

A Centrality Entropy Maximization Problem in Shortest Path Routing Networks



Vanniyarajan Chellappan^{1,*}, Krishna M. Sivalingam¹, Kamala Krithivasan

Department of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai - 600036, India

ARTICLE INFO

Article history:

Received 21 June 2015

Revised 11 March 2016

Accepted 17 April 2016

Available online 30 April 2016

Keywords:

Traffic engineering

Topology design

Betweenness centrality

Entropy

Routing

OSPF

ABSTRACT

In the context of an IP network, this paper investigates an interesting case of the inverse shortest path problem using the concept of network centrality. For a given network, a special probability distribution, namely the *centrality distribution* associated with the links of a network can be determined based on the number of the shortest paths passing through each link. An entropy measure for this distribution is defined, and the inverse shortest path problem is formulated in terms of maximizing this entropy. We then obtain a centrality distribution that is as broadly distributed as possible subject to the topology constraints. A maximum entropy distribution signifies the decentralization of the network. An appropriate change in the weight of a link alters the number of the shortest paths that pass through it, thereby modifying the centrality distribution. The idea is to obtain a centrality distribution that maximizes the entropy. This problem is shown to be NP-hard, and a heuristic approach is proposed. An application to handling link failure scenarios in Open Shortest Path First routing is discussed.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In the general context of network design, *a well designed topology is the basis for all stable networks* [1]. Two main design considerations for a good network topology design are: (i) reducing the single point of failures that can occur in the network; and (ii) reducing the hop count between any origin-destination (OD) pair. We investigate appropriate topology measurements based on the structural properties of the network, and study how these can be utilized to determine the maximally efficient topology. In this context, it is important to study the influence a node or link may have on the larger network based on its structural position in the topology. This will help in the identification of critical nodes and/or links in the network. A network is said to be highly centralized if some of its nodes or links are extremely critical to the operation of the network. Such a highly critical node or link conflicts with the design goal of eliminating single points of failure.

In this paper, we investigate a network-wide measurement called *network centrality* to determine the centralization of a network, as an instance of graph complexity measure [2]. The centrality distribution associated with the nodes or links of a

network is determined based on the function of betweenness centrality values. A network wide measure is arrived by computing the entropy of the centrality distribution. We then formulate the network topology design problem as a problem of minimizing the centralization of the entire network or maximizing the entropy of the centrality distribution. We present a few interesting use cases of this proposed measure in the context of determining the efficiency of routing for a given topology.

Next, we study the inverse problem of determining the appropriate centrality distribution using suitable link weight setting techniques that maximize the entropy. This Centrality Entropy Maximization (CEM) problem is inspired by an earlier work called Network Entropy Maximization (NEM) [3], that connects the principle of maximum entropy with Internet Protocol (IP) routing. The CEM problem is shown to be NP-hard by reducing the known Open Shortest Path First (OSPF) optimal weight setting NP-hard problem [4]. We present a heuristic algorithm for the same. It is then shown, how this can be useful in handling link failure cases in OSPF networks. This paper consolidates and extends our previous work presented in [5,6].

The important contributions of this paper are summarized as follows: (i) studying the applicability of network centrality measure for network design problems; (ii) definition, proof of NP-hardness and a proposed heuristic solution for the centrality entropy maximization (CEM) problem; (iii) discuss various applications and use cases of the CEM framework including measuring the efficiency of routing in IP networks and understanding Braess Para-

* Corresponding author.

E-mail addresses: 003497@iitm.ac.in, vanniaraj@gmail.com (V. Chellappan), skrishnam@iitm.ac.in, krishna.sivalingam@gmail.com (K.M. Sivalingam).

¹ India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services (IU-ATC), Chennai, India.

dox in network routing; and (iv) an application of CEM for network topology design in tactical wireless networks.

The remainder of the paper is organized as follows. Section 2 presents the related work on entropy based measure of graph complexity and various centrality measures. Section 3 presents definitions along with some notations. Section 4 introduces the measure of *network centrality* and its variants. In Section 5, we show how the proposed network centrality measure can be applied to measure routing efficiency and detect Braess's paradox. Section 6 introduces the CEM problem and presents a heuristic approach to solve the same. Section 7 presents a use case of handling OSPF link failure case. Section 8 discusses the applicability of the proposed measure and the CEM framework to other interesting networking problems. Section 9 concludes the paper.

2. Related work

Centrality measures are often used in social networks to estimate the potential monitoring and control capabilities a person may have on communication flowing in the network. The concept of centrality has been extended to communication networks. Various centrality measures such as degree, closeness, and betweenness have been studied in the literature in order to analyze the internal topology of a given network [7]. These measures have been studied to quantify the influence of nodes or links on the dynamics of the entire network.

Betweenness centrality (BC) is one such graph theoretic concept that measures the degree to which a node or a link acts as an intermediary in the communication between every pair of nodes in the graph or topology. This measure of centrality is higher for certain nodes or links indicating that these nodes or links play a critical role. More precisely, the betweenness centrality of a node or link is determined by its occurrence in the shortest paths between pairs of nodes. There are different contexts in which *betweenness centrality* measure has been considered in a network [8–10]. The concept of Routing Betweenness Centrality (RBC) is introduced in [11], as a measure of the expected number of packets passing through a given node. A new edge betweenness centrality called *traffic-aware edge betweenness centrality* (TEBC) is defined in [12]. It is shown that TEBC can be used to influence and improve the performance of the shortest-path routing algorithm with respect to dynamic routing. Note that this metric is based on the fraction of traffic flow on an edge, and is used to re-balance the link's importance and lessen the problem of any bottleneck build-up on a link. More information on centrality related work can be found in [13,14].

The concept of node or link centrality has been extended further to the measurement of *network centrality* or *graph centrality* [7]. There are two distinct views in proposing such a graph- or network-wide measure. The first view leads to the development of measures of graph centrality based on the degree that all of its nodes or links are central. The alternative view leads to the development of measures of graph centrality based on the dominance of one node or link. We consider the first approach since it is more applicable in the network topology design problem.

We note that such a network-wide measure of centrality is also a measure of graph complexity. Graph complexity can be measured based on different structural features of the graph. For example, connectivity of a graph is measured based on node connectivity or link connectivity. The node or link connectivity is the smallest number of nodes or links whose removal results in a disconnected graph. This measure has been extended to measure the robustness of a network [15].

A taxonomy and overview of approaches to the measurement of graph complexity are presented in [2]. The taxonomy distinguishes

between deterministic and probabilistic approaches. In the probabilistic approach, a probability distribution associated with the vertices or edges of a graph is determined based on the structural properties of the graph. Then, an entropy function is applied to the probability distribution to derive the measure of complexity. An entropy function measures how close a probability distribution is to being uniformly distributed or quantifies the *unevenness* of the probability distribution.

Shannon's entropy function [16] is one of the most commonly used entropy functions in measuring the complexity of graphs. In the context of an entropy function, the Principle of Maximum Entropy aims to determine a uniform or as broad a probability distribution as possible subject to the available constraints [17,18]. This principle has been used in solving some interesting networking problems [19,20]. The first work connecting the principle of maximum entropy with IP routing is called Network Entropy Maximization (NEM) [3].

This section summarized the related work. The next section presents the necessary definitions and notations.

3. Definitions and notations

This section provides necessary technical background, definitions and preliminaries of this paper. A network in its simplest form is a set of nodes or vertices joined together in pairs by edges or links. It can be represented as a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. An edge is labeled as (u, v) or simply uv , where $u, v \in V$. In a directed graph, $uv \neq vu$. For routing purpose, we assume that there are no self-loops, and the paths connecting any pair of vertices are loop-free. A graph is said to be *weighted* when we assign weight to each of its edges. Let $w : E \rightarrow \mathbb{R}_{\geq 0}$ be the weight function. If G is not provided with a weight function on the edges, we assume that each edge has unit weight. The weight is represented by $w_{u,v}$ for the link $(u, v) \in E$. A path in a graph is a finite sequence of edges which connect a sequence of vertices which are all distinct from one another. A graph is said to be connected when every pair of vertices is joined by a path.

Definition 1 (*Geodesic or Shortest Path*). Given a connected weighted directed graph $G(V, E, w)$, associated with each edge $\langle u, v \rangle \in E$, there is a weight $w(u, v)$. The length of a path $p = \langle v_0, v_1, \dots, v_k \rangle$ is the sum of the weights of its constituent edges:

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i).$$

The length of the shortest path from u to v is defined by $\delta(u, v) = \min\{w(p) : p \text{ is a path from } u \text{ to } v\}$. $\delta(u, v)$ is called the distance between u and v . The path that realizes this distance is called the shortest path or geodesic.

There can be more than one shortest path between a pair of nodes.

Definition 2 (*$\langle s, t \rangle$ induced subgraph*). A sub-graph induced by the set of paths that connects the given source-destination pair $\langle s, t \rangle$. The union of paths that begin with s and end with t is called as the $\langle s, t \rangle$ induced subgraph, and is denoted by G_{st} .

Degree is a count of the number of edges incident upon a given node. The degree of a node is the simplest centrality measure of a node. It implies that the node with a higher degree has more incoming or outgoing paths, and hence critical to the entire network. Dividing it by the maximum possible degree $n - 1$ gives us a normalized measure.

Definition 3 (*Closeness Centrality*). As defined in [21], a node's closeness centrality is defined as the sum of the distances from

all other nodes in the graph, where the distance from a node to another is the length of the shortest path.

Since the number of nodes is fixed in a network, the measure is equivalent to the mean distance of a node to the other nodes. Apparently, this measure is an inverse measure of centrality since larger values indicate less centrality. So, technically it measures fairness rather than closeness [10].

The intuitive conception of *centrality* in communication was based upon the structural property of *betweenness*. According to this view, a node or a link in a communication network is central to the extent that it falls on the shortest path between pairs of vertices.

Definition 4 (*Betweenness Centrality with respect to $\langle s, t \rangle$ pair*). Let $\sigma_{s,t}$ represent the total number of the shortest paths between a pair $\langle s, t \rangle : s$ and $t \in V$. Let $\sigma_{s,t}(v)$ represent the number of shortest paths between $\langle s, t \rangle$ pair that pass through v . The vertex betweenness centrality of a node v with respect to a pair $\langle s, t \rangle$ is denoted by

$$\eta_{s,t}(v) = \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

Definition 5 (*Shortest Path Betweenness Centrality*). Let $\sigma_{\cdot,\cdot}$ represent the total number of shortest paths between every pair of source-destination nodes $\langle s, t \rangle : s$ and $t \in V$. Let $\sigma_{\cdot,\cdot}(v)$ represent the number of shortest paths between every pair of nodes $\langle s, t \rangle$ that pass through v . We define the Shortest Path Betweenness Centrality (SPBC) of a node v as

$$\eta_{\cdot,\cdot}(v) = \frac{\sigma_{\cdot,\cdot}(v)}{\sigma_{\cdot,\cdot}} \quad (1)$$

Note that this is different from the definition in [11], where it is defined as the sum of fractions of all shortest paths between each pair of nodes in a network which traverse a given node

$$\sum_{s \neq v \neq t} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

However, we define it as a fraction of all shortest paths connecting every pair of nodes which traverse a given node.

The above definition can be considered for links in the place of nodes. Let $\sigma_{s,t}(u, v)$ represent the number of shortest paths between $\langle s, t \rangle$ pair that pass through a link $(u, v) \in E$. The shortest path betweenness centrality of a link (u, v) with respect to a pair $\langle s, t \rangle$ is denoted by

$$\eta_{s,t}(u, v) = \frac{\sigma_{s,t}(u, v)}{\sigma_{s,t}}$$

The SPBC of a link (u, v) is defined as:

$$\eta_{\cdot,\cdot}(u, v) = \frac{\sigma_{\cdot,\cdot}(u, v)}{\sigma_{\cdot,\cdot}} \quad (2)$$

where $\sigma_{\cdot,\cdot}(u, v)$ represent the number of shortest paths between every pair of source-destination nodes $\langle s, t \rangle$ that pass through (u, v) .

In the article [16], Shannon suggested the following entropy function.

Definition 6 (*Shannon Entropy*). Let α be a random variable with a finite range a_1, \dots, a_n . Let p_i be the probability of the event $\alpha = a_i$. Then the Shannon entropy of α is defined as

$$H_n(\alpha) = - \sum_{i=1}^n p_i \log p_i \quad (3)$$

3.1. Entropy properties

We are mentioning a few of the well known entropy properties as we will be using them in subsequent proofs.

Using the concavity of the function $p \mapsto -p \log p$, one can prove that the Shannon entropy of every random variable does not exceed its *max-entropy*, $\hat{H}_n(\alpha)$, defined as the logarithm of the cardinality of the range of α (and is equal to $\hat{H}_n(\alpha)$ only for uniformly distributed variables). All logarithms in the paper are base 2.

$H(\alpha)$ is seen to be a function of p_1, p_2, \dots, p_n . It is also a continuous function and is a symmetric function. It does not change when an impossible outcome is added to the probability scheme. When one of the probabilities is unity and the others are zero, its value is zero and this is its minimum value, since $H \geq 0$ when $0 \leq p_i \leq 1$. It does not change if an impossible outcome is added to the probability scheme i.e.

$$H_{n+1}(p_1, p_2, \dots, p_n, 0) = H_n(p_1, p_2, \dots, p_n) \quad (4)$$

To find its maximum value, we can use Lagrange's method to maximize

$$- \sum_{i=1}^n p_i \log p_i - \lambda \left[\sum_{i=1}^n p_i - 1 \right] \quad (5)$$

and this gives us

$$p_1 = p_2 = \dots = p_n = \frac{1}{n} \quad (6)$$

Since $x \log x$ is a convex function, $\sum_{i=1}^n p_i \log p_i$ is a convex function, $-\sum_{i=1}^n p_i \log p_i$ is a concave function. Its local maximum is a global maximum. The maximum value of H_n is

$$- \sum_{i=1}^n \frac{1}{n} \log \left(\frac{1}{n} \right) = \log n \quad (7)$$

The maximum value of H_n increases as n increases.

Maximum Entropy Principle. The Maximum Entropy Principle (MEP) is a technique that can be used to estimate input probabilities [18]. The result is a probability distribution that maximizes H_n . Basically, the Maximum Entropy Principle aims to give us a *uniform* or as *broad* a distribution as possible, subject to the constraints being satisfied. For example, this is helpful in formulating optimization problems which need a maximum entropy probability distribution function given the mathematical expectations of random variables. This principle is used by Penalized Exponential Flow-splitting (PEFT) [3] to prove that hop-by-hop forwarding achieves *optimal* traffic engineering by splitting the traffic over multiple paths with an exponential penalty on longer paths.

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. MEP is used in this paper in the context of building a resilient network topology, measuring network complexity, and network analysis.

4. Network centrality

In this section, we review the existing results that apply entropy measures to graph complexity, and derive the *network centrality* as an entropy measure of a graph based on betweenness centrality of edges. The main idea is inspired from graph complexity measure [2], and can be summarized as follows. A probability value is assigned to each individual vertex or edge in a graph. This probability value is determined based on certain structural features of the graph. This generates a probability distribution associated with the graph. A measure of graph complexity is then arrived at,

by applying Shannon's entropy function to this probability distribution. In what follows, we present important definitions and results of our proposed measure.

In [2], the parametric graph entropy is defined as follows.

Definition 7 (Parametric Graph Entropy). Let $G = (V, E)$ be a graph and let f be an information function representing a positive function that maps vertices to the positive reals using structural features of a graph. Then, the parametric graph entropy is

$$I_f(G) = - \sum_{i=1}^{|V|} \left(\frac{f(v_i)}{\sum_{j=1}^{|V|} f(v_j)} \log \left(\frac{f(v_i)}{\sum_{j=1}^{|V|} f(v_j)} \right) \right) \quad (8)$$

The vertex probabilities are defined by

$$p(v_i) = \frac{f(v_i)}{\sum_{j=1}^{|V|} f(v_j)} \quad (9)$$

In this paper, we extend this definition to consider edges in place of vertices. The edge probabilities are defined by

$$p(u, v) = \frac{f(u, v)}{\sum_{(x, y) \in E} f(x, y)} \quad (10)$$

for every $(u, v) \in E$. The parametric graph entropy is defined by

$$I_f(G) = - \sum_{(u, v) \in E} p(u, v) \log p(u, v) \quad (11)$$

We extend the concept of node or link betweenness centrality further to the measurement of *network centrality*. More precisely, we define the measure of network centrality based on the extent that all of its nodes or links are central. In this paper, the network centrality is defined in the context of shortest path betweenness centrality of a node or link.

Definition 8 (Network Centrality). We define the network centrality or the entropy of SPBC by setting the information function f in Eq. (10) to the shortest path defined in Eq. (2). The edge probabilities based on SPBC are defined by

$$p(u, v) = \frac{\eta_{\bullet, \bullet}(u, v)}{\sum_{(x, y) \in E} \eta_{\bullet, \bullet}(x, y)} \quad (12)$$

for every $(u, v) \in E$. We obtain the entropy of SPBC when Eq. (12) is applied in Eq. (11).

Note that the information function f can be defined based on other structural properties of a graph, and used for solving different problems. The above entropy definitions can also be applied to a sub-graph induced by the set of paths that connects the given source-destination pair (s, t) . The union of paths that begin with s and end with t is called as the (s, t) induced subgraph, and is denoted by G_{st} .

Let Δ_G represent the random variable associated with the probability distribution formed from Eq. (12). Let Δ_{st} represent the random variable associated with the probability distribution associated with the graph G_{st} . Now, the entropy of SPBC is denoted by $H(\Delta_G)$ and the entropy of (s, t) SPBC is denoted by $H(\Delta_{st})$. Intuitively, the entropy function is used here in the context of how measures between nodes or links in the whole network are uniformly distributed. The topology with maximal entropy should be chosen so that the network is not centralized. Now, it leads to the question: how close is $H(\Delta_G)$ to the maximum entropy $\log(|E|)$. The concept of relative entropy is introduced as a normalized entropy measure with respect to the maximum entropy, allowing comparison of networks of arbitrary size.

Definition 9 (Relative Entropy). Relative entropy is defined by

$$h_G = \frac{H(\Delta_G)}{\log(|E|)}, \quad \text{and } 0 \leq h_G \leq 1 \quad (13)$$

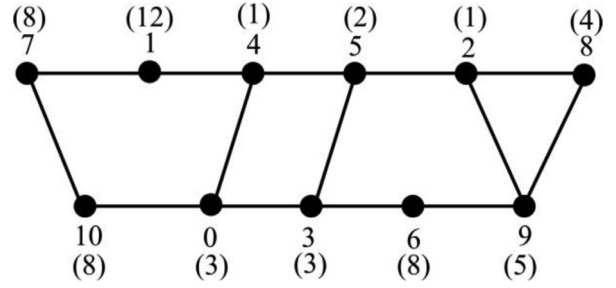


Fig. 1. The Abilene network topology.

Definition 10 (Diversity Index). Diversity index is relative entropy on (s, t) induced sub-graph G_{st} , and denoted by h_{st} .

The idea is to address some of the interesting network design and topology related problems by combining the MEP and parametric graph entropy with betweenness centrality as one of the parameters. It is important to note that the measures are not limited to shortest paths or paths in general. They can be extended and generalized for other structural features of a graph or network such as number of flows or packets, etc. Intuitively, these centrality measures can be regarded as the potential of a node or edge to control or influence communications in the network. Table 1 summarizes the major notations and symbols used in this paper.

5. Network centrality and routing

In this section, we present two important use cases of our proposed measure. The first use case is related to measuring the efficiency of routing in IP networks. The second use case is related to an interesting issue in the distribution of traffic flow called Braess's paradox in a network.

Topology studied

In our evaluation, we deal with the topology structure of the network, and use our proposed measure to determine the routing efficiency of the network. We considered the Abilene network topology as shown in Fig. 1, which has 11 core nodes in backbone and 14 bidirectional links connecting them. The core nodes are labeled from 0 to 10. These core nodes have multiple egress nodes, and the number of corresponding egress nodes are mentioned near each core node. These egress points serve to join the Abilene network with other Internet networks. The egress nodes are numbered, and given in Table 2. The traffic demands are extracted from the sampled Netflow data, and provided by Dahai Xu [3]. The network had 29 distinct egress points.

5.1. Routing efficiency

Traffic Engineering (TE) is considered as solving an optimization problem with an objective function that generally minimizes the congestion of the most utilized link. Optimal TE is realized by three important factors: (i) the topology structure, (ii) the traffic split decisions based on considerations of the internal structure of the routing topology and (iii) the traffic demands. Routing in a network deals with determining the best possible path for forwarding the traffic demands from sources to destinations. The routing architecture is generally a distributed one, where each router assumes the responsibility of determining the next best hop to forward the traffic towards its destination. All packets travel across the network along routes that are decided by routing protocols. Routing protocols maintain routing tables at each communication node or router to store the best possible paths to

Table 1
Table of notations

$w(u, v)$	\triangleq	Weight of link (u, v)
$\delta(u, v)$	\triangleq	Distance from u to v
$\sigma_{s, t}$	\triangleq	Number of shortest paths between $\langle s, t \rangle$ pair
$\sigma_{s, t}(u, v)$	\triangleq	Number of shortest paths between $\langle s, t \rangle$ pair that pass through the link (u, v)
$\sigma_{\cdot, \cdot}$	\triangleq	Total number of shortest paths between every $\langle s, t \rangle$ pair in graph G
$\sigma_{\cdot, \cdot}(u, v)$	\triangleq	Total number of shortest paths between every $\langle s, t \rangle$ pair that pass through the link (u, v)
$\eta_{s, t}(u, v)$	\triangleq	Shortest path betweenness centrality of the link (u, v) in $\langle s, t \rangle$ induced sub-graph
$\eta_{\cdot, \cdot}(u, v)$	\triangleq	Shortest path betweenness centrality of the link (u, v) in the graph G
$H_n(\alpha)$	\triangleq	Entropy of a discrete random variable α with n values
Δ_G	\triangleq	Random variable associated with betweenness centrality probabilities of graph G
h_G	\triangleq	Relative entropy or normalized entropy

Table 2

The egress set associated with each backbone node.

Core	Egress set
0	{12, 20, 29}
1	{14, 16, 17, 18, 20, 21, 23, 26, 32, 35, 38, 39}
2	{31}
3	{24, 25, 37}
4	{28}
5	{21, 30}
6	{13, 18, 23, 24, 29, 33, 34, 37}
7	{15, 20, 22, 23, 26, 33, 34, 39}
8	{11, 17, 32, 39}
9	{14, 20, 27, 35, 36}
10	{14, 15, 19, 32, 33, 36, 37, 38}

various destinations. Routing convergence is the process of updating the routing tables in each router such that each router has the same topological information. Any link or node failure causes re-convergence of the routing tables based on the modified topology.

OSPF. Dijkstra's shortest path algorithm is used widely in the Open Shortest Path First (OSPF) routing protocol. The OSPF protocol framework uses shortest-path algorithms to compute the shortest path tree rooted at a given node. In OSPF, traffic is routed along the shortest paths to the destination, where the path distance is determined by the link weights. The weights of the links, and thereby the shortest path routes, can be set by the network operator to meet the desired objectives. It supports destination-based hop-by-hop forwarding and Equal Cost Multi-path (ECMP) to evenly split the traffic over all available equal cost paths. Optimization of TE objectives, such as minimizing the maximum link utilization, can be accomplished by manipulating the link weight settings. Computing the optimal link weights ensuring even split of flows over equal cost multi-paths has been known to be an NP-hard problem [4]. The simplicity and distributed nature of OSPF allows it to scale for very large networks, but the network resource utilization may be sub-optimal.

It is important to study the performance of OSPF in comparison with the *optimal* routing. So, we consider routing betweenness centrality (RBC) [11] in the place of SPBC. This defines a new probability distribution based on the number of packets or the amount of flow realized on a link. In RBC, the information function f in Eqs. (8, [9,10]) represents the number of packets that pass through a node or link using the chosen routing scheme.

To illustrate this, on the example network, we computed the entropy of centrality values based on the number of packets that flow on a given link. The link cost function used is a piecewise-

linear approximation of the M/M/1 queue's delay formula [22]:

$$\Phi(f_{uv}, c_{uv}) = \begin{cases} f_{uv} & \frac{f_{uv}}{c_{uv}} \leq \frac{1}{3} \\ 3f_{uv} - \frac{2}{3}c_{uv} & \frac{1}{3} \leq \frac{f_{uv}}{c_{uv}} \leq \frac{2}{3} \\ 10f_{uv} - \frac{16}{3}c_{uv} & \frac{2}{3} \leq \frac{f_{uv}}{c_{uv}} \leq \frac{9}{10} \\ 70f_{uv} - \frac{178}{3}c_{uv} & \frac{9}{10} \leq \frac{f_{uv}}{c_{uv}} \leq 1 \\ 500f_{uv} - \frac{1468}{3}c_{uv} & 1 \leq \frac{f_{uv}}{c_{uv}} \leq \frac{11}{10} \\ 5000f_{uv} - \frac{16318}{3}c_{uv} & \frac{11}{10} \leq \frac{f_{uv}}{c_{uv}} \end{cases} \quad (14)$$

and the objective is to minimize $\sum_{(uv) \in E} \Phi(f_{uv}, c_{uv})$ subject to the link capacity constraints and flow conservation.

On the given network, the optimal flow values f_{uv} for the above objective function, given the traffic demand, are computed using CPLEX on AMPL. The edge or link probabilities based on the optimal flow values are computed by

$$p(u, v) = \frac{f_{uv}}{\sum_{(x,y) \in E} f_{xy}} \quad (15)$$

Eq. (15) is applied in Eq. (11) to obtain the entropy of flow distribution. The value of entropy of flow based distribution for the given network is $H_{28} = 4.453392$, and the relative entropy $h_G = 0.926371$.

Let \hat{f}_{uv} denote the flow realized by the OSPF ECMP routing for the same demand on the same network. The entropy function on the flow distribution \hat{f}_{uv} is $H_{28} = 4.137584$, and the associated relative entropy $h_G = 0.86$. This shows that OSPF ECMP routing strategy is sub-optimal not only in terms of the cost function, but also in terms of the flow centrality values of links. It provides a good handle in analyzing the efficiency of each link in the context of the chosen routing strategy and the topology design. As noted earlier, this also leads to the mechanism of a routing protocol that considers non-shortest paths and associated traffic splitting to achieve optimal Traffic Engineering [3]. The reader is referred to Appendix B for an interesting use case in Tactical Communication Systems.

5.2. Braess' paradox

Braess' Paradox [23] states that removing links from a network can improve its performance or adding a new link to a congested network can make it worse. In the case of OSPF routing, the shortest paths are computed based on the link weights. Hence, the routing patterns are easily predictable to occur along the shortest paths. The links with high betweenness centrality values are the ones that have the maximum shortest paths passing through them. Hence, when a new link is added to the given network, it induces several paths between various origin-destination pairs, and in turn induces the shortest paths between them based on the weight assigned to it. These newly created paths will alter the betweenness centrality probability distribution based on the structural position of the new link in the network. A reduction in the entropy of betweenness centrality distribution indicates that the new link has

Table 3
Remove links in the Abilene network.

Link	$H(\Delta_G)$	h_G
(2, 9)	4.590392	0.976588
(0,4)	4.591638	0.976853
(3, 5) [†]	4.617480	0.971102
(4, 0) [†]	4.624594	0.972598
(0, 4) [†]	4.626161	0.972928
(9, 2) [†]	4.639608	0.975756

created imbalance in the role of each node or link. If the entropy is maintained or increased, then it shows that the new link has indeed created a better, decentralized network. A formal proof on why the entropy increases when the network imbalance decreases is given in Section 6.4.

In a network, Braess' paradox occurs because, under the common approach to pricing the link utilization, traffic flows attempt to minimize their delay while ignoring the effect of their decisions on other flows. In the context of cloud networks elasticity and energy aware routing, it is possible for the total system to experience delays following an expansion or reduction of the network due to Braess' paradox. It is also common to have redundant paths between several nodes for achieving fault tolerance in networks. In such case, it is possible that the network may experience Braess' paradox due such redundant paths. To illustrate this, a few links, both unidirectional (marked †) and bidirectional, are removed, and the corresponding entropy values are given in Table 3. For example, when links (2, 9) and (9, 2) are removed, the relative entropy has increased. Note that the actual entropy has decreased to 4.590392 from 4.662206. This is because of the reduction in the total number of links (from 28 to 26). When the link (0, 4) is removed, a broader SPBC distribution is realized with increased relative entropy.

Flow. We also computed the entropy function on the actual flow values after removing the link (0, 4), and found that the relative entropy h_G has increased to 0.867383 (from 0.860681) in OSPF ECMP routing, and to 0.933951 (from 0.926371) in optimal routing. This concurs with the presence of such a paradox.

6. Centrality entropy maximization (CEM)

In this section, we model the SPBC Entropy Maximization problem, and present an approach for finding the weight assignment to each link that results in a shortest path betweenness centrality distribution that maximizes the entropy. The link weights determine the shortest paths between every origin-destination pair. The shortest paths in turn determine the betweenness centrality of a node or link, which in turn determine the associated probability value of a node or link. The problem of Centrality Entropy Maximization (CEM) can be regarded as a problem of entropy maximization based on the maximum entropy principle. The problem aims to obtain a uniform or as *broad* a shortest path betweenness centrality distribution as possible subject to the topology constraints.

The shortest path problem is: given the network and link weights, determine the shortest paths. Conceptually, the inverse shortest path problem is just the other way around. The problem is to determine the link weights such that the given sets of paths become the sets of the shortest paths. In the case of CEM, the problem is to determine the appropriate link weights that produce the desired betweenness centrality values that maximize the entropy. CEM combines the Maximum Entropy Principle with the inverse shortest path problem.

6.1. The CEM problem

We assume a weighted graph $G = (V, E, w)$, where V is the set of nodes, E is the set of edges and w is a weight function $w : E \rightarrow \mathbb{R}_{\geq 0}$. w_{uv} denotes the weight assigned to link $(u, v) \in E$. The problem is to determine the weights $\{w_{uv}\}_{uv \in E}$ such that we obtain a uniform or as *broad* a shortest path betweenness centrality distribution as possible that maximizes the entropy function. Recall the probability distribution obtained from the Eq. 12. We define the Centrality Entropy Maximization (CEM) problem as follows.

$$\text{Max } H(\Delta_G) = - \sum_{(u,v) \in E} p_{u,v} \log p_{u,v}$$

subject to the constraints

$$p_{u,v} \geq 0, \quad \sum_{(u,v) \in E} p_{u,v} = 1,$$

and $w_{u,v} \geq 0$

For the CEM problem stated above, we have

Theorem 1. *It is NP-hard to maximize the entropy of the shortest path betweenness centrality distribution*

The proof of NP-hardness is by using a reduction from the well known NP-hard problem of finding an optimal setting of the OSPF weights [4]. The detailed proof is given in Appendix A.

A local search heuristic Algorithm 1 (Centrality Entropy Maximization) is proposed to find the weight settings that maximize the entropy of the shortest path betweenness centrality distribution. The working details of the algorithm are presented in the following section.

Algorithm 1 Centrality Entropy Maximization

- 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: $D \leftarrow$ Path distance of k OD pairs
 - 9: $w \leftarrow w_{u,v}$
 - 10: $w_{u,v} \leftarrow \infty$
 - 11: $D' \leftarrow$ Path distance of k OD pairs
 - 12: $d_i \leftarrow D'_i - D_i \quad i = 1, 2, \dots, k$
 - 13: $\delta_1 \leftarrow$ first smallest d_i
 - 14: $\delta_2 \leftarrow$ second smallest d_i
 - 15: $w_{u,v} \leftarrow w + \frac{\delta_1 + \delta_2}{2}$ {Increment the weight}
 - 16: $P \leftarrow P \cup H(\Delta_G)$ (Neighborhood of P_0)
 - 17: **end for**
 - 18: $P_{max} \leftarrow \text{Max}(P)$ {Increased Entropy over P_0 }
 - 19: **if** no improvement over P_0 **then** return W
 - 20: **else** $W \leftarrow$ modified weight vector corresponding to P_{max}
 - 21: $P_0 \leftarrow P_{max}$
 - 22: **repeat** from step 4
-

6.2. Solving CEM using local search

We consider the optimal shortest-path routing algorithm discussed in [24] to solve the CEM problem. Let W_0 denote the initial weight assignment vector. Let P_0 denote the value of entropy in Eq. (16) obtained from the weight assignment vector W_0 . We define a

neighborhood of P_0 as $\{P\}$, where $\{P\}$ is a set of points with cardinality $|E|$. Each member in $\{P\}$ is a value of entropy function in Eq. (16), 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 $\{P\}$, we chose the maximum value 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} . We repeat this procedure with the newly discovered neighborhood until we do not find such a P_{max} .

Neighborhood discovery. The neighborhood discovery is also similar to [24]. Let us consider a link $(u, v) \in E$. Let $\sigma_{u,v}^0(u, v)$ represent the number of the 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 $\sigma_{u,v}(u, v)$, and in turn, the entropy.

Let k denote the number of origin-destination (OD) pairs that have the shortest paths pass through (u, v) . Let S_i denote the set of the 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 the 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, we have diverted all paths that go through the link (u, v) . 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.

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

If there is no d_i with value 0, then we increment the value of $w_{u,v}$ by the mid-point between the first smallest difference and the second smallest difference. This will ensure that we divert only those paths that have the smallest difference in their distance after and before removing the link. In this case, a few OD pairs will have an increase in their path distance. So, while we distribute the paths between OD pairs to improve the entropy, we also increase the path lengths of some OD pairs. This may lead to more hop counts for some OD pairs.

This condition conflicts with the design goal of reducing the hop count or path distance between OD pairs. We can balance these design goals, by having an additional constraint such that we don't divert any paths if the difference is beyond a specified limit. In our experiments, we have noted that in certain cases, diverting shortest paths from a link also results in the increase in the number of the shortest paths between some OD pairs. This approach has the same convergence properties and run time complexity is exactly as discussed in [24].

More precisely, the appropriate increase in the weight $w_{u,v}$ maximizes the entropy of SPBC distribution, and minimizes the number of the shortest paths throughout the network. This approach can also be extended for achieving maximum entropy of the 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 [25]. Maximizing the entropy bal-

Table 4
CEM - The Abilene network.

Link	Weight	Entropy	$\sigma_{u,v}$
Initial	N.A	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

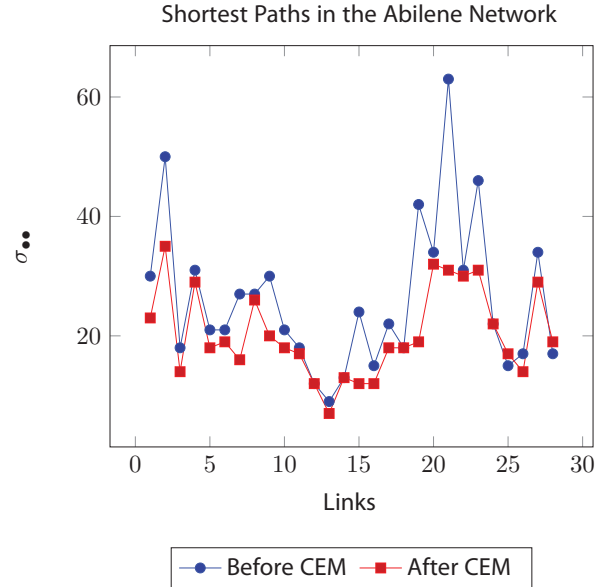


Fig. 2. Shortest Paths Distribution before and after CEM is shown. X axis represents the link numbers, and Y axis represents the number of the shortest paths that pass through a given link. After CEM, the number of the shortest paths is reduced for the highly centralized links. This is due to the CEM.

ances 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' paradox, quick routing convergence [26] 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.

6.3. Evaluation

We illustrate the procedure for maximizing the entropy of SPBC with two sample networks. The first one is the Abilene network shown in Fig. 1. We assume that all links have equal capacity, and each edge has unit weight. The progress of the algorithm is presented in Table 4. 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.

As we can see from Table 4, the number of the shortest paths throughout the network has come down. The distribution on the number of shortest paths that pass through each link before and after CEM is given in Fig. 2. We can observe that some highly centralized links are decentralized to the extent possible. This results in a broader SPBC probability distribution, which in turn results in increased entropy value. Intuitively, we have made the role of each link as uniform as possible, subject to topology constraints.

This experiment has been done on Hier50a Network. It consists of 50 nodes and 148 links. We assume that each edge has unit

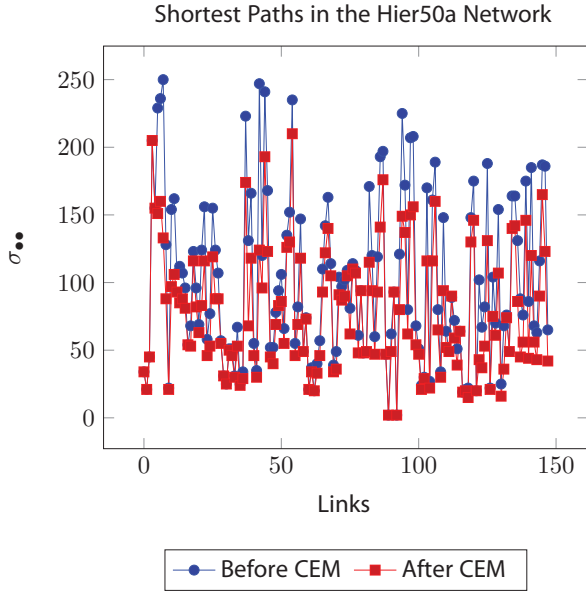


Fig. 3. Shortest Paths Distribution before and after CEM on Hier50a Network. Essentially CEM decentralizes some of the critical links, and makes the distribution as broad as possible.

weight. The summarized result is 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 of the shortest paths becomes 2811. The algorithm modified weights of 20 links. [Figure 3](#) shows the test results mentioned above.

This approach of weight setting in combination with the addition of new links or removal of existing links can assist in measuring and meeting the goals of network design. This approach can be extended considering the capacity of each link too. The centrality measure for a link (u, v) will then be $\frac{\sigma_{..}(u, v)}{C_{u, v}}$, where $C_{u, v}$ represents the capacity. This will ensure that the centrality values are normalized with respect to the capacity of each link.

6.4. Entropy and network imbalance

We formally prove how entropy is viewed as a measure of network imbalance represented by a centrality distribution, with higher entropy corresponding to a more balanced network and lower entropy corresponding to a more imbalanced network. More precisely, we need to show that the centrality distribution with maximum entropy, satisfying whatever constraints we impose, is the one that should be chosen in terms of the shortest paths distributed over the network. We present and prove the following theorem on network imbalance and entropy measure. The proof is based on the well known theorem on the upper bound of entropy that is presented and proved in [\[27\]](#).

Theorem 2 (Centrality entropy and network imbalance are inversely proportional). *The entropy $H(\Delta_G)$ increases as the network imbalance decreases, where Δ_G represents the shortest path betweenness centrality distribution for the given graph G .*

Proof. We prove this theorem based on the proof given for Theorem 3.1 in [\[27\]](#). The distribution Δ_G is defined on the set of edges $\{e_1, e_2, \dots, e_m\}$ with the corresponding probabilities p_1, p_2, \dots, p_m . The probabilities are computed from [Eq. 12](#) that add up to 1. Entropy is a function of the m -tuples. We want to show that when

we decrease the network imbalance, the entropy increases. Let us assume that Δ_G is not a uniform distribution, and hence the p_j are not all equal. Let us say $p_1 < p_2$. It implies that the number of the shortest paths passing through the edge e_1 is less than the number of the shortest paths passing through e_2 .

When we try to decrease the network imbalance, we try to either increase the number of the shortest paths passing through e_1 with respect to e_2 or decrease the number of the shortest paths passing through e_2 with respect to e_1 . More precisely, it adds a small value ϵ to p_1 , and reduces ϵ from p_2 so that the sum of probabilities add up to 1. So, the new probabilities after our act of balancing (adjusting weights) will be $p_1 + \epsilon, p_2 - \epsilon, \dots, p_m$. It is enough to show that the entropy of $\{p_1 + \epsilon, p_2 - \epsilon, \dots, p_m\}$ is greater than the entropy of $\{p_1, p_2, \dots, p_m\}$. In other words, $H(p_1 + \epsilon, p_2 - \epsilon, \dots, p_m)$ minus $H(p_1, p_2, \dots, p_m)$ should be positive.

Since $p_1 < p_2$, for small positive ϵ we have $p_1 + \epsilon < p_2 + \epsilon$. $H(p_1 + \epsilon, p_2 - \epsilon, \dots, p_m)$ minus $H(p_1, p_2, \dots, p_m)$ equals

$$\begin{aligned} & -p_1 \log \left(\frac{p_1 + \epsilon}{p_1} \right) - \epsilon \log(p_1 + \epsilon) \\ & -p_2 \log \left(\frac{p_2 - \epsilon}{p_2} \right) + \epsilon \log(p_2 - \epsilon) \end{aligned} \quad (17)$$

$$\begin{aligned} & -p_1 \log \left(\frac{p_1 + \epsilon}{p_1} \right) - \epsilon \log \left(p_1 \left(1 + \frac{\epsilon}{p_1} \right) \right) \\ & -p_2 \log \left(\frac{p_2 - \epsilon}{p_2} \right) + \epsilon \log \left(p_2 \left(1 - \frac{\epsilon}{p_2} \right) \right) \end{aligned} \quad (18)$$

$$\begin{aligned} & -p_1 \log \left(1 + \frac{\epsilon}{p_1} \right) - \epsilon \left(\log p_1 + \log \left(1 + \frac{\epsilon}{p_1} \right) \right) \\ & -p_2 \log \left(1 - \frac{\epsilon}{p_2} \right) + \epsilon \left(\log p_2 + \log \left(1 - \frac{\epsilon}{p_2} \right) \right) \end{aligned} \quad (19)$$

Since $\log(1 + x) = x + O(x^2)$ for small x , [Eq. 19](#) can be written as:

$$-\epsilon - \epsilon \log p_1 + \epsilon + \epsilon \log p_2 + O(\epsilon^2) = \epsilon \log \left(\frac{p_2}{p_1} \right) + O(\epsilon^2) \quad (20)$$

This value is positive when ϵ is small enough since $p_1 < p_2$. Therefore, the entropy increases when the network imbalance decreases. \square

7. OSPF single link failure

In this section, we present another important use of the CEM problem. It is an important goal of network and routing protocol design to quickly react to the changes in the network topology such as link or node failures. Also, as noted in [\[9\]](#), in the context of network resilience, a link or node failure may have varying impact on the entire network. It implies that the failure of a link with minimum centrality may imbalance the centrality distribution to a greater extent. In an earlier section, we have seen how the entropy of centrality distribution is affected when a link goes down. In this section, we will show how the entropy measure is useful in handling such single link failure cases.

Note that when a link goes down, the shortest paths that pass through the link are diverted through other links. This results in an 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 may not be efficient for a network with link failures. One option is to arrive a different set of weights that maximizes the entropy of centrality distribution with link failure. It is not a practical approach to assign new weights for a single link failure. In [\[28\]](#), the problem of single link failure issue in OSPF is well motivated, and it investigates the problem with an objective to find a weight setting which results in efficient shortest path first routing in normal and failure cases. The basic idea is to minimize the average cost of both cases, with and without link failure.

7.1. Proposed algorithm

We extend the principle of maximum entropy to find the appropriate weight setting for both cases. Let $G = (V, E)$ represent the network under normal condition. Recall the technique we used in CEM to improve the entropy of SPBC. We set the link weight $w_{u, v}$ to infinity, $(u, v) \in E$, and divert all paths that go through the link. Let us assume that there are k OD pairs that have the shortest paths pass through the link (u, v) . Now we find a suitable increment in the weight $w_{u, v}$ such that the paths that suffered the least increase in distance are diverted. We recompute the SPBC of each link, and then compute the entropy of SPBC of G . 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 Eq. (13). W_i denotes the weight vector assignment at the i th iteration.

Consider a link $(x, y) \in E$. Let G^{xy} represent the network without the link (x, y) . It is equivalent to setting w_{xy} to infinity in G . The remaining weights of the links are kept intact. Now, we compute the entropy of SPBC of G^{xy} . Let $H_m(\Delta_{G^{xy}})$ represent the entropy of SPBC, where m denotes the number of links in G^{xy} . Let $h_{G^{xy}}$ represent the relative entropy of SPBC as specified in Eq. (13). Our goal is to find the appropriate weights assignment vector that balances and improves both h_G and $h_{G^{xy}}$.

The idea is to apply entropy function on the relative entropy values and attempt to improve it. We normalize h_G and $h_{G^{xy}}$ 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 entropy values $\{\hat{h}_G, \hat{h}_{G^{xy}}\}$ form a probability distribution. Now, we maximize the objective function

$$z = -\hat{h}_G \log \hat{h}_G - \hat{h}_{G^{xy}} \log \hat{h}_{G^{xy}} \quad (21)$$

The maximum value of z is $\log 2 = 1$. The relative entropy value is a normalized measure with respect to the maximum entropy. Maximizing Eq. (21) results in a uniform distribution of relative entropy values. More precisely, the weights of the links are chosen such a way that the centrality distribution before and after link failure is as close as possible. This helps us maintain the same set of weights under certain link failure. This approach has one drawback, in that the relative entropy of normal network may come down significantly to match the relative entropy of failure case. To avoid this condition, we add one more condition 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 failure case is maximized at each iteration.

7.2. Evaluation

We illustrate the procedure for maximizing the entropy of SPBC for single link failure on the Abilene network shown in Fig. 1. We consider the failure of link (4, 1). The progress of the algorithm is shown in Table 5. In this example, we have added the condition that the relative entropy of normal network should not be less than the initial relative entropy value of 0.969807. The number of the shortest paths that pass through each link for the three cases are given in Fig. 4. Blue line represents the number of the shortest paths that pass through each link at the initial stage. Red line represents the number of the shortest paths that pass through each link under normal network condition when the relative entropy h_G reaches the maximum value of 0.969887. Brown line represents the number of the shortest paths that pass through each link under a link (4, 1) failure condition when the relative entropy $h_{G^{41}}$ reaches 0.848751.

Table 5

Entropy maximization for link failure (4, 1).

Link	Weight	h_G	$h_{G^{41}}$	$\sigma_{\cdot, \cdot}$
Initial	N.A	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

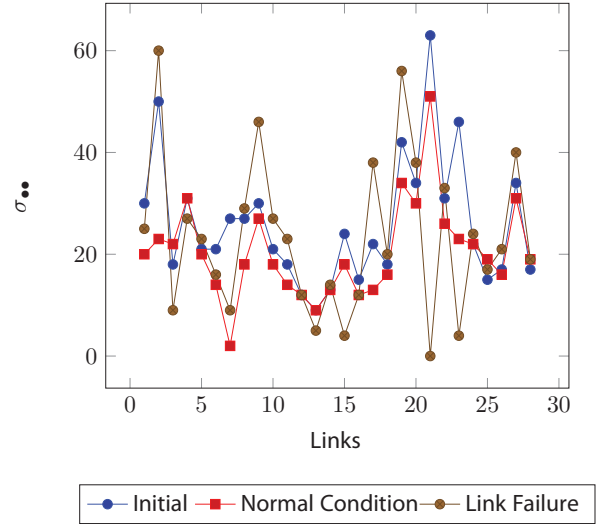


Fig. 4. The number of the shortest paths that pass through each link under three conditions for link (4,1) failure.(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6

Entropy maximization for link failure (0, 10)

Link	Weight	h_G	$h_{G^{0-10}}$	$\sigma_{\cdot, \cdot}$
Initial	N.A	0.969807	0.830578	404
(0, 4) [†]	1.5	0.967054	0.851834	363
(7, 10) [†]	3.0	0.959042	0.865836	363
(0, 3) [†]	1.25	0.958441	0.872296	352
(0, 4) [†]	2.625	0.954229	0.874558	352
(4, 1) [†]	1.5	0.956530	0.881265	339
(0, 3) [†]	2.3125	0.950855	0.882863	338

Though this approach balances both entropy values, 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 could make some other link highly central. This justifies the claim made in [9] 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.

The experiment has been repeated on the same network with the failure of link (0, 10). Table 6 represents the progress of the proposed algorithm. Fig. 5 represents the shortest paths distribution under different conditions stated above. The minimum relative entropy for the network under normal condition is set to 0.95. In this case, we also note that a few links experience very low centrality. For example, there is only one shortest path, which passes through the link (0, 4) under normal network condition.

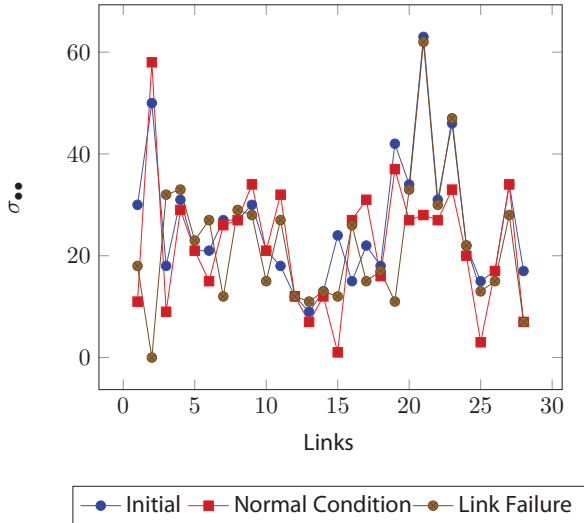


Fig. 5. The number of the shortest paths that pass through each link under three conditions for link (0,10) failure.

Table 7
Entropy of flows betweenness centrality - link down

Link	$H(\Delta_C)$	Cost	Link	$H(\Delta_C)$	Cost
(0, 3)	4.07677	71,830	(0, 4)	4.13694	69,411
(0, 10)	3.99211	75,734	(1, 4)	4.27560	73,913
(1, 7)	4.04998	74,023	(10, 7)	4.13189	70,423
(2, 5)	4.14413	72,318	(2, 8)	4.09053	85,244
(2, 9)	4.13029	70,963	(3, 0)	4.04982	71,552
(3, 5)	4.13375	69,552	(3, 6)	4.06995	73,775
(4, 0)	4.07474	70,795	(4, 1)	4.08642	74,568
(4,5)	4.30861	74,375	(5, 2)	4.218387	85,048
(5, 3)	4.08700	69,871	(5, 4)	4.08722	72,405
(6, 3)	4.11653	71,191	(6, 9)	4.11116	70,565
(7, 1)	4.17431	70,525	(7, 10)	4.08910	71,021
(8, 2)	4.14992	69,844	(8, 9)	4.12591	69,582
(9, 2)	4.13951	69,807	(9, 6)	4.09230	70,808
(9, 8)	4.11609	71,116	(10, 0)	4.08462	71,397

7.3. OSPF weights

The CEM leads to some interesting observations in OSPF traffic engineering. A simple and intuitive method in OSPF weight setting is inverse capacity weight setting. In this method, the weight of each link is the inverse of the link capacity. Intuitively, the shortest paths will have links with higher capacities, and hence a better utilization can be considered. This method does not consider traffic demands. Using CEM, we can consider the same approach, but improve the performance of OSPF routing for the given traffic demand.

For the Abilene network shown in Fig. 1 with a given traffic demand, we used the inverse capacity weight setting, and the entropy of flow centrality for OSPF routing is 4.137584. The cost is 69568. Instead of computing weights for optimal performance, we simply turned off each link, and computed the entropy of flow centrality. Table 7 represents the entropy values and the cost in the absence of each link. When the link (4, 5) is shut down, and with inverse capacity weight setting, the entropy is increased considerably with marginal increase in the routing cost. This helps network operators use inverse weight setting with an option of turning off a link to improve the routing performance. Several such *what-if* scenarios can be experimented based on different traffic demands and other conditions, and a suitable network topology and link weights can be identified to improve the OSPF routing performance. More

precisely, determining appropriate centrality distribution subject to maximum entropy helps us solve several network design problems.

8. Discussion

In this section, we present some interesting applications of CEM framework. We note that the problem of robust network design can be transformed to a problem of determining appropriate maximum entropy betweenness centrality distribution subject to certain constraints. We can also consider other distributions based on the structural features of the graph or network other than betweenness centrality as discussed below.

8.1. MPLS

MultiPath Label Switching (MPLS) works on tunnel based technology, and packet forwarding is based on the labels. The primary objective is to optimize the bandwidth utilization and fast re-route protection in case of any link or node failures. It uses the topology information, traffic demands along with link metrics information for path computation. Once an admissible path between a pair of nodes is found, based on various constraints, a tunnel will be created. This is also called Label-Switched Path (LSP).

Link or node protection is a technique to ensure that the traffic will be carried by pre-provisioning protection tunnels when a link or node goes down. An important consideration in link or node protection is that the protection tunnel or backup path should be carefully chosen so that it does not create unbalanced link utilization. We can extend the concept of SPBC to LSP betweenness centrality. This is defined as the number of LSPs that pass through a link or a node. This can be normalized over the total number of LSPs throughout the network. When a secondary path is chosen for fast re-route, the alternate paths are computed and appropriate entry and merger points are identified. The objective is to maximize the entropy of LSP betweenness centrality distribution so that the LSPs are distributed efficiently.

8.2. Software defined networks

Software Defined Networking (SDN) [29,30] is an approach to separate the control and data planes in the network architecture, where a central controller can implement a customized control plane, interacting with the network elements through programmatic interfaces such as OpenFlow [31]. The controller can modify the configuration of the network element, and effectively centralizes the network intelligence and state. This enables the incorporation of network specific customizations such as modification of the packet forwarding logic to suit the requirements of applications running in the network, and provides for easier implementation of on-line Traffic Engineering methods.

The main goal of Software Defined Networks (SDN) is to provide programming capability to networks, particularly routing. The main idea is to provide dynamic programmable controls to achieve optimal traffic engineering. Since the demands are not fixed, and varying under several circumstances, SDNs provide the needed intelligence to maximize the network resources utilization. We believe, the concepts of centrality, its distribution and entropy provide vital information on significance of network resources or elements for the given traffic demand. For example, topologies can be modified by the way of adding or removing wired or wireless links dynamically to meet the traffic demand, energy constraints using this framework. Since these measures are easy to compute, and can quickly reflect the state of network, they can be very useful in SDNs.

8.3. Autonomic networks

Autonomic Networking aims to create self-managing networks to overcome the rapidly growing complexity of the Internet and other networks and to enable their further growth, far beyond the size of today. IP networking with hop-by-hop forwarding, enables each router to take an independent routing decision. However, subsequent developments have seen more and more intelligence embedded into the configuration of the network element by a central manager or manually, rather than into the protocols. Autonomic networking aims to restore the intelligence back into the protocols and remove the dependency on central management.

The concepts and approach presented in this paper fit in into this autonomic networking framework, where we can model the optimization problem based on the proposed entropy function to solve a class of network synthesis problems where a certain characteristic of the network is specified, and one has to design a network that satisfies the constraints. This can be an overlay network design evolved by the network elements autonomously, independent of the physical topology. In the same context of autonomic computing, in [32], a probabilistic routing algorithm is proposed with the goal of robustly managing the demands from any source to any destination and have the minimum service interruption. This is based on the concepts of Path Criticality metric, in turn derived from Betweenness Centrality. The paper postulates that in order to have a robust system, we need to find the *least biased* setting, implying entropy maximization. Maximizing entropy gives the appropriate capacity of the network elements to maintain the highest degree of robustness.

8.4. Network design

We now consider the significance of this work in the context of the challenges posed by the complexity of today's networks which are growing rapidly at an exponential rate. In [33], it is argued that the increasing scale and complexity of emerging networks implies a transition from human centric network management to more autonomic and adaptive mechanisms, and proposes an autonomic topology control approach that builds on concepts of emergence, self-organization, and graph theory to evolve and adapt the network topology to satisfy dynamically changing application requirements. This is based on the insight that the topology of a network is fundamental to its operation, and the lack of control over it has direct implications on performance, resilience, and security of the network. It also proposes a network entropy metric as a composite of several graph theoretic metrics, and demonstrate its application to a specific application of restructuring the topology to a given set of constraints.

Note that the formal definition of Traffic Engineering [34] encompasses performance evaluation and performance optimization of operational networks. As a control system, it has a proactive perspective, involving capacity planning and augmentation, routing control, traffic control and resource control. In the reactive case, the control system responds correctively and perhaps adaptively to events that have already transpired in the network, such as changes in traffic demand. The packet level processing functions react to the real-time statistical behavior of traffic in milliseconds. Path selection and traffic split across equal or unequal cost paths can be viewed as a reactive control.

It is interesting to note that this paper has provided a theoretical approach to address both the short term and long term control functions. Viewed in another way, in the context of contemporary and emerging trends in network management, the concepts developed in this paper can be applied to Autonomic Networking and to SDN. Autonomic Networking is also a natural complement to SDN. For example, an autonomic network needs central input

for consistent network policy. A centrally controlled network, on the other hand, needs embedded intelligence in order to reroute quickly enough around a link failure. Re-routing will need to happen in no more than 50 ms, so that the failure is not noticeable. In the context of this paper, it may be seen that the proposed CEM algorithm and the design of a maximally efficient topology design using the network wide entropy measure can be embedded into the SDN controller for the network. The reactive packet processing for path selection and traffic split in response to dynamic conditions can serve as the autonomic component in this architecture.

9. Conclusions

The paper presents a new way of measuring the *centralization* of the network based on the entropy of the shortest path betweenness centrality distribution. We opine that this measurement makes a strong case for considering betweenness centrality distribution and its entropy in designing maximally efficient topologies, and improving the resiliency and quality of routing. The tactical network only serves as a use-case model; however, the idea is applicable to other networks too. The CEM framework can be extended to other centrality measures such as Routing Betweenness centrality to solve inverse shortest path problems in the areas of Traffic Engineering and Routing optimization. Additionally, there is a class of multi-layer topology design problems that can synthesize the optimal topology at both physical and logical layers, that can be analyzed using this framework. It will be worthwhile to build a routing strategy that can automatically detect the occurrence of Braess' paradox in the network based on the entropy of centrality distribution.

Appendix A. Proof of NP-hardness of the CEM problem

In this section, we prove the NP-hardness of the CEM problem in Eq. 16. A problem is *NP-hard* if every problem in the class NP of problems, solvable in polynomial time on a non-deterministic Turing Machine, can be transformed to it. Usually a problem is shown to be NP-hard by reducing a known NP-complete problem to it. We prove the hardness result by reducing the problem of optimizing OSPF weight setting with respect to maximum link utilization to the problem of maximizing the entropy of the shortest path centrality distribution.

First, we present an overview of the NP-hardness result of the OSPF weight setting problem [4]. Consider a network as a directed graph $G = (V, E)$. The traffic demands are represented by a demand matrix D . Let D_{st} represent the traffic demand between a source-destination pair (s, t) ; f_{uv} denote the flow on link (u, v) , and c_{uv} denote the capacity of the link (u, v) . The link utilization is defined by $\Phi(f_{uv}, c_{uv}) = \frac{f_{uv}}{c_{uv}}$. The maximum link utilization is $\max_{uv \in E} \frac{f_{uv}}{c_{uv}}$. The objective is to distribute the traffic demands to minimize the maximum link utilization subject to the link capacity constraints and flow conservation. The general routing problem can be formulated as follows.

$$\text{Min } \Phi = \text{Max}_{(x,y) \in E} \Phi(f_{xy}, c_{xy})$$

subject to

$$\forall D_{st} :$$

$$\begin{aligned} f_+(x) &= f_-(x) \quad \forall x \in V \setminus \{s, t\} \\ f_+(x) &= \sum_{y:xy \in E} f_{xy} \quad \text{and} \quad f_-(x) = \sum_{y:yx \in E} f_{yx} \end{aligned}$$

$$0 \leq f_{xy} \leq c_{xy} \quad \forall xy \in E$$

(A.1)

In a general routing problem, there are no constraints on how the flows can be distributed between the paths. However, in OSPF routing, the routing of the traffic demands is determined only by the shortest paths. The shortest paths in turn are determined by the weights chosen for the links. The additional constraint of routing through only the shortest paths transforms the general routing problem into a NP-hard problem. In [4], the following hardness result of optimal setting of OSPF weights has been proved.

Theorem 1. *It is NP-hard to optimize the maximum link utilization in OSPF routing.*

The hardness result has been shown by reducing 3SAT to the problem of optimizing OSPF weight setting with respect to maximum link utilization. Refer [4] for the detailed proof. We made the following observation from the construction of the OSPF weight setting from 3SAT problem.

Let $S = (X, C)$ be an instance of 3SAT with variable set X and clause set C where each clause has three literals. There are several steps involved in the reduction process. One of the steps is to derive the traffic demand. There is a source-destination pair (s, t) associated with each $x \in X$. The demand between (s, t) is set to $2|x|$, where $|x|$ denotes the least power of two bounding both the number of negative and the number of positive occurrences of x in S . It is possible that the traffic demands between all source-destination pairs could be the same. More precisely, the problem of optimal setting of OSPF weights is NP-hard even when the traffic demands are uniform. This is also *true* when the capacity of each link is same. Refer [35] for the hardness result of the OSPF Flow Allocation Problem (FAP) in which the demands are considered uniform. We use this observation in the following proof.

Theorem 2 (CEM is NP-hard). *It is NP-hard to maximize the entropy of the shortest path betweenness centrality distribution.*

Proof. We prove this result by reducing the problem of optimizing OSPF weight setting with respect to maximum link utilization to the problem of maximizing the entropy of the shortest path betweenness centrality distribution. We construct an instance of the CEM problem as follows.

The objective function, minimizing the maximum link utilization, is a convex function [36]. $p \log p$ is a convex function on \mathbb{R}_+ , assuming $0 \log 0 = 0$. So, the negative entropy of the centrality distribution, $\sum_{uv \in E} p(u, v) \log p(u, v)$ is convex. The objective function can be restated to minimize the negative entropy of the centrality distribution. So, both problems are instances of *convex optimization* problem.

We assume that the traffic demands and the capacity of links are uniform. As stated earlier, f_{uv} denotes the flow on link $(u, v) \in E$ realized by the chosen routing protocol. In the OSPF routing, for each source-destination pair $(s, t) \in V \times V$, and for each node $u \in V$, we have that $f_{uv}^{st} = 0$ if (u, v) is not on a shortest path from s to t , and that $f_{uv}^{st} = f_{uv'}^{st}$ if both (u, v) and (u, v') are on shortest paths from s to t . The total flow f_{uv} is the sum of individual flows with each flow being associated with a source-destination pair.

$$f_{uv} = \sum_{\forall (st) \in V \times V} f_{uv}^{st} \quad (\text{A.2})$$

This individual source-destination pair flow f_{uv}^{st} is again the sum of path-specific individual flows. Let Q_{st} denote the set of the shortest paths from s to t that need not be edge-disjoint. Then, we have

$$f_{uv}^{st} = \sum_{Q_{st}} f_{uv}^{st(q_i)} \quad (\text{A.3})$$

where q_i denotes the i -th path in Q_{st} , $1 \leq i \leq |Q_{st}|$. From the above equation, $f_{uv}^{st(q_i)}$ denotes the amount of flow on the link $(u,$

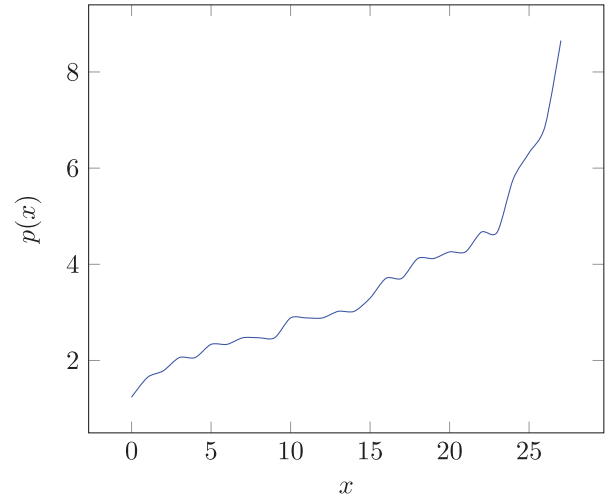


Fig. B.6. Abilene Network - probabilities of links based on shortest path betweenness centrality values of network links; X axis represents the links, and Y axis represents the probability associated with each link. The probabilities are sorted in ascending order.

$v)$ that appears in the shortest path $q_i \in Q_{st}$ from s to t . It is possible that more than one shortest path in Q_{st} pass through the link (u, v) .

For each flow variable $f_{uv}^{st(q_i)}$, we associate a variable $b_{uv}^{st(q_i)}$ such that

$$b_{uv}^{st(q_i)} = \begin{cases} 1 & \text{if } f_{uv}^{st(q_i)} > 0 \\ 0 & \text{if } f_{uv}^{st(q_i)} = 0 \end{cases}$$

This variable helps us determine if the link is in a shortest path from s to t . Now the shortest path betweenness centrality of a link (u, v) can be derived as follows.

$$\sigma_{\bullet, \bullet}(u, v) = \sum_{s, t} \sigma_{s, t} = \sum_{s, t} \sum_{Q_{st}} b_{uv}^{st(q_i)} \quad (\text{A.4})$$

When the traffic demands and the capacity are uniform, the SPBC of a link is directly proportional to the flow assigned by the OSPF routing. More precisely, the probability value associated with a link (u, v) is directly proportional to the flow realized by the OSPF routing. So, any modification to the weight of a link will have the same effect on the flow realized by the OSPF routing as well as the SPBC distribution. More precisely, it is the change induced by the weight setting in the betweenness centrality of a link which causes the change in the flow on the link.

Hence, if we are able to find an optimal weight setting for entropy maximization of SPBC distribution in polynomial time, then we could determine optimal weight setting for the OSPF routing with respect to maximum link utilization in polynomial time. But it has been proved not possible. This concludes our proof on hardness of the CEM problem. \square

Appendix B. Tactical networks topology design

In this section, we evaluate the proposed network centrality measure in the context of the shortest path betweenness centrality on some select network topologies and interpret the results.

A tactical communication system (TCS) consists of a number of communication nodes connected in a partial mesh topology over the operational area Fig. B.6. In a mesh architecture, a full mesh is a complete graph where each node is *directly* connected to every other node, and a partial mesh is a network topology where each node is connected to select neighbors with a point-to-point radio

Table B.8
Centrality Distribution Δ_G for the Abilene Network, probabilities rounded to 3 decimal places.

(u, v)	$\sigma_{..uv}$	$p(u, v)$	(u, v)	$\sigma_{..uv}$	$p(u, v)$
(0, 3)	32	0.041	(0, 4)	24	0.033
(0, 10)	50	0.069	(1, 4)	15	0.021
(1, 7)	18	0.025	(2, 5)	31	0.043
(2, 8)	18	0.025	(2, 9)	21	0.029
(3, 0)	42	0.058	(3, 5)	21	0.029
(3, 6)	34	0.047	(4, 0)	27	0.037
(4, 1)	63	0.087	(4, 5)	27	0.037
(5, 2)	31	0.046	(5, 3)	30	0.041
(5, 4)	46	0.063	(6, 3)	21	0.029
(6, 9)	22	0.030	(7, 1)	18	0.025
(7, 10)	15	0.021	(8, 2)	12	0.016
(8, 9)	17	0.023	(9, 2)	9	0.012
(9, 6)	34	0.047	(9, 8)	13	0.018
(10, 0)	17	0.023	(10, 7)	22	0.030

link. The full mesh is too costly and not practical in tactical networks. A partial mesh provides the required redundancy of routes between any pair of nodes. The nodes are classified into trunk nodes and access nodes. Trunk nodes are connected by high capacity point-to-point radio links, and form backbone network. The access nodes are connected to trunk nodes, and capable of providing connectivity to several mobile users. When a node moves from one point to another point in the operational area, it discovers its neighbors and establishes point-to-point link to them. The network topology formation is highly dynamic. The battle conditions, terrain and disposition of the adversaries determine the position of network elements.

In our evaluation, we deal with the topology structure of the network, and use our proposed measure to determine the maximally efficient topology. In particular, we are interested in determining the centralization or decentralization of the entire network in the context of the shortest paths available in a graph. We modeled a sample tactical network based on the Abilene network topology as shown in Fig. 1. For the purpose of illustration, we assume that each edge has unit weight. The trunk nodes are labeled from 0 to 10. These trunk nodes have multiple access nodes, and the corresponding access nodes are mentioned as dots near each trunk node.

Entropy of SPBC

Consider the network topology given in Fig. 1. We use AMPL for computing the shortest paths, betweenness centrality values, and calculating the entropy values. An efficient algorithm for computing betweenness centrality has been proposed in [13]. Table B.8 shows the number of shortest paths that pass through each link and its associated probability value. For example, the link (0, 3) has 32 shortest paths passing through it, and its associated probability value is 0.041209. The total number of the shortest paths in the entire network $\sigma_{..}$ is 404.

The probability distribution of the Abilene network is shown in Fig. B.6.

From Table B.8, we see that the link (4, 1) has the maximum number of the shortest paths passing through it. It is also due to the fact that the trunk node 1 has the maximum number of access nodes. The entropy of SPBC distribution for the given topology $H_{28}(\Delta_G)$ is 4.662206. The information distance is: $\log(28) - H_{28}(\Delta_G) = 0.145149$. The relative entropy $h_G = 0.969807$. This implies that the centrality distribution is closer to maximum entropy, and hence it is a decentralized network. We find the trunk node 1 has 12 access nodes, and redistributing 5 nodes to the trunk node 4 has shown increase in entropy: $H_{28}(\Delta_G) = 4.686911$ and $h_G = 0.974946$. This implies that the distribution of the shortest paths among the links has improved.

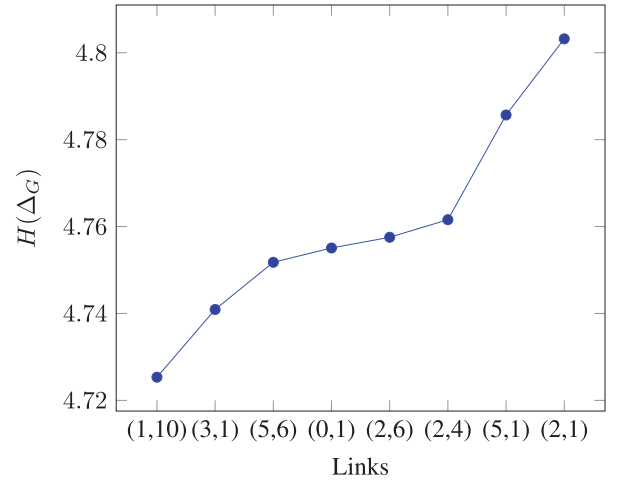


Fig. B.7. Entropy values when new links are created; The entropy increases when a new link (2,1) is added to the network.

Adding links

We now consider the case of adding a new point-to-point radio link between two arbitrary nodes. We compute the entropy and relative entropy for a few cases as shown in Fig. B.7, and illustrate how our proposed measure helps in validating such design decisions. Fig. B.7 shows only cases wherein the entropy increases due to addition of new links. For example, when a new link (1, 10) is added to the network given in Fig. 1, the entropy has increased to 4.725330 and subsequently the relative entropy has increased to 0.962999. From Fig. B.7, we can see that adding a new link (2, 1) is preferred over other links as it results in higher entropy value.

Topology changes

In tactical networks, when a node moves from one point to another point in the operational area, it discovers its neighbors and establishes point-to-point links to them. Our proposed network centrality measure based on betweenness centrality helps in identifying the right neighbors that provide maximum efficiency in routing. We propose an algorithm *Neighbor Discovery* given in Algorithm 2 to determine the neighbors from the set of reachable

Algorithm 2 Neighbor Discovery

Input $G = (V, E), A \subset V$ {A is the set of neighbors}

Input k {Number of links in the node}

Output G_{MAX} {A graph with maximum entropy}

$u \leftarrow$ new node to join the network and $m \leftarrow |A|$

for all k combinations in m **do**

Let $\{i_1, i_2, \dots, i_k\}$ be the choice of k neighbors

$V' \leftarrow V \cup \{u\}, E' \leftarrow E \cup \{(u, i_1), \dots, (u, i_k)\}, G' = (V', E')$

Compute W_k^\uparrow and W_k^\downarrow {bidirectional weights}

Construct all pair $(s, t), s, t \in V'$ shortest paths

Compute $\sigma_{..}$ and $p(u, v)$

Compute $H(\Delta_{G'})$ and $h_{G'}$

$G_{MAX} \leftarrow G'$ if $H(\Delta_{G'})$ and $h_{G'}$ are increased

end for

G_{MAX} is maximum entropy network

nodes based on the entropy values. Our objective is to connect to the appropriate neighbors that increase the entropy of SPBC distribution. The algorithm runs through all possible combinations on the choice of neighbors with appropriate weights on the bidirectional links. For each combination, it computes the entropy and

relative entropy to measure the efficiency of the new network. The algorithm finally produces the network topology that has the maximum entropy. This is useful in addressing a class of network synthesis problems where we need to determine the links or capacities on links at minimum cost in order to satisfy the traffic demands.

The time complexity to compute the SPBC distribution for a given network is in $O(nm + n^2 \log n)$, where n is the total number of nodes and m is the total number of links in the network [13]. Under practical conditions, the number of links and the number of neighbor nodes are very small. Also, a centralized controller can precompute the entropy for some of the possibilities based on tactical operations, and pair them based on higher entropy.

Diversity index

In tactical networks, the traffic demand between a source-destination pair changes dynamically. It is important to assess the routing sub graph between the chosen pair of nodes so that there is no unexpected congestion or delays. For example, the overall entropy of a tactical network may be higher, but there could be a *cut edge* between a given source-destination pair. This may lead to a single point of failure. We can use the diversity index between source-destination pairs to analyze such cases. In the above network, the minimum diversity index $h_{1,24} = 0.951796$. Note that one of these measures or a combination of these measures can be used to evaluate or design the network topology based on the stated design goals. For example, one may want to maximize the diversity index between a select source-destination pair while maintaining the overall network centrality at a certain value.

References

- [1] A. Retana, D. Slice, R. White, *Advanced IP network design (CCIE Professional Development)*, vol. 17, Cisco Press, 1999.
- [2] A. Mowshowitz, M. Dehmer, Entropy and the complexity of graphs revisited., *Entropy* 14 (3) (2012) 559–570.
- [3] D. Xu, M. Chiang, J. Rexford, Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering, *IEEE/ACM Trans. Netw.* 19 (6) (2011) 1717–1730.
- [4] B. Fortz, M. Thorup, Increasing Internet capacity using local search, *Comput. Optim. Appl.* 29 (2004) 13–48.
- [5] V. Chellappan, K.M. Sivalingam, Application of entropy of centrality measures to routing in tactical wireless networks, in: *IEEE Workshop on Local Metropolitan Area Networks*, 2013, pp. 1–6.
- [6] V. Chellappan, K.M. Sivalingam, K. Krithivasan, An entropy maximization problem in shortest path routing networks, in: *IEEE Workshop on Local Metropolitan Area Networks*, 2014, pp. 1–6.
- [7] L.C. Freeman, A set of measures of centrality based on betweenness, *Sociometry* 40 (1) (1977) 35–41.
- [8] D.R. White, S.P. Borgatti, Betweenness centrality measures for directed graphs, *Soc. Netw.* 16 (4) (1994) 335–346.
- [9] Y. Shavitt, Y. Singer, Beyond centrality - classifying topological significance using backup efficiency and alternative paths, *New J. Phys.* 9 (8) (2007) 266:1–266:17.
- [10] S.P. Borgatti, Centrality and network flow, *Soc. Netw.* 27 (1) (2005) 55–71.
- [11] S. Dolev, Y. Elovici, R. Puzis, Routing betweenness centrality, *J. ACM* 57 (4) (2010) 25:1–25:27.
- [12] E.C. Juan Segovia, P. Vilá, New applications of the betweenness centrality concept to reliability-driven routing, in: *Proceedings of the Workshop in G/MPLS Networks*, 2009.
- [13] U. Brandes, A faster algorithm for betweenness centrality, *J. Math. Sociol.* 25 (2001) 163–177.
- [14] M.J. Newman, A measure of betweenness centrality based on random walks, *Soc. Netw.* 27 (1) (2005) 39–54.
- [15] A.H. Dekker, B.D. Colbert, Network robustness and graph topology, in: *Proceedings of the 27th Australasian Conference on Computer Science*, 2004, pp. 359–368.
- [16] C.E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.* 27 (1948) 379–423.623–656
- [17] E.T. Jaynes, Information theory and statistical Mechanics, *Phys. Rev.* 106, 108 (1995) 620–630.171–197
- [18] J.N. Kapur, *Maximum-Entropy Models in Science and Engineering*, Wiley, Wiley Eastern Limited, Chichester, 1989.
- [19] J.A. Tomlin, S.G. Tomlin, Traffic distribution and entropy, *Nature* 220 (1968) 974–976.
- [20] A.K. Agarwal, D. Mohan, R.S. Singh, Traffic planning in a constrained network using entropy maximisation approach, *J. IEI* 85 (2005) 236–240.
- [21] L.C. Freeman, Centrality in social networks: Conceptual clarification, *Soc. Netw.* 1 (1979) 215–239.
- [22] B. Fortz, M. Thorup, Internet traffic engineering by optimizing OSPF weights, in: *Proceedings of the IEEE INFOCOM*, 2000, pp. 519–528.
- [23] D. Braess, A. Nagurny, T. Wakolbinger, On a paradox of traffic planning, *Transport. Sci.* 39 (4) (2005) 446–450.
- [24] K.G. Ramakrishnan, M.A. Rodrigues, Optimal routing in shortest-path data networks, *Bell Labs Tech. J.* 6 (1) (2001) 117–138.
- [25] S. Nelakuditi, Z. Li Zhang, On selection of paths for multipath routing, in: *Proceedings of IWQoS*, 2001, pp. 170–186.
- [26] B. Zhang, D. Massey, L. Zhang, Destination reachability and BGP convergence time, in: *Proceedings of the IEEE GLOBECOM*, 3, 2004, pp. 1383–1389.
- [27] K. Conrad, Probability distributions and maximum entropy is notes from Keith Conrad, and available online at: <http://www.math.uconn.edu/~kconrad/blurb/analysis/entropypost.pdf>.
- [28] M.H. Sqalli, S.M. Sait, S. Asadullah, OSPF weight setting optimization for single link failures, *Int. J. Comput. Netw. Commun.* 3 (1) (2011) 168–183.
- [29] V.K. Gurbani, M. Scharf, T.V. Lakshman, V. Hilt, E. Marocco, Abstracting network state in Software Defined Networks (SDN) for rendezvous services, in: *Proceedings of IEEE ICC*, 2012, pp. 6627–6632.
- [30] B. Lantz, B. Heller, N. McKeown, A network in a laptop: rapid prototyping for software-defined networks, in: *Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2010, pp. 19:1–19:6.
- [31] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, *ACM SIGCOMM Comput. Commun. Rev.* 38 (2) (2008) 69–74.
- [32] A. Tizghadam, A. Leon-Garcia, AORTA: autonomic network control and management system, in: *Proceedings of the IEEE Conference on Computer Communications Workshops INFOCOM*, 2008, pp. 1–4.
- [33] V. Matossian, M. Parashar, Towards autonomic control of network topologies, in: *Proceedings of the IEEE workshop on Modeling Autonomic Communication Environments (MACE)*, 2006, pp. 235–250.
- [34] D.O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, Overview and Principles of Internet Traffic Engineering, RFC 3272, IETF, 2002.
- [35] M. Pióro, A. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, S. Kozdrowski, On OSPF related network optimisation problems, *Perform. Eval.* 48 (1/4) (2002) 201–223.
- [36] D. Medhi, K. Ramasamy, *Network Routing - Algorithms, Protocols, and Architectures.*, Morgan Kaufmann, San Francisco, CA, USA, 2007.



Vanniyarajan Chellappan is consultant Principal Engineer at LinkBee Inc. USA. He was the CTO and one of the co-founders of Helyxon Healthcare Solutions. He has a Ph.D. from the Indian Institute of Technology (IIT) Madras in the field of Computer Science. From 1995 to 2014, he has held various positions with M/s HCL Technologies Ltd at Chennai, most recently as General Manager. He has extensive expertise in communications and networking technologies, specifically routing protocols, high-speed packet classification algorithms, traffic engineering, high-end router architecture, etc. He has co-authored several papers in IEEE publications, and has filed for grant of patents for some his ideas. He won Cisco's Pioneer Technology Innovation Award 2002 (Team). He was also a Sr. Project Officer, India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services from DST Gol, working under the guidance of Prof. Krishna M. Sivalingam in CSE Department, IIT Madras, India.



Krishna M. Sivalingam Krishna M. Sivalingam is a Professor in the Department of CSE, IIT Madras, Chennai, INDIA. Previously, he was a Professor in the Dept. of CSEE at University of Maryland, Baltimore County, Maryland, USA from 2002 until 2007; with the School of EECS at Washington State University, Pullman, USA from 1997 until 2002; and with the University of North Carolina Greensboro, USA from 1994 until 1997. He has also conducted research at Lucent Technologies' Bell Labs in Murray Hill, NJ, and at AT&T Labs in Whippany, NJ. He received his Ph.D. and M.S. degrees in Computer Science from State University of New York at Buffalo in 1994 and 1990 respectively; and his B.E. degree in Computer Science and Engineering in 1988 from Anna University's College of Engineering Guindy, Chennai (Madras), India. While at SUNY Buffalo, he was a Presidential Fellow from 1988 to 1991. His research interests include wireless networks, wireless sensor networks, optical wavelength division multiplexed networks, and performance evaluation. His work has been supported by several sources including AFOSR, DST India, IBM, NSF, Cisco, Intel, Tata Power Company and Laboratory for Telecommunication Sciences. He holds three patents in wireless networks and has published several research articles including more than fifty journal publications. He has co-edited a book on Next Generation Internet Technologies in 2010; on Wireless Sensor Networks in 2004; on optical WDM networks in 2000 and 2004. He is serving or has served as a member of the Editorial Board for journals including ACM Wireless Networks Journal, IEEE Transactions on Mobile Computing, and Elsevier Optical Switching and Networking Journal. He is presently serving as Editor-in-Chief of Springer Photonic Network Communications Journal and EAI Endorsed Transactions on Ubiquitous Environments. He serves on the Steering Committee of IEEE Advanced Networks and Telecommunications Symposium (ANTS) and EAI/ICST MobiQuitous conferences. He is a Fellow of IEEE, a Fellow of INAE and an ACM Distinguished Scientist.



Kamala Krithivasan received her Ph.D. from the University of Madras, and she joined the Indian Institute of Technology Madras (IITM) in 1975. She has more than 30 years of teaching and research experience at IITM, in which she served as Chairperson during 1992–1995. Her research interests include formal language theory and unconventional models of computing like DNA computing, membrane computing and discrete tomography. A recipient of the Fulbright fellowship in 1986, Professor Kamala is also a fellow of the Indian National Academy of Engineering.