint such that no path is diverted if the difference is beyond a specified limit. The maximum number of links that can be modified as a constraint can also be specified.

[00109] More precisely, the appropriate increase in the weight $w_{u,v}$ maximizes the entropy of SPBC distribution, and minimizes the number of shortest paths throughout the network. This approach can also be extended for achieving maximum entropy of shortest path betweenness-centrality of nodes. It is one of the design goals to minimize the number of paths used for routing under certain design considerations. Maximizing the entropy balances two important goals of network design. At one end, it tries to minimize the number of shortest paths in the network, which in turn helps in resolving Braess's paradox, quick routing convergence and the overhead of reassembly when multiple paths are chosen for routing. At the other end, the shortest paths are distributed between the links as even as possible.

[00110] Further, at step 408, the method 400 includes displaying the network topology as 'stable network', when the entropy of the network topology is not less than the maximum entropy value. After step 406, the

method 400 repeats from step 402 and continues to dynamically modify the weights of at least one network element until the entropy of the network topology is maximized.

[00111] The various actions, acts, blocks, steps, and the like in the method 400 can be performed in the order presented, in a different order or simultaneously. Further, in some embodiments, some actions, acts, blocks, steps, and the like can be omitted, added, modified, skipped, and the like without departing from the scope of the invention.

[00112] Referring again to the FIG. 6, for the depicted network topology, let us assume that all links have equal capacity, and each edge has unit weight. The progress of the algorithm is presented in Table 6. Column link specifies the link for which the weight has been modified. Column weight specifies the actual weight that has been set. Column entropy specifies the entropy of SPBC distribution, and it can be seen that it steadily increases at each step.

[00113] As shown in the Table 6, the number of shortest paths throughout the network has come down. The number of shortest paths that pass through each link before and after CEM is given in Table 7. Note that the highly centralized links (0, 10) and (4, 1) are decentralized to the extent possible. This results in a broader SPBC probability distribution, which in turn results in increased entropy value. Intuitively, the role of each link is made as uniform as possible subject to topology constraints.

Link	Weight	Entropy	$\sigma_{\bullet,\bullet}$
Initial	-	4.662206	404
(4,1) ↑	1.5	4.688495	374
(3,0) ↑	1.25	4.705924	352
(4,0) ↑	1.125	4.712875	342
(0,10) ↑	1.125	4.716082	339
(2,9) ↑	1.0625	4.716538	336
(1,4) ↑	1.9375	4.717794	336

Table 6

Link	Before	After	Link	Before	After
(0,3)	30	23	(0,4)	24	12
(0,10)	50	35	(1,4)	15	12
(1,7)	18	14	(10,7)	22	18
(2,5)	31	29	(2,8)	18	18
(2,9)	21	18	(3,0)	42	19
(3,5)	21	19	(3,6)	34	32
(4,0)	27	16	(4,1)	63	31
(4,5)	27	26	(5,2)	31	30
(5,3)	30	20	(5,4)	46	31

(6,3)	21	18	(6,9)	22	22
(7,1)	18	17	(7,10)	15	17
(8,2)	12	12	(8,9)	17	14
(9,2)	9	7	(9,6)	34	29
(9,8)	13	13	(10,6)	17	19

Table 7

[00114] Another example is described which includes 50 nodes and 148 links. Assume that each edge has unit weight. The summarized results are as follows. The initial entropy of SPBC distribution is 6.937850. After running the algorithm, the maximum entropy is 6.971319. The total number of initial shortest paths throughout the network is 3466, and after maximizing the entropy, the number shortest paths become 2811. The algorithm modified weights of 20 links.

[00115] In an embodiment, OSPF single link failure condition is described. The relative entropy of SPBC can be calculated as equation:

$$h_G = \frac{H(\Delta_G)}{\log(\|E\|)}, \text{ and } 0 \le h_G \le 1.$$

[00116] In an embodiment, the principle of maximum entropy can be applied to find the single weight setting for both cases (i.e., path distance between the OD pairs and routing re-convergence). Let $H_n(\Delta_G)$ represent the entropy of SPBC, where n is the number of links in G. Let h_G represent the relative entropy of SPBC as specified in the above equation. Wi denotes

the weight vector assignment at i^{th} iteration. Consider a link (x, y) ∈ E. Let G^{xy} represent the network without the link (x, y). It is equivalent to setting Wxy to infinity in G. The remaining weights of the links are kept intact. Now, the entropy of SPBC of G^{xy} can be computed. Let Hm (ΔGxy) represent the entropy of SPBC, where m denotes the number of l_i^{nk}s in Gxy. Let h_G^{xy} represent the relative entropy of SPBC. The goal is to find the appropriate weights assignment vector that balances and improves both h_G and h_G^{xy}.

[00117] The entropy function can be applied on the relative entropy values and attempt to improve it. The h_G and h_G^{xy} can be normalized such that $\hat{h}_G = \frac{h_G}{h_G + h_G xy}$, $\hat{h}_G xy = \frac{h_G xy}{h_G + h_G xy}$. The relative entropies { $\hat{h}_G, \hat{h}_G xy$ } form a probability distribution. Now, the objective function can be maximized.

$$z = -\hat{h}_G \log_e \hat{h}_G - \hat{h}_G xy \log \hat{h}_G xy$$

[00118] Maximizing the above equation results in a uniform distribution of relative entropies. More precisely, the weights of the links are chosen such a way that the centrality distributions of both before and after link failure are as close as possible. Here the relative entropy of normal network can come down significantly to match the relative entropy of failure case. To avoid this condition, one or more condition is added

such that the relative entropy of normal network should not come down below a given threshold value. This condition will ensure the relative entropy of normal network is always maintained greater than the threshold value, while the entropy of the failure case is maximized at each iteration.

[00119] In an embodiment, the procedure for maximizing the entropy of SPBC for single link failure on the Abilene network shown in the above network. Consider the failure of link (4, 1). The progress of the algorithm is shown in Table 8. In this example, the condition is added that the relative entropy of normal network should not be less than the initial relative entropy value of 0.969807. The number of shortest paths that pass through each link for the three cases are given in Table 9. Column A represents the number of shortest paths that pass through each link at the initial stage. Column B represents the number of shortest paths that pass through each link under normal network condition when the relative entropy h_G reaches the maximum value of 0.969887. Column C represents the number of shortest paths that pass through each link under a link (4, 1) failure condition when the relative entropy h_{G41} reaches 0.848751. Though this approach balances both entropies, it also produces some undesired effect on select links. For example, the link (0, 10) will have 60 shortest paths passing through it when the link (4, 1) goes down. This is higher than the initial and normal condition of the network. This also illustrates the point that failure of a link will make some other link highly central. This

justifies that the criticality of a link is also determined based on the back up quality of the link or alternate path centrality of the link.

Link	Weight	h_G	$h_{G4,1}$	$\sigma_{\bullet,\bullet}$
Initial	-	0.969807	0.812987	404
(5,4) ↑	1.5	0.974439	0.825250	370
(4,0) ↑	1.5	0.971615	0.833669	356
(7,1) ↑	1.5	0.970589	0.838221	351
(0,10) ↑	1.75	0.972575	0.844789	347
(4,0) ↑	2.5	0.970564	0.848350	339
(4,5) ↑	1.25	0.970161	0.848722	338
(10,7) ↑	1.125	0.969888	0.848751	338
(0,3) ↑	1.125	0.969887	0.848751	338

Table 8

Link	A	B	C	Link	A	B	C
(0,3)	30	20	25	(0,4)	24	18	4
(0,10)	50	23	60	(1,4)	15	12	12
(1,7)	18	22	9	(10,7)	22	13	38
(2,5)	31	31	27	(2,8)	18	16	20
(2,9)	21	20	23	(3,0)	42	34	56

(3,5)	21	14	16	(3,6)	34	30	38
(4,0)	27	2	9	(4,1)	63	51	0
(4,5)	27	18	29	(5,2)	31	26	33
(5,3)	30	27	46	(5,4)	46	23	4
(6,3)	21	18	27	(6,9)	22	22	24
(7,1)	18	14	23	(7,10)	15	19	17
(8,2)	12	12	12	(8,9)	17	16	21
(9,2)	9	9	5	(9,6)	34	31	40
(9,8)	13	13	14	(10,6)	17	19	19

Table 9

[00120] Referring to FIG. 7,a computing environment 702 implementing the method to optimize a network topology of a communication networks illustrated. As depicted, the computing environment 702includes at least one processing unit 704 that is equipped with a control unit 708 and an Arithmetic Logic Unit (ALU) 706, a memory 714, a storage unit 716, plurality of networking devices 710 and a plurality Input output (I/O) devices 712. The processing unit 704 is responsible for processing the instructions of the algorithm. The processing unit 704 receives commands from the control unit in order to perform its processing. Further, any logical and arithmetic operations involved in the execution of the instructions are computed with the help of the ALU 706.

[00121] The overall computing environment 702 can be composed of multiple homogeneous or heterogeneous cores, multiple CPUs of different kinds, special media and other accelerators. The processing unit 704 is responsible for processing the instructions of the algorithm. Further, the plurality of processing units 704 can be located on a single chip or over multiple chips.

[00122] The algorithm comprising of instructions and codes required for the implementation are stored in either the memory 714 or the storage 716 or both. At the time of execution, the instructions can be fetched from the corresponding memory 714or storage 716, and executed by the processing unit 704.

[00123] In case of any hardware implementations various networking devices 710 or external I/O devices 712can be connected to the computing environment to support the implementation through the networking unit and the I/O device unit.

[00124] The embodiments disclosed herein can be implemented through at least one software program running on at least one hardware device and performing network management functions to control the elements. The FIGS. 1 through 7 includes blocks which can be at least one of a hardware device, or a combination of hardware device and software module.

[00125] The foregoing description of the specific embodiments will so fully reveal the general nature of the embodiments herein that others can, by applying current knowledge, readily modify or adapt for various applications such specific embodiments without departing from the generic concept, and, therefore, such adaptations and modifications should and are intended to be comprehended within the meaning and range of equivalents of the disclosed embodiments. It is to be understood that the phraseology or terminology employed herein is for the purpose of description and not of limitation. Therefore, while the embodiments herein have been described in terms of preferred embodiments, those skilled in the art will recognize that the embodiments herein can be practiced with modification within the spirit and scope of the embodiments as described herein.