Robust Network Planning in Nonuniform Traffic Scenarios

Ali Tizghadam, Alberto Leon-Garcia

University of Toronto

Abstract

Robustness to the environmental variations is an important feature of any reliable communication network. This paper reports on a network theory approach to the design of such networks where the environmental changes are traffic fluctuations, topology modifications, and changes in the source of external traffic. Motivated by the definition of betweenness centrality in network science, we define the notion of traffic-aware betweenness (TAB) for data networks, where usually an explicit (or implicit) traffic matrix governs the distribution of external traffic into the network. We use the average normalized traffic-aware betweenness, which is referred to as traffic-aware network criticality (TANC), as our main metric to quantify the robustness of a network. We show that TANC is directly related to some important network performance metrics, such as average network utilization and average network cost. We prove that TANC is a linear function of end-to-end effective resistances of the graph. As a result, TANC is a convex function of link weights and can be minimized using convex optimization techniques. We use semi-definite programming method to study the properties of the optimization problem and derive useful results to be employed for network planning purposes.

Keywords:

1. Introduction

According to Darwin's theory of natural selection, each slight variation, if useful, is preserved. As a result of natural selection, every process receives a survival value representing its overall sensitivity or robustness to the external variations. Our goal in this paper is to design an appropriate survival value for communication networks. The survival value quantifies the adaptability of a network to unexpected changes in environmental parameters.

In Darwin's theory there is no "final target" for the evolutionary changes in the nature, however, viewing survival as the goal can lead to an implicit optimization problem, where the first goal of the optimization is to keep the system alive under unforeseen circumstances.

Since a reliable communication takes place only in a connected network, any metric for survival value should capture the effect of network connectivity. Moreover, a robust network algorithm should force the network to evolve in a way that maximizes the probability of future connectivity. This implies that the optimization must address the real-time efficiency and performance of the whole network as a short-term goal, while striving to maintain and improve the survival value of the network as a long-term goal. Response to nonuniform traffic shifts including the effect of modifications in the sources and sinks of traffic is another key factor in defining an appropriate survival value for communication networks. The present paper tries to find such survival value and investigate its main attributes.

In summary, one of the key properties of reliable communication networks is the robustness to the unexpected environmental changes. To be more specific, in this paper we consider three types of environmental changes:

1. Modifications in network topology such as node failure or variations in link capacities.
2. Changes in active sources or sinks for external traffic, which we refer to it as community of interest.
3. Traffic demand shifts (nonunifom).

Throughout, a network (or a network algorithm) is robust if its performance is insensitive to the above mentioned environmental changes to the extent possible. This infers that a robust network is more reliable, since unpredicted environmental changes cannot dramatically impact its performance. In this paper, we intend to propose methods to design such robust networks using the concept of betweenness centrality and resistance distance from graph theory . We try to find a metric for robustness (as the survival value for data networks) to capture and quantify the effect of topology, community of interest, and nonuniform traffic. The main contribution of this work (which makes it different from previous ones) is that we take into account the effect of a given external set of traffic demands in the definition of the survival value. We use this survival value to design a network in such a way that the average sensitivity to the environmental changes is minimized.

The paper is organized as follows. Section 2 provides a summary of related works in literature. Section 3 reviews previous work on random-walk betweenness and network criticality, and introduces necessary notations. The proposed metrics traffic-aware betweenness and traffic-aware network criticality are introduced and investigated in section 4. Section 5 investigates the special case of uniform traffic scenarios and proposes some important interpretations of the network criticality. In section 6 the relationship between network utilization and traffic-aware network criticality is discovered. The optimization problem to minimize traffic-aware network criticality is investigated in section 7. Section 8 discusses the application of the proposed optimization problem in network planning for different traffic scenarios. A comprehensive evaluation of the network planning method is provided in section 9. In section 10 we discuss the main limitation of the proposed method. Conclusions are presented in section 11.

2. Previous Work

There is a wealth of literature focusing on different aspects of network robustness. [Dekker et al. (2004)] considers node similarity and optimal connectivity as main robustness metrics and arrives at the result that a node-similar and optimally connected network provides maximum resistance to node failure. The paper discusses some methods to design node-similar and optimally connected networks. The main focus of [Dekker et al. (2004)] in on topological aspects of robustness.

[Zhang-Shen et al. (2005)] presents a novel approach to design backbone networks based on the concept of Valiant load-balancing, which is insensitive to the traffic matrix (i.e. it works equally well for all valid traffic matrices), and provides guaranteed performance under a predefined number of interface and router failures. The main idea is to forward the traffic destined for a sink d to intermediate hops with equal probability, and then route the flow to the destination d. Delay propagation is one of the disadvantages of the load-balancing method proposed in [Zhang-Shen et al. (2005)].

In graph theory literature, Freeman [Freeman (1978)] first introduced shortest-path betweenness to measure the centrality (importance) of a node (or link) in a graph. Shortest-path betweenness of node k for trajectories from source node i to destination node j is defined as the proportion of instances of the shortest paths from node i to j which include node k. The overall shortest-path betweenness centrality of node k is the sum of the betweennesses over all source-destination pairs. Link betweenness is defined likewise. Due to the fundamental limitation of shortest-path betweenness (shortest path considers only a small subset of available paths between every pair of nodes), a series of other related metrics defined in the field of network science, including flow-betweenness [Freeman et al. (1991); Borgatti (2005)] and random-walk betweenness [Newman (2003)].

Using the concept of betweenness, [Tizghadam et al. (2007)] introduced a framework for robust routing in core networks based on the idea of "link criticality" and "path criticality". In [Tizghadam et al. (2008)] we presented an analysis of betweenness centrality, and provided a framework to study network robustness. The concept of *Network Criticality* is investigated in [Tizghadam et al. (2009)], where a mathematical framework for the definition of criticality is proposed within the context of Markov chain theory. [Tizghadam et al. (2010)] further elaborates on the concept of network criticality and focuses on designing autonomic control loops and algorithms for core networks. In [Tizghadam et al. 2 (2009)] we have shown that "network criticality" is an appropriate metric to be used as survival value in uniform traffic scenarios, and in [Tizghadam et al. 3 (2009)] we presented a semi-definite approach to optimize this survival value.

In this paper we extend the idea of survival value to the nonuniform traffic case. Previous definitions of betweenness (shortest-path betweenness, flow-betweenness, random-walk betweenness) are purely topological and do not take into account the effect of nonuniform traffic between different source-destination pairs, while in Internet and other communication networks the external traffic is a major factor in analyzing the behavior of networks. Here we introduce a new notion of betweenness, Traffic-Aware Betweenness (TAB), to account for the mutual effect of topology and traffic in a network. This new definition can be applied to all different versions of betweenness, but our derivations are for traffic-aware random-walk betweenness (TARWB). We assume that the traffic between every node pair is given by a traffic matrix $\Gamma = [\gamma_s(d)]$. We will extend the concept of network criticality from [Tizghadam et al. (2010)] to the traffic-aware network criticality (TANC). We will show that some critical features and performance metrics of real networks are directly related to the TANC. We consider TANC as the survival value and study its robustness properties. TANC can be thought as a generalization of network criticality, our previous candidate for survival value. As a matter of fact, we will show that in uniform traffic scenarios, TANC is reduced to network criticality.

3. Review of Random-Walk Betweenness and Network Criticality

This section briefly summarizes previous findings on random-walk betweenness and network criticality. In graph theory, the concept of centrality (and in particular betweenness centrality) is used to locate critical nodes and links of a graph which have significant impact on the performance of the network. Our interest in this paper is on probabilistic version of betweenness centrality which is the basis for defining network criticality. While betweenness centrality depends only on the topology of the graph, network criticality tries to involve other environmental parameters affecting the performance of a network. This section briefly reviews the foundations of betweenness and network criticality.

In this paper we model a network by graph $G(N, E, W)$, where N, E are the set of nodes and links respectively, and W is the matrix of link weights.

3.1. Random-Walk Betweenness

Our work is motivated by Newman's probabilistic view on betweenness centrality [Newman (2003)]. We define a finite-state irreducible Markov Chain on the network, where the states of the Markov chain are the graph nodes and the edges correspond to permissible transitions, and labels associated with the edges denote the transition probabilities p_{ij} of moving from state i at time t to neighbor j at time $t+1$ (discrete time). The Markov chain can be viewed as a random walk on the graph with next-step transition probabilities p_{ij}.

We intend to quantify the betweenness of a node in the random-walk corresponding to the above Markov chain. Consider the set of trajectories starting at node s and terminating when the walk first arrives at node d, that is, destination node d is an absorbing node. The random-walk betweenness $b_{sk}(d)$ of node k for the $s-d$ trajectories is defined as the average number of times node k is visited in trajectories from s to d. Note that $b_{dk}(d) = 0$ for

$k \neq d$ as such walks are terminated at step zero. Also note that $b_{sd}(d) = 1$ ($s \neq d$) since the walk terminates exactly with the occurrence of the first arrival to node d.

Let $B_d = [b_{sk}(d)]$ denote the n-by-n matrix of betweenness metrics when d is an absorbing node (destination). Clearly the d^{th} row of the matrix is zero. Betweenness matrix B_d can be written as [Tizghadam et al. (2009)]:

$$B_d = (I - P_d)^{-1}\Theta_d \tag{1}$$

$$\Theta_d = [\theta_{sk}(d)] = \begin{cases} 1 & if\ s = k \neq d \\ 0 & otherwise \end{cases}$$

Matrix P_d equals P except that the entries of its d^{th} row and d^{th} column are set to zero. In this paper we are interested in a special type of random-walks referred to as weight-based or generic random-walk, where the probability of transitioning along a link is proportional to the weight of the link. More specifically:

$$p_{sk}(d) = \frac{w_{sk}}{\sum_{q \in A(s)} w_{sq}}(1 - \delta_{sd}) \tag{2}$$

where $A(s)$ is the set of adjacent nodes of s and w_{sk} is the weight of link (s, k) (if there is no link between node s and k, then $w_{sk} = 0$), and δ_{sd} is the Kronecker delta function (i.e. if $s = d$, then $\delta_{sd} = 1$, otherwise $\delta_{sd} = 0$). The delta function in equation (2) is due to the fact that the destination node d is an absorbing node, and any random-walk coming to this state, will be absorbed or equivalently $p_{dk}(d) = 0$. Clearly, equation (2) defines a Markovian random-walk.

3.2. Network Criticality or Total Resistance Distance

This section reviews *network criticality* [Tizghadam et al. (2009)] which is a metric to quantify the robustness of a network in uniform traffic cases, where the average traffic demands between all active source-destination pairs are the same (entries of the traffic matrix are equal). We first define node/link criticality.

Definition 3.1. *Node criticality is defined as the random-walk betweenness of a node normalized by the node weight (weight of a node is defined as the sum of the weights of its incident links). Link criticality is defined similarly.*

Node (link) criticality can be written in terms of generalized inverse of the graph Laplacian matrix. Let η_k denote the criticality of node k and

η_{ij} denote the criticality of link $l = (i, j)$, then we have [Tizghadam et al. (2009)]:

$$\begin{aligned}
\frac{b_{sk}(d)}{W_k} &= l^+_{dd} - l^+_{sd} - l^+_{dk} + l^+_{sk} && (3)\\
\tau_{sd} &= l^+_{ss} + l^+_{dd} - 2l^+_{sd} \;\; or \;\; \tau_{sd} = u^t_{sd} L^+ u_{sd} && (4)\\
\tau_{sd} &= \frac{b_{sk}(d) + b_{dk}(s)}{W_k} && (5)\\
\eta_k &= \frac{b_k}{W_k} = \frac{1}{2}\tau \;\; , \quad \tau = \sum_s \sum_d \tau_{sd} && (6)\\
\eta_{ij} &= \frac{b_{ij}}{w_{ij}} = \tau && (7)
\end{aligned}$$

where L^+ is the Moore-Penrose inverse of graph Laplacian matrix L [Rao et al. (1971)], n is the number of nodes, and $u_{ij} = [0 \ldots 1 \ldots -1 \ldots 0]^t$ (1 and -1 are in i^{th} and j^{th} positions respectively). We define the average (or normalized) network criticality as $\hat{\tau} = \frac{1}{n(n-1)}\tau$.

Observation 3.2. *According to equations (4) to (7), node criticality (η_k) and link criticality (η_{ij}) are independent of the node/link position and only depend on τ (or $\bar{\tau}$) which is a global quantity of the network.*

Definition 3.3. *We refer to τ_{sd} as* point-to-point network criticality *and τ as* network criticality.

Point-to-point network criticality has a nice interpretation in electrical circuits. Suppose we build a resistive electrical network, where conductance of a link in the resistive electrical circuit equals the weight of the corresponding link in original network, then τ_{sd} is numerically the same as *resistance distance* or *effective resistance* observed between two end points s and d [Klein et al. (1993)], and τ is the total resistance distance of the network with many useful interpretations [Ghosh et al. (2008)].

Based on equations (6) and (7), the node (link) betweenness consists of a local parameter (weight) and a global metric (network criticality). τ can capture the effect of topology through the betweenness values, where a high value for the betweenness of a node/link means that there is a high risk (criticality) in using the node/link. Moreover, if we consider the capacity of a link as its weight, then the higher the weight of a node/link, the lower the risk of using the node/link. Hence, network criticality can quantify the risk of using a node/link in a network which is an indication the degree of robustness.

4. Traffic-Aware Betweenness

In this paper we extend the definition of betweenness, and network criticality by considering the effect of an explicit nonuniform traffic matrix in the system. In our previous works ([Tizghadam et al. (2009); Tizghadam et al. 2 (2009)]) we implicitly assumed that the average input traffic to all the nodes of the network are uniform. In this work we consider a general traffic matrix $[\gamma_s(d)]$ and will derive a generalized expression for network criticality to account for the effect of traffic matrix. We start by developing an expression for traffic-aware betweenness in the present section. Generally speaking, the traffic-aware betweenness centrality is defined as follows.

Definition 4.1. *Let $\Gamma = [\gamma_s(d)]$ and γ denote the traffic matrix and the total external traffic (i.e. $\gamma = \sum_{s,d} \gamma_s(d)$) respectively. We define traffic-aware betweenness (TAB) of node k as:*

$$
\begin{aligned}
b'_{sk}(d) &= b_{sk}(d) + \frac{\gamma_s(d)}{\gamma} b_{sk}(d) \\
b'_{sk}(d) &= \beta_{sd} b_{sk}(d), \quad \beta_{sd} = 1 + \frac{\gamma_s(d)}{\gamma} && (8) \\
b'_k &= \sum_{s,d} \beta_{sd} b_{sk}(d) && (9)
\end{aligned}
$$

The main motivation behind equation (8) is the fact that when nonuniform external traffic exists, we would like to see its explicit effect, however, we want to recover the topological definition of betweenness when there is no traffic, or at the presense of uniform traffic (i.e. $\gamma_s(d)$ is zero or fixed for all possible node pairs $s \neq d$). Definition 4.1 reduces to the topological definition of betweenness when there is no traffic, or when we have uniform traffic.

Definition 4.1 is generally applicable for different types of betweenness centralities. Considering $b_{sk}(d)$ as the shortest-path betweenness, definition

4.1 gives traffic-aware shortest-path betweenness, and so on. However, in this paper our focus is on random-walk betweenness centrality. In the following we provide an expression for traffic-aware random-walk betweenness.

4.1. Traffic-Aware Random-Walk Betweenness (TARWB)

In this section we develop traffic-aware random-walk betweenness, where $b_{sk}(d)$ denotes the random-walk betweenness (weight-based random-walk). We try to find an expression for TARWB based on point-to-point network criticalities or resistance distances (equation (4)). To this end we notice that:

$$\begin{aligned} b'_k &= \sum_{s,d} \beta_{sd} b_{sk}(d) \\ &= \frac{1}{2} \sum_{s,d} (\beta_{sd} b_{sk}(d) + \beta_{ds} b_{dk}(s)) \end{aligned} \tag{10}$$

But, from equation (5) we know that:

$$b_{sk}(d) + b_{dk}(s) = W_k \tau_{sd} \tag{11}$$

Substituting equation (11) in (10) will result in:

$$b'_k = \frac{W_k}{2} \sum_{s,d} \beta_{ds} \tau_{sd} + \frac{1}{2} \sum_{s,d} (\beta_{sd} - \beta_{ds}) b_{sk}(d) \tag{12}$$

Now we write $b_{sk}(d)$ in terms of point-to-point network criticalities. Considering equation (3) and (4), it is easy to verify that:

$$\tau_{sd} + \tau_{dk} - \tau_{sk} = 2(l^+_{dd} - l^+_{sd} - l^+_{dk} + l^+_{sk}) = 2\frac{b_{sk}(d)}{W_k}$$

Therefore

$$b_{sk}(d) = \frac{W_k}{2}(\tau_{sd} + \tau_{dk} - \tau_{sk}) \tag{13}$$

Using equation (13) in (12), we will have

$$b'_k = \frac{W_k}{4} \sum_{s,d} (\beta_{sd} + \beta_{ds}) \tau_{sd} + \frac{W_k}{4} \sum_{s,d} (\beta_{sd} - \beta_{ds})(\tau_{dk} - \tau_{sk}) \tag{14}$$

Equation (14) can be written in the following form:

$$\frac{b'_k}{W_k} = \frac{1}{2}(\frac{1}{2}\sum_{s,d}(\beta_{sd}+\beta_{ds})\tau_{sd}+\frac{1}{2}\sum_{s,d}(\beta_{sd}-\beta_{ds})(\tau_{dk}-\tau_{sk})) \quad (15)$$

In analogy with the notion of $\frac{b_k}{W_k}=\frac{\tau}{2}$ and using equation (15) we can define Traffic-Aware Node Criticality (TANOC) τ'_k:

$$\tau'_k = \frac{1}{2}\sum_{s,d}(\beta_{sd}+\beta_{ds})\tau_{sd}+\frac{1}{2}\sum_{s,d}(\beta_{sd}-\beta_{ds})(\tau_{dk}-\tau_{sk}) \quad (16)$$

We can also define traffic-aware criticality between two nodes as follows:

$$\tau'_{sd}(k) = \frac{1}{2}(\beta_{sd}+\beta_{ds})\tau_{sd}+\frac{1}{2}(\beta_{sd}-\beta_{ds})(\tau_{dk}-\tau_{sk}) \quad (17)$$

Observation 4.2. *Equation (16) shows that TANOC depends on the node position.*

Observation 4.3. *By averaging over k in equations (16) and (17) we obtain a measure of average traffic-aware criticality. Let $\tau'_{sd}=\frac{1}{n}\sum_k\tau'_{sd}(k)$, then*

$$\tau'_{sd} = \frac{1}{2}(\beta_{sd}+\beta_{ds})\tau_{sd}+\frac{1}{2n}(\beta_{sd}-\beta_{ds})(\tau_{d*}-\tau_{s*}) \quad (18)$$

where $\tau_{i*}=\sum_k\tau_{ik}$. In the same way, Let $\tau'=\frac{1}{n}\sum_k\tau'_k$, then:

$$\tau' = \frac{1}{2}\sum_{s,d}(\beta_{sd}+\beta_{ds})\tau_{sd}+\frac{1}{2n}\sum_{s,d}(\beta_{sd}-\beta_{ds})(\tau_{d*}-\tau_{s*}) \quad (19)$$

We refer to τ' as Traffic-Aware Network Criticality (TANC).

Considering the fact that $\beta_{sd}=1+\frac{\gamma_s(d)}{\gamma}$ (see equation(8)), equation (19) can be rearranged as follows:

$$\tau' = \sum_{s,d}\alpha_{sd}\tau_{sd} \;\; where \;\; \alpha_{sd}=1+\frac{\gamma_{sd}+\gamma_{ds}}{2\gamma}+\frac{\gamma_{*s}-\gamma_{s*}}{n\gamma} \quad (20)$$

It can be easily shown that $1-\frac{1}{n}\leq\alpha_{sd}\leq 1+\frac{1}{n}$ which in turn means that $\alpha_{sd}\geq 0\ \forall\ n\geq 1$.

Observation 4.4. *According to equation (20), TANC can be written as a linear function of τ_{sd}'s (note that $\alpha_{sd} \geq 0$). Since τ_{sd} is a convex function of link weights ([Ghosh et al. (2008)]), TANC is also a convex function of link weights.*

Observation 4.5. *There are two special cases of interest in equation (14).*

1. $\gamma_s(d) = 0$, *or* $\gamma_s(d) = \frac{\gamma}{n(n-1)}$ $\forall\, s-d$ *pairs* $(s \neq d)$
   When there is no traffic, equation (19) is reduced to the original definition of network criticality given by equation (6). Moreover, when the external traffic is uniform, TANC is proportional to the original network criticality.
2. $\gamma_s(d) = \gamma_d(s)$ $\forall\, s-d$ *pairs*
   In the case of symmetric traffic demand matrix, equation 19 can be simplified as follows.

$$\tau'_k = \sum_{s,d} \beta_{sd}\tau_{sd} = \tau' \tag{21}$$

   In this case, according to equation (21), TANOC is independent of the node position.

5. Case of Uniform Traffic: Importance of Network Criticality

Our main goal in this part of the paper is to show why the concept of network criticality is important in communication networks. In fact, we try to shed more light on the concept of network criticality as the survival value for networks by providing some of its interpretations. We will show that network criticality can quantify the average path cost in a journey from one arbitrary source to an arbitrary destination (averaged over all node pairs). We will also show that network criticality quantifies connectivity properties of the network, and finally we show that network criticality can delay the onset of congestion in data communication networks. In this section, we consider a special case of the traffic-aware network criticality, where all the entries of the traffic matrix are the same (uniform traffic). In the case of uniform traffic the traffic-aware network criticality is proportional to the network criticality or total effective resistance of a network (Observation 4.5).

5.1. Network Criticality and Average Path Cost

We consider certain cost to traverse a link (and to move along a path) and study the relationship between network criticality τ and average cost incurred by a message walking from a source s to a destination d averaged over all possible $s-d$ pairs. Suppose for each link $l=(i,j)$ there is a cost $z_l = z(i,j)$ (which is different from the weight of the link). Suppose a message (random-walk) starts from source node s, at each step it passes one link, incurs a cost, and continues until it is absorbed by destination d. We intend to calculate the average cost of such journey over all $s-d$ pairs. The following theorem summarizes the main result:

Theorem 5.1. *The average network cost (denoted by $\bar{\varphi}$) is the product of normalized network criticality and total weighted graph cost ($\sum_k(\sum_j w_{kj}z(k,j))$). If $\sum_k(\sum_j w_{kj}z(k,j))$ is fixed at constant value C (maximum budget) then the average network cost is proportional to the criticality of the network. More specifically:*

$$\bar{\varphi} = \frac{1}{2}\hat{\tau}\sum_k(\sum_j w_{kj}z(k,j) \tag{22}$$

Proof See Appendix A.

□

This interpretation of network criticality is important because in many practical situations we aim to minimize the average cost of a network. For example most of the traffic engineering algorithms try to minimize a kind of cost in the system. Another example is network planning (or re-planning). In network design we have an optimization criteria where a cost metric is minimized.

One special case of interest is when different links of the network have equal costs (normalized by one). In this case the average travel cost equals the average hop length of a path (along which a message or random-walk travels). Average hop length is an important quantity in the mathematical study of communication networks. The following lemma reveals the relevance of network criticality and average hop length.

Lemma 5.2. *Let T denote the average length (number of hops) of a path over all source-destination pairs, and $\bar{W}$ denote the average weight of all nodes. Then:*

$$T = \frac{n}{2}\bar{W}\hat{\tau}$$

Proof It is enough to set $z(i, j) = 1$ for all the network links in equation (22).

$\square$

Lemma 5.2 shows that the average hop length of a random-walk is proportional to the product of normalized network criticality and average node weights. If we fix the total weight of a network at a budget C, then the average hop length of a walk would be proportional to the normalized network criticality, therefore, the normalized network criticality can quantify the average path length for the network flows.

5.2. Network Criticality and Average Betweenness Sensitivity

Here we explain a key interpretation of network criticality. We will show that network criticality equals the average link betweenness sensitivity of the network, where sensitivity is defined as the partial derivative of link betweenness with respect to its weight. According to this fact, minimization of network criticality results in least sensitive network coniguration, which is one of our main goals in using network criticality as the survival value.

To see this, we note that $\tau = \frac{b_{ij}}{w_{ij}}$, therefore for $w_{ij} > 0$ we have:

$$\frac{\partial \tau}{\partial w_{ij}} = \frac{1}{w_{ij}} \frac{\partial b_{ij}}{\partial w_{ij}} - \frac{\tau}{w_{ij}} \quad or \quad w_{ij} \frac{\partial \tau}{\partial w_{ij}} = \frac{\partial b_{ij}}{\partial w_{ij}} - \tau \tag{23}$$

By adding the results of equation (23) for different links of the network one can see:

$$\sum_{(i,j)\in E} w_{ij} \frac{\partial \tau}{\partial w_{ij}} = \sum_{(i,j)\in E} \frac{\partial b_{ij}}{\partial w_{ij}} - m\tau \tag{24}$$

Combining equation (24) and lemma Appendix D.1 results in:

$$\tau = \frac{1}{m-1} \sum_{(i,j)\in E} \frac{\partial b_{ij}}{\partial w_{ij}} \tag{25}$$

where m is the number of links of the network.

Observation 5.3. *According to equation (25) network criticality τ can be interpreted as the average of link betweenness derivatives or sensitivities with respect to link weight.*

Equation (25) suggests an effective approach to design routing and flow assignment algorithms. If we can estimate the variation of each link betweenness with respect to its weight (i.e. $\frac{\partial b_{ij}}{\partial w_{ij}}$), then we can use this variation as a cost to develop routing strategies and to find min-cost paths.

5.3. Network Criticality and Connectivity

Now we investigate the effect of connectivity in the behavior of network criticality, and the effect of controlling network criticality on the connectivity properties of a network. To develop the relationship we need another metric to quantify the connectivity of a network. Fiedler [Fiedler (1973)] defined algebraic connectivity as the smallest non-zero eigenvalue (λ_2) of the Laplacian matrix of a connected graph. Algebraic connectivity is a lower bound for node connectivity and link connectivity. Therefore, *the further λ_2 is from zero, the higher the node and link connectivity of a graph*. In graph literature, the algebraic connectivity is widely used as the main metric to quantify the connectivity of a network.

We now establish lower and upper bounds for network criticality based on algebraic connectivity.

Theorem 5.4. *Normalized network criticality satisfies the following bounds :* $\frac{2}{(n-1)\lambda_2} \leq \hat{\tau} \leq \frac{2}{\lambda_2}$.

Proof See Appendix B

□

Theorem 5.4 shows the relationship between network criticality and connectivity. Since normalized network criticality is upper bounded by the reciprocal of algebraic connectivity, improvement of connectivity (increasing λ_2) improves the robustness as well (decreasing the upper bound of $\hat{\tau}$), but it is important to note that increasing connectivity at the same time decreases the lower bound of network criticality, which in turn causes more variance in network criticality. In other words, we can't uniformly improve the robustness of a network just by increasing the connectivity.

5.4. Network Criticality in Communication Networks

In this section we show the importance of network criticality in the study of communication networks. Let λ be the average input rate of the network, and let the weight of each link be the capacity of the link $(i, j) = l$ (i.e. $w_{ij} = c_{ij} = c(l)$). Further, let x_k be the average load on node k and c_k be the capacity of node k. By applying Little's formula and using lemma 5.2 we have:

$$x_k = \lambda \pi_k T = \lambda \frac{W_k}{\sum_i W_i} \frac{n}{2} \bar{W} \hat{\tau} = \frac{\lambda}{2} W_k \hat{\tau} \tag{26}$$

But W_k is the total capacity of node k, therefore

$$x_k \leq W_k \;\; \Rightarrow \;\; \frac{\lambda}{2} W_k \hat{\tau} \; \leq \; W_k \;\; \Rightarrow \;\; \lambda \;\; \leq \;\; \frac{2}{\hat{\tau}} \tag{27}$$

We can summarize these results in theorem 5.5.

Theorem 5.5. *To maximize the carried load of a network, one needs to minimize the (normalized) network criticality, where the link weight is defined as the link capacity:*

$$\max_W \lambda = \frac{2n(n-1)}{\min_W \tau} = \frac{2}{\min_W \hat{\tau}}$$

□

All of these interpretations have a common message: network criticality is an important performance metric in a communication network, and it should be minimized to have the most robust network. Next section will provide an important interpretation of the concept of criticality in general nonuniform traffic situations.

6. Importance of Traffic-Aware Network Criticality in Data Networks

In this section we will show that the traffic-aware random-walk betweenness and network criticality as defined in section 3, are directly related to the load in data networks. We will show that the average network utilization can be expressed in terms of TANC and network criticality, which again manifests the importance of TANC.

6.1. Packet Networks and Random-Walk Betweenness

First, we show that random-walk betweenness is closely related to packet network models. Consider a packet switching network in which packets arrive to packet switches from outside the network according to independent arrival processes. Each external packet arrival has a specific destination and the packet is forwarded along the network until it reaches said destination. We suppose that packet switches are interconnected by transmission lines that can be modeled as single-server queues. Furthermore, suppose that packet switches use a form of routing where the proportion of packets at queue i forwarded to the next-hop queue j is p_{ij}.

We calculate the total arrival/departure rate of the traffic to/from each node. The total input rate of node k (internal plus external) is denoted by x_k. After receiving service at the i^{th} node, the proportion of customers that proceed to node k is p_{ik}. To find x_k we need to solve the following set of linear equations (see [Kleinrock (1975)]):

$$x_k = \gamma_k + \sum_{i=1}^{n} x_i p_{ik} \tag{28}$$

where γ_k is the external arrival rate to node k. Note that equation (28) is essentially similar to KCL (Kirchhoff's Current Law). If we denote $\overrightarrow{x} = [x_1, x_2, ..., x_n]$ and $\overrightarrow{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_n]$, then equation (28) becomes:

$$\overrightarrow{x} = \overrightarrow{\gamma} + \overrightarrow{x}P \tag{29}$$

Suppose we focus on traffic destined to node d, then node d is an absorbing node, and we suppose that the arrival rate at node d is zero (since said arrivals do not affect other nodes) and equation (29) can be written as:

$$\overrightarrow{x_d} = (\overrightarrow{\gamma_d} + \overrightarrow{x_d}P_d)\Theta_d \tag{30}$$

where $\overrightarrow{x_d}$ and $\overrightarrow{\gamma_d}$ are the same as $\overrightarrow{x}$ and $\overrightarrow{\gamma}$ except for the d^{th} element which is 0. Matrix P_d is also the same as P except that its d^{th} row and d^{th} column are zero vectors. Equation (30) can be solved for $\overrightarrow{x_d}$.

$$\overrightarrow{x_d} = \overrightarrow{\gamma_d} \times \Theta_d \times (I - P_d \times \Theta_d)^{-1} \tag{31}$$

To find the relationship of betweenness B_d and the input arrival rate x_k we notice that $p_{dk}(d) = 0$ which means that $P_d = \Theta_d \times P_d$. Thus:

$$\begin{aligned} P_d \times \Theta_d &= \Theta_d \times P_d \times \Theta_d \\ \Theta_d - P_d \times \Theta_d &= \Theta_d - \Theta_d \times P_d \times \Theta_d \\ \Theta_d \times (I - P_d \times \Theta_d)^{-1} &= (I - P_d)^{-1} \times \Theta_d \end{aligned}$$

Using equation (1) we will have:

$$\Theta_d \times (I - P_d \times \Theta_d)^{-1} = B_d \tag{32}$$

We substitute equation (32) in (31) to find the relationship between the node traffic and node betweenness.

$$\overrightarrow{x_d} = \overrightarrow{\gamma_d} \times B_d \tag{33}$$

If we denote the k^{th} element of $\overrightarrow{x_d}$ and $\overrightarrow{\gamma_d}$ by $x_k(d)$ and $\gamma_k(d)$ respectively, we have:

$$x_k(d) = \sum_s \gamma_s(d) b_{sk}(d) \tag{34}$$

Now we can find the total load at node k by adding the effect of all destinations in equation (34).

$$x_k = \sum_d x_k(d) = \sum_{s,d} \gamma_s(d) b_{sk}(d) \tag{35}$$

Comparing equation (35) and equation (9) provide us with an important fact. Suppose the packets are routed according to the random-walk transition probability matrix P. Then, the total load on each link of the network is proportional to the traffic-aware random-walk betweenness, more precisely: $x_k = \gamma b'_k$.

6.2. Network Utilization and Traffic-Aware Network Criticality

We now show that the average network utilization can be expressed in terms of traffic-aware network criticality. Node utilization is defined as the load of a node normalized by its capacity (or in a more general sense by its weight), the utilization of node k is equal to $V_k = \frac{x_k}{W_k}$. We denote the average network utility by $\bar{V} = \frac{\sum_k V_k}{n}$. Considering equations (35), (9), and using $\frac{b'_k}{W_k} = \frac{1}{2}\tau'_k$, one can see that $V_k = \frac{\gamma}{2}(\tau'_k - \tau)$ and $V = \frac{\gamma}{2}(\tau' - \tau)$.

The important consequence of the above facts is that some key performance measures in communication networks are directly related to the concept of traffic-aware betweenness and traffic-aware network criticality. For example, we saw that the average network utilization is equal to $V = \frac{\gamma}{2}(\tau' - \tau)$, therefore, minimizing average network utilization is equal to minimizing the difference of traffic-aware network criticality and original network criticality. Hence, the problem of minimizing average network utilization can be formulated as minimization of a convex+convace function which can be converted to a convex maximization problem.

7. Minimizing Traffic-Aware Network Criticality

Considering all the interpretations for network criticality and TANC which are discussed in previous sections, we arrive at the result that the minimization of TANC (or network criticality as special case) is desired. In this section we formulate an optimization problem for TANC. We consider minimization of a general linear function of effective resistances (or end-to-end network criticalities) as $\tau_\alpha = \sum_{s,d} \alpha_{sd}\tau_{sd},\ \alpha_{sd} \geq 0\ \forall s, d \in N$. Traffic-aware network criticality is clearly one example of τ_α with appropriate selection of coefficients (according to equation (20)).

First we show that the minimization of τ_α is possible. To this end we need the following lemma.

Lemma 7.1. *The partial derivative of τ_α with respect to link weight w_{ij} is always non-positive.*

Proof See Appendix C.

□

Since τ_α is a convex function and its derivative with respect to the weights is always non-positive (according to lemma 7.1), the minimization of τ_α subject to some convex constraint set is possible.

In order to construct the optimization problem, we add a maximum budget constraint to the problem. We assume that there is a cost z_{ij} to deploy each unit of weight on link (i, j). We also assume that there is a maximum budget of C to establish all network links. This constraint means that $\sum_{(i,j)\in E} w_{ij}z_{ij} = C$. Now we can write our optimization problem as follows:

$$\begin{array}{cc} & Minimize \quad \tau_\alpha \\ Subject\ to & \sum_{(i,j)\in E} z_{ij}w_{ij} = C \quad , C\ is\ fixed \\ & w_{ij} \geq 0 \ \ \forall (i,j) \in E \end{array} \tag{36}$$

Theorem 7.2. *The condition of optimality for optimization problem (36) can be written as:*

$$\min_{(i,j)\in E} \frac{C}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}} + \tau_\alpha \geq 0$$

More specifically:

$$w^*_{ij}(C\frac{\partial \tau_\alpha}{\partial w_{ij}} + z_{ij}\tau_\alpha) = 0 \ \ \forall (i,j) \in E \tag{37}$$

where w^*_{ij} denotes the optimal weight for link (i,j).

Proof See Appendix D.

□

7.1. Semi-definite Program (SDP) to Minimize τ_α

Optimization problem (36) can be converted to the following semi-definite programming problem (SDP):

$$\begin{array}{c} Minimize \quad \sum_{s,d\in N} \alpha_{sd}t_{sd} \\ Subject\ to \quad Tr(Z^tW) = C \quad , C\ is\ fixed \\ diag(Vec(W)) \geq 0 \\ \begin{pmatrix} t_{sd} & u^t_{sd} \\ u_{sd} & L + \frac{J}{n} \end{pmatrix} \geq 0 \quad \forall s,d \in N \end{array} \tag{38}$$

where $\geq$ means positive semi-definite. Furthermore, $Vec(W)$ is a vector obtained by concatenating all the rows of weight matrix W and $diag(x)$ means a diagonal matrix with main diagonal equal to vector x.

Before closing this section, we discuss some special cases of interest, where we can simplify our semi-definite program (38).

7.1.1. Uniform Traffic case

In case of uniform traffic, τ_α is proportional to the network criticality $\hat{\tau}$. Suppose we let $\Gamma = (L + \frac{J}{n})^{-1}$, where J is a n-by-n matrix with all elements equal to 1, then Γ can be written as a semi-definite inequality as follows. We consider matrix $\Theta = \begin{pmatrix} \Gamma & I \\ I & L + \frac{J}{n} \end{pmatrix}$. The necessary and sufficient condition for positive semi-definiteness of Θ is that its Schur complement [Bernstein (2005)] be positive semi-definite. In general, the Schur complement of a matrix of the form $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ is: $A - BD^{-1}C$. Hence the Schur complement of Θ is $\Gamma - (L + \frac{J}{n})^{-1}$, and

$$\Theta = \begin{pmatrix} \Gamma & I \\ I & L + \frac{J}{n} \end{pmatrix} \geq 0 \Leftrightarrow \Gamma \geq (L + \frac{J}{n})^{-1} \tag{39}$$

On the other hand, in [Tizghadam et al. (2010)] it is shown that average network criticality is proportional to the some of the reciprocals of the non-zero eigenvalues of the graph Laplacian matrix (or the trace of Laplacian).

$$\hat{\tau} = \frac{2}{n-2} Tr(L^+) = \frac{2}{n-1} \sum_{i=2}^{n} \frac{1}{\lambda_i} \tag{40}$$

Since the optimization problem (36) should minimize $Tr(\Gamma)$ (according to equation (40)), the equality in equation (39) is chosen: $\Gamma = (L + \frac{J}{n})^{-1}$. Now optimization problem (36) can be converted to a semi-definite programming:

$$\begin{gathered} Minimize \quad \tfrac{2}{n-1} Tr(\Gamma) - \tfrac{2}{n-1} \\ Subject\ to \quad Tr(Z^t W) = C \quad , C\ is\ fixed \\ diag(Vec(W)) \geq 0 \\ \begin{pmatrix} \Gamma & I \\ I & L + \frac{J}{n} \end{pmatrix} \geq 0 \end{gathered} \tag{41}$$

7.1.2. Minimizing Maximum of Point-to-Point Network Criticality

We consider one more extension. In some scenarios we need to minimize the maximum of point-to-point network criticality (instead of its linear weighted average). The optimization problem in this case is as follows.

$$\begin{aligned} & Minimize \quad \max_{i,j \in N} \tau_{ij} \\ Subject\ to \quad & \textstyle\sum_{(i,j)\in E} w_{ij} z_{ij} = C \quad , C\ is\ fixed \\ & w_{ij} \geq 0 \;\; \forall (i,j) \in E \end{aligned} \tag{42}$$

Suppose $t = \max_{i,j\in N} \tau_{ij}$. We have:

$$\begin{aligned} t &= \max_{i,j\in N} \tau_{ij} \\ t &\geq u_{ij}^t L^+ u_{ij} \\ t &\geq u_{ij}^t (L + \frac{J}{n})^{-1} u_{ij} \end{aligned}$$

But, according to Schur's complement, this can be written in the following form: $\begin{pmatrix} t & u_{ij}^t \\ u_{ij} & L + \frac{J}{n} \end{pmatrix} \geq 0$. Therefore, optimization problem (42) can be written as the following semi-definite program.

$$\begin{aligned} & Minimize \quad t \\ Subject\ to \quad & Tr(Z^t W) = C \quad , C\ is\ fixed \\ & diag(Vec(W)) \geq 0 \\ & \begin{pmatrix} t & u_{ij}^t \\ u_{ij} & L + \frac{J}{n} \end{pmatrix} \geq 0 \quad \forall i,j \in N \end{aligned} \tag{43}$$

Note that this optimization problem is different from problem (38).

8. Network Planning

Optimization problem (38) provides a framework for designing appropriate link weights to minimize TANC, or equivalently to optimize many network performance metrics according to the interpretations provided for TANC and network criticality. As an instance, optimization problem (38) can be used to minimize the average cost of traveling along a network, or a modified version of optimization problem (38) can be used to minimize the average network utilization. One can then use the properties of optimization problem (38) to derive network control algorithms (such as traffic engineering). In this paper we concentrate on the first goal and try to solve optimization problem (38) for some representative networks. The optimization problem can be solved with standard methods for solving semi-definite programs. There are also various commercial and academic software tools to solve semi-definite programs. We used open-source CVX package [Grant et al. (2008); Grant et al. 2 (2008)] for our examples in this paper.

8.1. Capacity Planning

In this section we study the capacity planning as an important special case of network planning problem. Consider a network $G(N, E, W)$ where the link weights are equal to the link capacities, that is, $w_{ij} = c_{ij}\ \forall (i, j) \in E$ (c_{ij} denotes the capacity of link (i, j)). We assume that a routing strategy is already in place and the demands (based on traffic matrix) are mapped to link loads according to this routing method (for example we can assume the traffic is distributed using shortest path routing). We investigate the capacity assignment problem in which network topology and link traffic load $\lambda_{ij}\ \forall (i, j) \in E$ are assumed known and fixed. The goal is to find the capacity of the links so as to minimize TANC (or network criticality) under the constraint that the total cost of the planning is fixed. Let z_{ij} be the symmetric cost of assigning capacity c_{ij} to link (i, j), and suppose that we have a linear cost function. The total cost of the capacity assignment problem is $\sum_{(i,j)\in E} z_{ij}c_{ij}$. We fix this total cost at maximum budget C. We can write the optimization problem for capacity assignment problem as follows:

$$\begin{array}{rcl} & Minimize & \tau_\alpha \\ Subject\ to & \sum_{(i,j)\in E} c_{ij}z_{ij} = C & , C\ is\ fixed \\ & c_{ij} \geq \lambda_{ij} & \end{array} \qquad (44)$$

By applying the change of variable $c_{ij} = c'_{ij} + \lambda_{ij}$ to the optimization problem (44), we will have the following convex optimization problem.

$$\begin{array}{rcl} & Minimize & \tau_\alpha \\ Subject\ to & \sum_{(i,j)\in E} c'_{ij}z_{ij} = C' & , C'\ is\ fixed \\ & c'_{ij} \geq 0 & \end{array} \qquad (45)$$

where $C' = C - \sum_{(i,j)\in E} z_{ij}\lambda_{ij}$. This has the same form of the optimization problem (36) (with $w_{ij} \rightarrow c'_{ij}$ and $C \rightarrow C'$), therefore, all the results developed for optimization problem (36) are applicable for the capacity assignment problem.

Observation 8.1. *In our capacity assignment problem we assumed that a known routing method governs the distribution of traffic along different paths of the network. A more general problem is the joint optimal allocation of capacities and assignment of flows (routing). In other words, a more sophisticated problem is the simultaneous routing and resource allocation in order to minimize TANC. This problem can also be formulated with a similar approach. We just need to consider link flows as our variables (in addition to weights or capacities) and add the flow conservation constraints to the optimization problem.*

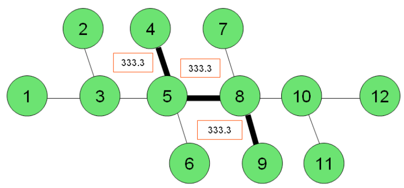
9. Evaluation

In this section we solve optimization problem (36) for various network scenarios and topologies to show our framework can be used to design robust networks.

9.1. Parking-Lot Network

The parking-lot network topology, shown in Fig. 1-(a), has been a challenging network despite its simple topology. Suppose the capacity of all the links are 1. Suppose one unit of bandwidth is requested to be sent between the following source-destination pairs: $(1, 12)$, $(2, 6)$, $(4, 9)$, and $(7, 11)$. If the demand from node 1 to node 12 is serviced first, almost all of the existing routing algorithms will choose the straight path resulting in the blocking of demands of one unit coming from all other source-destination pairs. A wiser decision is to block the first request from node 1 to node 12 so the network will be able to route the other requests. This experiment suggests that when the traffic is symmetric, in weight allocation for parking-lot topology, it is desired to assign more weight for middle (core) links. To verify this guess, we solve our previously discussed optimization problems for parking-lot.

First suppose that there is traffic only between two specific nodes a and b (i.e. $\gamma_{ij} = 0 \quad \forall i, j \in N$ except for $\gamma_{ab} = \gamma_{ba}$). In this case TANC (and average network utilization) can be written as: $\tau' = \gamma_{ab}\tau_{ab}$. Our desired optimization problem in this case is (38). Solution of optimization problem (38) suggests that we can minimize the end-to-end resistance distance between points a and b by allocating non-zero weights along the



Link	Weight (Uniform)	Weight (Nonuniform)	Weight (Minimax)
(1,3)	77.2654	138.3109	92.5021
(2,3)	77.2654	125.3322	92.5021
(3,5)	121.0498	**144.1829**	130.8176
(4,5)	77.2656	93.0090	54.1865
(5,6)	77.2	74.2826	54.1865
(5,8)	**139.7768**	119.2677	**151.6105**
(7,8)	77.2655	67.6480	54.1865
(8,9)	77.2656	57.5601	54.1865
(8,10)	121.049	82.9718	130.8176
(10,11)	77.2654	50.2098	92.5021
(10,12)	77.2654	47.2251	92.5021

(a) Parking-Lot Topology (b) Optimal Weight in different Scenarios

Figure 1: Optimal Weights for Parking-Lot

links between a and b. Thus if $a = 4$ and $b = 9$, the cost of all links is 1, and the budget is $C = 2000$, then the optimal weight assignment assigns equal weights as indicated in 1-(a) by thick lines.

Next suppose that all pairs in the network have the same volume of traffic between them. The optimum weight assignment which is the solution of optimization problem (41) is shown in the second column of the table in Fig. 1-(b), assigns higher weights to the bottleneck links $(3,5)$, $(5,8)$, and $(8,10)$.

Finally consider a nonuniform traffic matrix given by: $\gamma_i(j) = \frac{1}{(i+j)^2}$ $\forall i, j \in N,\ \ i \neq j$. In this case the nodes with lower indices have higher traffic volumes flowing between them. The third column of the table in Fig. 1-(b) shows how the optimal weight assignment places higher weight in the links between lower-indexed nodes.

Instead of minimizing network criticality (total resistance distance) we can minimize the maximum point-to-point resistance subject to a cost constraint. The fourth column of the table in Fig. 1-(b) shows the optimal weight assignment (solution of optimization problem (42)) that achieves the minimax value of point-to-point resistance distance for parking-lot network.

9.2. A General Tree Network

In this section we derive a general formula for TANC of a tree with a nonuniform traffic matrix $[\gamma_{ij}]$. We note that a tree is an acyclic simple graph, which means that there is exactly one path between every two nodes of a tree. It follows that network criticality of a tree can be found from the following equation.

$$\tau_\alpha = \sum_{(i,j)\in E} \frac{\lambda_{ij}}{w_{ij}} \tag{46}$$

where λ_{ij} denotes the total traffic passing through link (i, j) (λ_{ij} is the sum of those components of the traffic matrix whose end-to-end path traverses link (i, j)). By equation (46) we have:

$$\frac{\partial \tau_\alpha}{\partial w_{ij}} = -\frac{\lambda_{ij}}{w_{ij}^2} \tag{47}$$

The condition of optimality given by equation (37) along with equation (47) results in:

$$\frac{\partial \tau_\alpha}{\partial w_{ij}} = -\frac{\lambda_{ij}}{w_{ij}^2} = -\frac{z_{ij}\tau_\alpha}{C}$$

Hence:

$$w_{ij} = \left(\frac{\lambda_{ij}C}{z_{ij}\tau_\alpha}\right)^{\frac{1}{2}} \tag{48}$$

On the other hand we have $\sum_{(i,j)\in E} z_{ij}w_{ij} = C$ (constraint of the optimization problem), therefore:

$$\sum_{(i,j)\in E} \left(\frac{\lambda_{ij}z_{ij}C}{\tau_\alpha}\right)^{\frac{1}{2}} = C \tag{49}$$

$$\tau_\alpha = \left(\sum_{(i,j)\in E} \left(\frac{\lambda_{ij}z_{ij}}{C}\right)^{\frac{1}{2}}\right)^2 \tag{50}$$

Now it is enough to substitute τ from equation (50) in equation (48) to have optimal weight for tree.

$$w_{ij} = \frac{C}{z_{ij}} \times \frac{(\lambda_{ij}z_{ij})^{\frac{1}{2}}}{\sum_{(i,j)\in E}(\lambda_{ij}z_{ij})^{\frac{1}{2}}} \tag{51}$$

Equation (51) shows that the optimal weight of a link in a tree is proportional to the square root of λ_{ij}.

9.2.1. Capacity Planning for a Tree

The capacity assignment problem for a tree when the link loads are known can be solved by applying the following changes in equation (51):

$$\begin{aligned} w_{ij} &\rightarrow c_{ij} - \lambda_{ij} \\ C &\rightarrow C - \sum_{(i,j)\in E} z_{ij}\lambda_{ij} \end{aligned}$$

The optimal capacity assignment for a tree would be:

$$c_{ij} = \gamma_{ij} + \frac{C - \sum_{(i,j)\in E} z_{ij}\gamma_{ij}}{z_{ij}} \times \frac{(\lambda_{ij} z_{ij})^{\frac{1}{2}}}{\sum_{(i,j)\in E}(\lambda_{ij} z_{ij})^{\frac{1}{2}}} \tag{52}$$

Our result and Kleinrock's result for capacity assignment coincide when the network is acyclic (tree). In [Kleinrock (1975)] Kleinrock showed that under the independence assumption the optimal capacity of a link, in order to minimize average delay of the network, is proportional to the square root of the link rate. Note that λ_{ij} is the link load, as a result, equation (52) is similar to the Kleinrock's equation for optimal capacity ([Kleinrock (1975)], §5.7, equation 5.26). As a matter of fact, this result is expected because the network criticality of a tree according to equation (46) is equal to $\tau = \sum_{(i,j)\in E} \frac{\lambda_{ij}}{c_{ij}-\lambda_{ij}}$ (considering $w_{ij} = c_{ij} - \lambda_{ij}$), which is exactly the same expression that is used in [Kleinrock (1975)] to find the average delay of a network (Kleinrock (1975), §5.6, equation 5.19), therefore, the minimization of network criticality equals the minimization of the average network delay in acyclic networks.

9.3. Kleinrock's Network

Here we compare our proposed optimal weight assignment strategy with two other well-known capacity assignment methods, Kleinrock's approach [Kleinrock (1964, 1975)] and Meister's extension [Meister et al. (1971)] using the example of telegraph network in Kleinrock illustrated in Fig. 2 (see [Kleinrock (1964)], pp. 22-23). We let $z_{ij} = 1$ for all the links in this test. Kleinrock assumes that shortest path routing is used for routing and end-to-end traffic requirements (traffic matrix) are converted to link loads, therefore, the link loads are assumed to be known. Kleinrock's method finds the link capacities that minimize the average delay of the

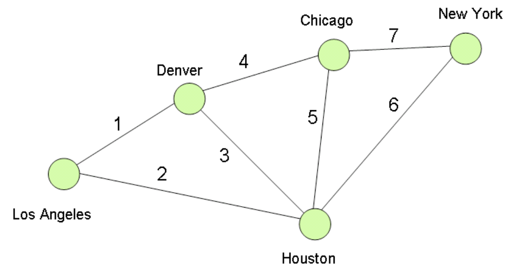


Figure 2: Kleinrock's Network

network. However this solution assigns very long delays to the links with small flows. Meister's method modified the objective function to reduce this unfairness, and in an extreme case leads to a capacity assignment that results in equal delays in all the links at the expense of a large deviation from optimal average network delay which can be achieved by Kleinrock's solution. The proposed solution in this paper balances the individual link delays so as to have fair link delays while the average network delay is still kept small. Table 1 shows the capacity assigned to the links using all the methods. The second column of table 1 shows the individual link loads.

Link	Load	Kleinrock	Meister	Criticality Method
1	3.15	27.93	27.00	29.63
2	3.55	29.85	27.40	33.31
3	0.13	5.16	23.98	12.67
4	3.64	30.28	27.49	32.95
5	0.82	13.46	24.67	13.36
6	3.88	31.38	27.73	33.64
7	9.95	53.99	33.80	36.43

Table 1: Capacity Assignment using 3 Different Methods

Columns 3, 4, and 5 show the optimal capacity assignment using Kleinrock's method, Meister's method, and our proposed method (which we call it criticality method) respectively. The second column of table 2 shows the minimum average network delay for all three methods. The third column shows the value of network criticality computed using optimal capacity

Method	Average Network Delay	Network Criticality
Kleinrock	44.72	1.06
Meister	55.01	0.80
Criticality Method	49.30	0.56

Table 2: Average Network Delay and Network Criticality using Different Methods

in each one of the methods. In our method we optimize the robustness (not the average delay as it is the case in Kleinrock and Meister), therefore it is not surprising to see that the average delay obtained by our method is between Kleinrock's approach (minimizing the average network delay) and Meister (minimizing the maximum link delay). Finally, table 3 shows

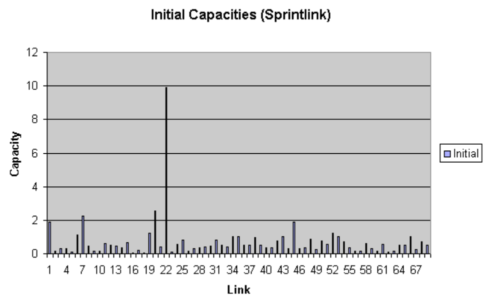
Link	Kleinrock	Meister	Criticality Method
1	40.36	41.93	37.76
2	38.02	41.93	33.60
3	198.67	41.93	79.71
4	37.54	41.93	34.12
5	79.10	41.93	79.71
6	36.36	41.93	33.60
7	22.71	41.93	37.76

Table 3: Individual Link Delays using 3 Different Methods

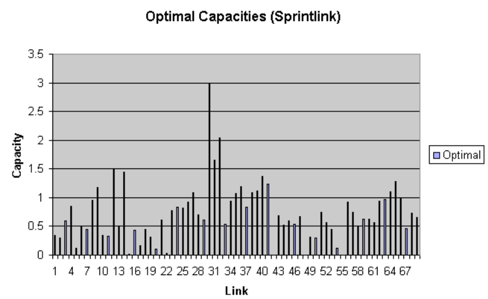
individual link values for all three methods. As one can see, Kleinrock's method assigns very large delay to link 3 because the demand on link 3 is much less than other links. Meister's method assigns equal delays for all the links. This resolves the issue with Kleinrock's method, but introduces a fairness problem. In our proposed method, the link delays are not equal to allow for fairness based on the demand for each link, and at the same time the individual link delays are kept in a reasonable range.

9.4. ISP Topologies

Our next experiment is on real ISP (Internet Service Provider) maps from Rocketfuel dataset [Spring et al. (2004)]. We followed the method



(a) Initial Link Capacities



(b) Optimal Link Capacities

Figure 3: Initial and Optimal Capacity Allocation for Rocketfuel Topology (Sprintlink)

described in [Applegate et al. (2006)] and collapsed the Rocketfuel ISP topologies into PoP to PoP connectivity networks. In other words, we consolidated all the nodes within a city into a single node, and aggregated all the links from one city to another one in a single link, where the capacity of the link equals the sum of the capacity of all the original links connecting different sub-nodes between two cities. There are six ISP topologies in Rocketfuel dataset, whose topological information are given in [Applegate et al. (2006)]. The topologies in Rocketfuel dataset did not include the capacities of the links, but we can use OSPF weight information which is provided in Rocketfuel dataset to associate compatible capacities using Cisco recommendation as described in [Applegate et al. (2006)]. Cisco recommends that the link capacities are proportional to the reciprocal of the weights.

Among six Rocketfuel topologies, We chose Sprintlink topology (1239) as our test network because it is the largest ISP topology in Rocketfuel dataset. Sprintlink originally consists of 315 routers and 1944 links. The collapsed Sprintlink network includes 30 nodes (reduced cities) and 69 reduced links. Using Cisco recommendation we estimated nominal (initial) capacities for Sprintlink topology and used the sum of the nominal capacities (shown in Fig. 3-(a)) as our total capacity budget (this budget is approximately $C = 100$ Gigabits for Sprintlink). Then we solved optimization problem (36) for Sprintlink topology. Fig. 3-(b) shows the optimal link capacity assignment for Sprintlink.

Comparing Fig. 3-(a) and Fig. 3-(b) suggests that the optimal capacity distribution for Sprintlink is more uniform than the initial capacity assignment. This result is expected in dense network with large number of paths between each source-destination pair.

9.5. Complete Graph on n Nodes (K_n)

For K_n and when we have uniform traffic, we can obtain the solution of optimization problem (41) analytically. We let $z_{ij} = 1 \ \forall \ (i, j) \in E$ (we will investigate the effect of link cost on complete graph in the next part of the experiment). We need the following lemma to find the optimal weight set for K_n.

Lemma 9.1. *Consider optimization problem (41) and suppose that the cost matrix $Z = [z_{ij}]$ is equal to $Z = J - I$ (this is a square-matrix with all the components equal to 1, except the main diagonal components which are zero) . If there is an automorphism on a graph G(N,E,W) that can map link $l = (i, j)$ on link $l' = (i', j')$, then these links should have equal optimal weights.*

Proof Let G′ be the transformed graph (after applying the automorphism). An automorphism on a graph G can be represented by a matrix operator T such that the Laplacian of transformed graph G′ can be obtained from Laplacian of original graph G as $L(G') = TL(G)T^t$. This means that L(G) and L(G′) have the same eigenvalues. As a result according to equation (40) the network criticality of graph G equals the network criticality of G′ : $\hat{\tau}(G) = \hat{\tau}(G')$. But we know that the solution of optimization problem (41) is unique due to strict convexity of average network criticality $\hat{\tau}$ [Tizghadam et al. (2010)]. As a result the weights of link l and l′ are equal.

□

Corollary 9.2. *Consider optimization problem (41) and suppose that the graph of the network is an edge-transitive graph with equal link costs. The optimal weight for a link $(i, j) \in E$ is $w_{ij} = \frac{C}{m}$, where m is the number of graph links.*

Proof We say a graph is edge-transitive, if there is an automorphism that can map any two links of the graph. According to lemma 9.1 all the link weights of an edge-transitive graph are equal. In addition, suppose $w_{ij} = w \ \forall \ (i, j) \in E$, then constraint $\sum_{(i,j)\in E} w_{ij} = C$ implies that $w = \frac{C}{m}$. This completes the proof of corollary 9.2.

□

Complete graph K_n is an edge-transitive graph, therefore, according to corollary 9.2 the optimal weight of all the links of K_n are equal. Let denote this common weight by w. It is an easy task to verify that the eigenvalues of the Laplacian matrix of K_n are $\lambda_i = nw \;\; \forall \;\; i \in N$. We can also find link weight w from corollary 9.2. The total number of links in K_n is $m = n(n-1)$, therefore, considering corollary 9.2 we have: $w = \frac{1}{n(n-1)}C$. As a result we find: $\lambda_i = \frac{1}{n-1}C \quad \forall \; i \in N$. Now we can calculate network criticality for graph K_n using this equation and lemma 9.1.

$$\hat{\tau} = \frac{2}{n-1}\sum_{i=2}^{n}\frac{1}{\lambda_i} = \frac{2(n-1)}{C} \tag{53}$$

According to equation (53) the optimal network criticality in a complete graph linearly increases with the size of the network.

9.5.1. Case of Unequal Link Costs for K_n

Now we would like to study the effect of link costs on the optimal solution. We can use the semi-definite programming to find optimal weights of K_n when the link costs (z_{ij}'s) are not equal. We use a numerical example to show the effect of changes in link costs. We consider the complete graph on 6 nodes (K_6), and we let the link cost matrix be:

$$Z = [z_{ij}] = \begin{pmatrix} 0 & 1.2 & 1 & 1 & 1 & 2 \\ 1.2 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Moreover, let $C = 2000$. We used the semi-definite form of the optimization problem which is described in equation (41) and solved it for K_6 using `CVX`. The optimal weight assignment is given in the following matrix.

$$W = \begin{pmatrix} 0 & 54.982 & 85.765 & 85.746 & 85.738 & \mathbf{0.003} \\ 54.982 & 0 & 64.057 & 63.995 & 63.975 & 78.027 \\ 85.765 & 64.057 & 0 & 54.321 & 54.233 & 81.188 \\ 85.746 & 63.995 & 54.321 & 0 & 54.435 & 81.258 \\ 85.738 & 63.975 & 54.233 & 54.435 & 0 & 81.276 \\ \mathbf{0.003} & 78.027 & 81.188 & 81.258 & 81.276 & 0 \end{pmatrix}$$

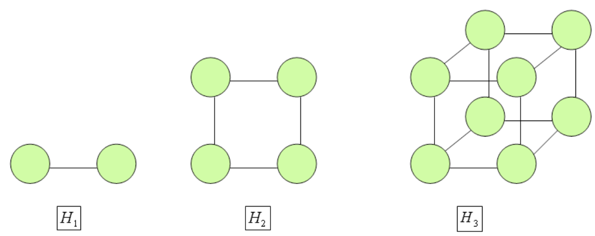


Figure 4: Hypercube Topology (H_1, H_2, H_3)

The weight matrix shows that the optimal weight assignment is not uniform in this case. The optimal weight of link $(1, 6)$ is $w_{16} = w_{61} = 0.003$ which means that link $(1, 6)$ is effectively disconnected. In other words, the topology of the optimal network is not K_6 anymore.

9.6. Hypercube with 2^n nodes (H_n)

In our last example, we consider hypercube of order n (H_n), another well-structured graph whose criticality can be obtained analytically. Fig. 4 shows hypercube topology for $n = 1$ *to* 3. Hypercube is a good choice to be used at the core of data centers assuming that there is a load-balancing mechanism to uniformly distribute the traffic of the edge into the core of the data center. In this example we assume uniform traffic and we let $z_{ij} = 1 \;\; \forall \; (i, j) \in E$. Hypercube is an edge-transitive graph, therefore, by corollary 9.2 the optimal solution of the optimization problem (41) for hypercube has equal weights. Hence, we consider a hypercube with weight w for all the links. In order to find the optimal value of network criticality for hypercube, we need the following lemma.

Lemma 9.3. *Suppose we have a hypercube of order n (H_n) with all the link weights equal to 1. The eigenvalues of the adjacency matrix of H_n are $2k - n$, $k = 0, 1, ..., n$ with multiplicity $C(n, k) = \frac{n!}{k!(n-k)!}$.*

Proof See Appendix E.

□

Lemma 9.3 is true when all the weights are set to 1. In general case where we have a weight w for each link, the eigenvalue is also multiplied by this weight.

We notice that hypercube is a regular graph (degree of all nodes are n). This means that we can find eigenvalues of Laplacian using eigenvalues of the adjacency matrix of H_n:

$$\begin{aligned} L_n &= nI - H_n \\ \lambda_k &= n - (2k - n) \textit{ with multiplicity } C(n,k) \\ \lambda_k &= 2(n - k) \textit{ with multiplicity } C(n,k) \end{aligned}$$

Now one can easily find the network criticality for H_n.

$$\begin{aligned} \hat{\tau} &= \frac{2}{2^n - 1} \sum_k \frac{1}{\lambda_k} \\ &= \frac{1}{(2^n - 1)w} \sum_{k=0}^{n-1} \frac{C(n,k)}{n-k} \end{aligned} \tag{54}$$

On the other hand, considering the fact that the number of links in H_n is $m = n2^n$, by corollary 9.2, we have:

$$w = \tfrac{C}{n2^n} \tag{55}$$

The final expression for network criticality of H_n can be found by applying equation (55) in (54):

$$\hat{\tau} = \frac{n}{(1 - \frac{1}{2^n})C} \sum_{i=1}^{n} \frac{C(n,i)}{i} \tag{56}$$

To obtain the last equation we applied the change of variable $i = n - k$ and used the fact that $C(n, n - i) = C(n, i)$. Equation (56) shows the behavior of normalized network criticality when the size of hypercube increases.

We can compare the normalized criticality of a hypercube H_n with a complete graph K_{2^n} to see how the robustness is decreased by changing a complete graph to a hypercube (with the same number of nodes).

$$\begin{aligned} \frac{\hat{\tau}(H_n)}{\hat{\tau}(K_{2^n})} &= \frac{\frac{n}{(1-\frac{1}{2^n})C} \sum_{i=1}^{n} \frac{C(n,i)}{i}}{\frac{2(2^n-1)}{C}} \\ &\rightarrow \frac{n}{2^{n+1}} \sum_{i=1}^{n} \frac{C(n,i)}{i} \end{aligned} \tag{57}$$

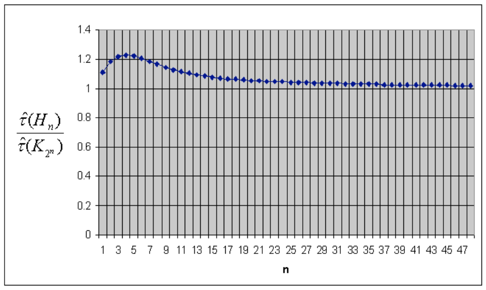


Figure 5: The Ratio of Normalized Network Criticality of Hypercube and Complete Graph

Fig. 5 shows the behavior of equation (57) for different values of n. Clearly for higher values of n, fraction $\frac{\hat{\tau}(H_n)}{\hat{\tau}(K_{2^n})}$ approaches 1. Note that even for high values of n there is a significant gap between the normalized criticality of H_n and K_{2^n}, although the ratio is decreasing.

10. Limitations

The main limitation of the proposed method discussed in this paper is that our results are valid for networks that can be modeled by undirected graphs with symmetric weights. It is not difficult to find communication networks that require directed graph models. For example consider a wireless network that consists of nodes with known transmit powers. The Shannon capacity of each wireless link depends on the value of node transmit power, inter-node distances, and the interference among different nodes. Depending on the arrangement of nodes, the capacity of a link (i, j) may be different from that of link (j, i), and so, we cannot model such a network with an undirected graph. While the results for random-walks exists for directed graphs and the notion of random-walk betweenness remains unchanged, the concept of resistance distance does not apply for directed graphs, therefore, the concept of network criticality in its present form is not applicable in asymmetric graphs.

While the analogy between network criticality and random-walks does not hold in directed graphs, the interpretation of average travel cost (or equivalently average commute time) for network criticality still holds. In fact we have recently shown that the average travel time in a directed graph can be found using the exact same analytical machinery (i.e. the trace of generalized inverse Laplacian matrix) when we use combinatoric Laplacian matrix of a directed graph Tizghadam et al. (2010). However, our expression for the average travel time of directed graph (i.e. the trace of generalized inverse of the Laplacian matrix), in its most general form, is not a convex function of link weights. Therefore, its properties and behavior is far from what we have discussed for undirected graphs. Investigation of this problem is one of the main topics in our present research.

11. Conclusion

In this paper we proposed an approach to the robust network design problem and network planning using graph-theoretical concepts. We defined a new performance metric,traffic-aware network criticality (TANC), to quantify the robustness of a network in nonuniform traffic scenarios and investigated the properties of TANC. We proposed various interpretations for TANC and showed that TANC can measure some important network parameters, such as network utilization and connectivity. We proved that TANC is a convex function of link weights and investigated the convex optimization problem to minimize the traffic-aware network criticality under some constraints on the weight matrix. We found a semi-definite programming representation of the problem which permits us to use available literature on semi-definite programming to solve the optimization problem and find the optimal weights. Capacity assignment problem can be considered as a special case of this general problem where the weight of a link is equal to its capacity.

We believe that framework we have presented in this paper opens a number of venues for further research on a variety of network control algorithms. Our results can be used to design different distributed control mechanisms (for example in multi-agent control networks), or distributed mechanisms for traffic engineering in data networks, in particular where the network utilization needs to be kept at minimum level. One immediate application of the proposed planning method is in data center design, where we need to accommodate heterogeneous traffic at the core of the data center.The framework presented in this paper also reveals the fundamental relationship of communication network design to classical (non-linear) electrical circuit design. We are currently elaborating on this relationship with a view to develop novel design algorithms. Extension of the results of this research to the case of directed graphs is another challenging area that needs further work.

Appendix A. Proof of Theorem 5.1

Let $\varphi(s,d)$ denote the average cost incurred by paths between source s and destination d. We have:

$$\varphi(s,d) = E\{\sum_{k=0}^{\infty} z(d_k, d_{k+1})\} \quad \textit{where} \;\; s = d_0 \tag{A.1}$$

Using elementary probability we can expand equation (A.1) as:

$$\begin{aligned}
\varphi(s,d) &= \sum_{d_1,d_2,...} p(d_1,d_2,...|d_0 = s)(\sum_{k=0}^{\infty} z(d_k,d_{k+1})) \\
&= \sum_{d_1} p_{sd_1}\{z(s,d_1) + \sum_{k=1}^{\infty} p(d_2,...|d_0)z(d_k,d_{k+1})\} \\
\varphi(s,d) &= \sum_j p_{sj}z(s,j) + \sum_j p_{sj}\varphi(j,d) \qquad (A.2)
\end{aligned}$$

Recursive equation (A.2) can be expressed in matrix form as follows. We relabel the nodes so that node d is the last node. Moreover, let $f_s = \sum_j p_{sj}z(s,j)$. Equation (A.2) can be written as:

$$\begin{aligned}
\overrightarrow{\phi}_d(d|d) &= \overrightarrow{f}(d|d) + P_d(d|d)\overrightarrow{\phi}_d(d|d) \\
\overrightarrow{\phi}_d(d|d) &= (I - P_d(d|d))^{-1}\overrightarrow{f}(d|d) \qquad (A.3)
\end{aligned}$$

where $\overrightarrow{\phi}_d = [\phi(s_1,d), \phi(s_2,d), ..., \phi(s_n,d)]^t$, $\overrightarrow{f} = [f_{s_1}, f_{s_2}, ..., f_{s_n}]^t$, and $\overrightarrow{\phi}_d(d|d)$ denotes reduced vector where row d is removed from the vector (similarly square matrices $P_d(d|d)$ and $B_d(d|d)$ denote the reduced matrices where row and column d are removed). Now we can use equation (1) to write the cost as a function of random-walk betweenness.

$$\begin{aligned}
\overrightarrow{\phi}_d(d|d) &= B_d(d|d)\overrightarrow{f}(d|d) \;\; \textit{or} \\
\varphi(s,d) &= \sum_{k=1}^{n} b_{sk}(d)f_k = \sum_{k=1}^{n} b_{sk}(d)\sum_j p_{kj}z(k,j) \qquad (A.4)
\end{aligned}$$

Now we find the average cost over all node pairs.

$$
\begin{aligned}
\bar{\varphi} &= \frac{1}{n(n-1)} \sum_{s,d} \varphi(s,d) \\
&= \frac{1}{n(n-1)} \sum_{k} (\sum_{s,d} b_{sk}(d) \sum_{j} p_{kj} z(k,j)) \\
&= \frac{1}{n(n-1)} \sum_{k} (\sum_{j} p_{kj} z(k,j) b_k)
\end{aligned}
$$

One can use relation (6) to find the relationship between average cost and criticality.

$$
\begin{aligned}
\bar{\varphi} &= \frac{1}{n(n-1)} \sum_{k} (\sum_{j} p_{kj} z(k,j) \frac{1}{2} \tau W_k) \\
&= \frac{1}{2n(n-1)} \tau \sum_{k} (\sum_{j} \frac{w_{kj}}{W_k} z(k,j) W_k) \\
&= \frac{1}{2} \hat{\tau} \sum_{k} (\sum_{j} w_{kj} z(k,j))
\end{aligned}
$$

□

Appendix B. Proof of Theorem 5.4

Since λ_2 is the smallest non-zero eigenvalue of graph Laplacian and all the eigenvalues are non-negative, using equation (40) we have:

$$
\hat{\tau} = \frac{2}{n-1} \sum_{i=2}^{n} \frac{1}{\lambda_i} \leq \frac{2}{n-1} \times \frac{n-1}{\lambda_2} \leq \frac{2}{\lambda_2} \tag{B.1}
$$

This establishes the upper bound for normalized network criticality. To get the lower bound we observe that:

$$
\hat{\tau} = \frac{2}{n-1} \sum_{i=2}^{n} \frac{1}{\lambda_i} \geq \frac{2}{n-1} \frac{1}{\lambda_2} \tag{B.2}
$$

combining inequalities (B.1) and (B.2) completes the proof of theorem 5.4.

□

Appendix C. Proof of Lemma 7.1

We notice that:

$$\begin{aligned}
\tau_\alpha &= \sum_{s,d} \alpha_{sd}\tau_{sd} \\
&= \sum_{ij} \alpha_{sd} u_{sd}^t L^+ u_{sd} \\
&= \sum_{ij} \alpha_{sd} Tr(u_{sd}u_{sd}^t L^+) \\
&= Tr(U_\alpha L^+) \qquad \text{(C.1)}
\end{aligned}$$

where $U_\alpha = \sum_{sd} \alpha_{sd} U_{sd}$ and $U_{sd} = u_{sd}u_{sd}^t$. It is easy to see that U_α is a symmetric matrix with the sum of the entries of its rows equal to zero, and for $\alpha_{sd} \geq 0 \ \forall s, d \in N$, it is a positive semi-definite matrix.

Considering equation (C.1) we have:

$$\begin{aligned}
\frac{\partial \tau_\alpha}{\partial w_{ij}} &= \frac{\partial Tr(U_\alpha \Gamma^{-1})}{\partial w_{ij}} \\
&= -Tr(U_\alpha \Gamma^{-1} \frac{\partial \Gamma}{\partial w_{ij}} \Gamma^{-1}) \\
&= -Tr(U_\alpha \Gamma^{-1} u_{ij} u_{ij}^t \Gamma^{-1}) \\
&= -Tr(F_\alpha^t F_\alpha \Gamma^{-1} u_{ij} u_{ij}^t \Gamma^{-1}) \\
&= -Tr(F_\alpha L^+ u_{ij} u_{ij}^t L^+ F_\alpha^t) \\
&= -Tr((F_\alpha L^+ u_{ij})(F_\alpha L^+ u_{ij})^t) \\
&= -Tr((F_\alpha L^+ u_{ij})^t (F_\alpha L^+ u_{ij})) \\
&= -\left\| F_\alpha L^+ u_{ij} \right\|^2
\end{aligned}$$

where $U_\alpha = F_\alpha^t F_\alpha$. This decomposition is always possible since U_α is a symmetric matrix. Therefore, the first partial derivative of τ_α is always non-positive.

□

Appendix D. Proof of Theorem 7.2

We need the following lemma.

Lemma Appendix D.1. *For any weight matrix W of links of a graph: $Vec(W)^t\nabla\tau+\tau=0$, where $Vec(W)$ is a vector obtained by concatenating all the rows of matrix W to get a vector of w_{ij}'s.*

Proof In [Ghosh et al. (2008)] it has been shown that if we scale all the link weights with t, the effective resistance τ_{ij} will scale with $\frac{1}{t}$. Since τ_α is a linear combination of point-to-point effective resistances, τ_α will also scale with $\frac{1}{t}$:

$$\tau_\alpha(tVec(W)) = \frac{1}{t}\tau_\alpha(Vec(W)) \tag{D.1}$$

By taking the derivative of τ with respect to t, we have

$$Vec(W)^t\nabla\tau_\alpha = \frac{-1}{t^2}\tau_\alpha(W) \tag{D.2}$$

It is enough to consider equation D.2 at $t = 1$ to get $Vec(W)^t\nabla\tau_\alpha + \tau_\alpha = 0$.
□

In general, one can apply the condition of optimality [Bertsekas et al. (2003); Boyd et al. (2004)] on optimization problem (36) to get necessary condition for a weight vector to be optimal. Let W^* be the optimal weight matrix, and let W_t be another weight matrix satisfying the constraints of optimization problem (36), then according to the condition of optimality:

$$\nabla\tau_\alpha.(Vec(W_t) - Vec(W^*)) \geq 0$$

Now, we choose W_t as follows:

$$W_t = [w_{uv}] = \begin{cases} \frac{C}{2z_{ij}} & if \;\; u = i \;\&\; v = j \\ \frac{C}{2z_{ij}} & if \;\; u = j \;\&\; v = i \\ 0 & otherwise \end{cases}$$

Clearly, W_t satisfies the constraints of optimization problem (36), therefore, using the condition of optimality and considering lemma Appendix D.1 we have:

$$\begin{aligned} \nabla\tau_\alpha.(Vec(W_t) - Vec(W^*)) &\geq 0 \\ \nabla\tau_\alpha.Vec(W_t) - \nabla\tau_\alpha.Vec(W^*) &\geq 0 \\ \frac{C}{z_{ij}}\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha &\geq 0 \quad \forall(i,j)\in E \\ \min_{(i,j)\in E}\frac{C}{z_{ij}}\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha &\geq 0 \end{aligned} \tag{D.3}$$

Now, to prove the second part of the theorem we write the constraint of the optimization problem as an inner product of costs and weights.

$$(Vec(Z).Vec(W^*))\tau_\alpha = (\sum_{(i,j)\in E} w^*_{ij} z_{ij})\tau_\alpha = C\tau_\alpha \tag{D.4}$$

Combining lemma Appendix D.1 and equation D.4 one can see:

$$\begin{aligned} C\nabla\tau_\alpha.Vec(W^*) + Vec(Z).Vec(W^*)\tau_\alpha &= 0 \\ Vec(W^*).(C\nabla\tau_\alpha + \tau_\alpha Vec(Z)) &= 0 \\ w^*_{ij}(C\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha z_{ij}) &= 0 \end{aligned}$$

This completes the proof of theorem 7.2.

□

Appendix E. Proof of Lemma 9.3

Hypercube can be recursively built using the Cartesian Product of a graph with K_2 (complete graph on 2 nodes):

$$H_{n+1} = H_n \square K_2$$

where $\square$ denotes the cartesian product. Alternatively, this equation can be written using Kronecker Product:

$$\begin{aligned} H_{n+1} &= H_n \square K_2 \\ &= H_n \otimes I_2 + I_{2^n} \otimes K_2 \end{aligned} \tag{E.1}$$

We have used the symbol $\otimes$ to denote Kronecker product. We try to obtain eigenvalues of adjacency matrix of H_n using equation (E.1).

$$\begin{aligned} H_{n+1} &= H_n \otimes I_2 + I_{2^n} \otimes K_2 \\ &= \begin{pmatrix} H_n & 0 \\ 0 & H_n \end{pmatrix} + \begin{pmatrix} 0 & I_{2^n} \\ I_{2^n} & 0 \end{pmatrix} \\ H_{n+1} &= \begin{pmatrix} H_n & I_{2^n} \\ I_{2^n} & H_n \end{pmatrix} \end{aligned}$$

For the sake of simplicity, we drop the subscript from I_{2^n} and use I instead, which means the identity matrix of appropriate order. Now we try to build the determinant of characteristic matrix $H_{n+1} - \lambda I$:

$$\begin{aligned} H_{n+1} - \lambda I &= \begin{pmatrix} H_n - \lambda I & I \\ I & H_n - \lambda I \end{pmatrix} \\ d_{n+1} &= |H_{n+1} - \lambda I| = \det \begin{pmatrix} H_n - \lambda I & I \\ I & H_n - \lambda I \end{pmatrix} \\ &= \det \begin{pmatrix} I & H_n - \lambda I \\ H_n - \lambda I & I \end{pmatrix} \end{aligned}$$

Now we multiply the first row by $H_n - \lambda I$, and then subtract the first row from the second row. We have:

$$\begin{aligned} d_{n+1}(\lambda) &= \det \begin{pmatrix} I & H_n - \lambda I \\ H_n - \lambda I - (H_n - \lambda I) & I - (H_n - \lambda I)^2 \end{pmatrix} \\ &= \det \begin{pmatrix} I & H_n - \lambda I \\ 0 & I - (H_n - \lambda I)^2 \end{pmatrix} \\ &= |I - (H_n - \lambda I)^2| \\ &= |H_n - (\lambda - 1)I||H_n - (\lambda + 1)I| \\ d_{n+1}(\lambda) &= d_n(\lambda - 1) d_n(\lambda + 1) \end{aligned}$$

Using this recursive formula for determinant of hypercube, one can find with induction that the eigenvalues of H_n are $2k - n, \quad k = 0, 1, ..., n$ with multiplicity $C(n, k) = \frac{n!}{k!(n-k)!}$.

□

References

A. H. Dekker and B. D. Colbert. Network Robustness and Graph Topology. *Australasian Computer Science Conference*, 26:359–368, Jan. 2004.

R. Zhang-Shen and N. McKeown. Designing a Predictable Internet Backbone with Valiant Load-Balancing. In *Thirteenth International Workshop on Quality of Service (IWQoS)*, Passau, Germany, June 2005.

L. C. Freeman. Centrality in Networks: I. Conceptual Clarification. *Social Networks*, (1):215–239, 1978/79.

L.C. Freeman, S.P. Borgatti, and D.R. White. Centrality in valued graphs: a measure of betweenness based on network flow. *Social Networks*, 13:141–154, 1991.

L. S. P. Borgatti. Centrality and Network Flow. *Social Networks*, 27(1):55–71, 2005.

M. Newman. A Measure of Betweenness Centrality Based on Random Walks. *arXiv cond-mat/0309045.*, 2003.

A. Tizghadam and A. Leon-Garcia. A Graph Theoretical Approach to Traffic Engineering and Network Control Problem. In 21^{th} *International Teletraffic Congress (ITC21)*, Paris, France, September 2009.

C. R. Rao and S. K. Mitra. *Generalized Inverse of Matrices and its Applications*. John Weily and Sons Inc., 1971.

D. J. Klein and M. Randic. Resistance Distance. *Journal of Mathematical Chemistry*, 12(1):81–95, December 1993.

A. Ghosh, S. Boyd, and A. Saberi. Minimizing Effective Resistance of a Graph. *SIAM Review, problems and techniques section*, 50(1):37–66, February 2008.

A. Tizghadam and A Leon-Garcia. Survival Value of Communication Networks. In *INFOCOM Workshop on Network Science for Communications (NetSciCom)*, pages 1–6, Rio de Janeiro, Brazil, April 2009.

L. Kleinrock. *Queueing Systems*, volume II. John Wiley & Sons, 1975.

Dennis S. Bernstein. *Matrix Mathematics*. Prinston University Press, 2005.

M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming (Web Page and Software). $http://stanford.edu/boyd/cvx$. September 2008.

M. Grant and S. Boyd. Graph Implementations for Nonsmooth Convex Programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, $http://stanford.edu/boyd/graph_dcp.html$. *Lecture Notes in Control and Information Sciences, Springer*, 2008.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

D. Eppstein. Finding the K-Shortest Paths. *Society for Industrial and Applied Mathematics (SIAM) Journal of Computing*, 28(2):652–673, 1998.

A. Tizghadam and A. Leon-Garcia. A Robust Routing Plan to Optimize Throughput in Core Networks. *ITC20, Elsvier*, pages 117–128, 2007.

M. Ziegelmann. *Constrained Shortest Paths and Related Problems - Constrained Network Optimization*. VDM Verlag, 2007.

G. Leduca, H. Abrahamssone, S. Balona, S. Besslerb, M. D'Arienzoh, O. Delcourta, J. Domingo-Pascuald, S. Cerav-Erbasg, I. Gojmeracb, X. Masipd, A. Pescaph, B. Quoitinf, S.P. Romanoh, E. Salvadoric, F. Skivea, H.T. Tranb, S. Uhligf, and H. Umitg. An Open Source Traffic Engineering Toolbox. *Computer Communications*, 29(5):593–610, March 2006.

K. Kar, M. Kodialam, and T. V. Lakshman. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. *IEEE Journal on Selected Areas in Communications*, 18(12):2566–2579, Dec. 2000.

D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, April 2003.

A. Tizghadam and A. Leon-Garcia. On Robust Traffic Engineering in Core Networks. In *IEEE GLOBECOM*, December 2008.

A. Tizghadam and A. Leon-Garcia. On Congestion in Mission Critical Networks. In *Workshop on Mission Critical Networks (MCN), INFOCOM*. IEEE, April 2008.

A. Tizghadam and A. Leon-Garcia. Autonomic Traffic Engineering for Network Robustness. *IEEE Journal of Selected Areas in Communications (J-SAC)*, 28(1):39 – 50, January 2010.

A. Tizghadam and A. Leon-Garcia. On Robust Network Planning. 7^{th} *Workshop on Designing Reliable Communication Networks (DRCN)*, October 2009.

M. Fiedler. Algebraic Connectivity of Graphs. *Czechoslovak Math. Journal*, 23(98):298–305, 1973.

L. Kleinrock. *Communication Nets, Stochastic Message Flow and Delay*. McGraw-Hill, New York, 1964.

B. Meister, H. R. Muller, and H. R. Rudin. New optimization criteria for message switching networks. *IEEE Transactions on Communication Technology*, 19(3):256–260, June 1971.

N. Spring, R. Mahajan, and D. Wetherall, T. Anderson. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Transactions on Networking (TON)*, 12(1):2-16, February 2004.

D. Applegate, E. Cohen. Making routing robust to changing traffic demands: algorithms and evaluation. *IEEE/ACM Transactions on Networking (TON)*, 14(6):1193 - 1206, December 2006.

A. Tizghadam, A. Leon-Garcia. On Random Walks in Direction-Aware Network Problems. *ACM SIGMETRICS Twelfth Workshop on Mathematical Performance Modeling and Analysis (MAMA)*, New York, USA, June 2010.