Betweenness Centrality and Resistance Distance in Communication Networks

Ali Tizghadam and Alberto Leon-Garcia, University of Toronto

Abstract

In this article we report on applications and extensions of weighted graph theory in the design and control of communication networks. We model the communication network as a weighted graph and use the existing literature in graph theory to study its behavior. We are particularly interested in the notions of betweenness centrality and resistance distance in the context of communication networks. We argue that in their most general form, the problems in a communication network can be converted to either the optimal selection of weights or optimal selection of paths based on the present values of weights in a graph. Motivated by this, we propose a two-loop general architecture for the control of networks and provide directions to design appropriate control algorithms in each control loop. We show that the total resistance distance (network criticality) of a graph has very useful interpretations in the context of communication networks; therefore, we propose to use network criticality as the main objective function, and we provide guidelines to design the control loops to minimize network criticality. We also discuss the development of new directed weighted graph models and their application to communication networks.

Many current problems in communication networks can be modeled as graphs with some dynamism due to changes in traffic demand and patterns as well as faults. Therefore, one can apply and extend graph theoretic results to develop design and control algorithms for communication networks. We model a communication network as a weighted graph. We are interested in the allocation of weights to links to optimize a variety of network goals. For example, we are concerned with the selection of paths to transfer traffic between source-destination pairs in the context of a specific community of interest. We are also interested in the allocation of capacities to links in a network, the design of network topologies, and the dynamic control of virtual and physical network topologies to achieve communication goals, such as optimizing robustness. In this article we describe how the existing literature in graph theory, in particular centrality measures, can be used to provide frameworks for the development of a rich set of network control and design algorithms.

In the next section we review the notion of betweenness centrality in graphs and discuss its relationship to random walks. In the following section we first define resistance distance (effective resistance) and network criticality in terms of betweenness. Then we elaborate on the relationship between random walks on undirected graphs and resistive electrical circuits, and we explain how these motivate the use of network criticality as a key network design parameter. We then survey the wide range of network control and design problems that can be addressed. We also indicate promising directions for the further development of this approach. The final section gives concluding remarks.

Betweenness Centrality Measures

Shortest-path betweenness centrality is a very useful metric in graph theory. For node k the shortest-path betweenness centrality with respect to flows from source node s to destination node d is defined as the proportion of instances of the shortest paths from node s to d that traverse node k. The overall shortest-path betweenness centrality of node k is the sum of the centralities over all source-destination pairs. Link betweenness is defined similarly. A major drawback of shortest-path betweenness is that it uses only a subset of available information on the graph. Moreover, it is frequently desirable to take a path other than the shortest path. To overcome this issue other betweenness centrality metrics have been proposed. Flow betweenness centrality is one of the important extensions to shortest-path betweenness. The flow betweenness of a node k is defined as the proportion of flow through node k when maximum flow is transmitted from source s to destination d averaged over all s-d pairs. Flow betweenness considers more diverse paths; however, it suffers from the same problem (i.e., it considers only a subset of available paths).

Random-walk betweenness is a probabilistic and tractable approach to define and analyze betweenness. We suppose that the path selected corresponds to the path traversed by a random walk in a graph. We assume that the graph G consists of a finite node set V which contains n nodes and a set of links E, where link (i, j) has a non-negative symmetric weight w_{ij}. The random walk picks its next node in the graph according to transition probability matrix $P = [p_{ij}]$. In this article we consider a connected graph and use the weighted random walk whose transition probability of link (i, j) is defined as

 0890-8044/10/$25.00

$$p_{ij} = \frac{w_{ij}}{W_i^o},$$

where W_i^o is the sum of the weights for links exiting node i. For an undirected graph with symmetric weights the out-weight of a node is equal to its total weight because a link can be considered as either outgoing or incoming, and the corresponding Markov chain is reversible. Before we proceed, we introduce two Laplacian matrices, which will be referred to later. We define the standard graph Laplacian as $L_w = D_o - W$, where D_o is a diagonal matrix whose main diagonal entries are $D_o(i, i) = W_i^o$. We define the combinatorial Laplacian of a graph as $L = \Phi(I - P)$, where Φ is a diagonal matrix with main diagonal equal to the stationary probability vector of P.

To define the random-walk betweenness, consider the set of trajectories in the transient random walk that begins at node s and terminates when the walk first arrives at node d (i.e., 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 number of times node k is visited in trajectories from s to d. For any given destination d, the matrix $B_d = [b_{sk}(d)]$ is easily found and is given by $B_d = (I - P_d)^{-1}$, where P_d is the transition probability matrix with the dth row and column replaced by zero vectors. Clearly, nodes that have a high (low) tendency to appear in trajectories from s to d will have a high (low) value of betweenness. The overall random-walk betweenness b_k of node k is $b_k = \sum_{s,d} b_{sk}(d)$.

Random-walk betweenness treats all source-destination pairs equally and does not account for differences in traffic volume or community of interest. Traffic-aware betweenness is a natural extension to explicitly involve the effect of traffic demands in the definition of betweenness centrality index [1]. Let γ_{sd} denote the traffic demand from source node s destined for node d. The traffic-aware betweenness (for any type of betweenness) of node k is defined as

$$b'_k = \sum_{s,d}\left(1 + \frac{\gamma_{sd}}{\gamma}\right) b_{sk}(d),$$

where $\gamma = \sum_{s,d}\gamma_{sd}$.

We refer the reader to [2, references therein] for an excellent review of all betweenness centrality measures.

Betweenness Centrality in Networking Literature

We close this section by reviewing the existing literature on using betweenness centrality in communication networks. There is a recent trend in the networking community to use centrality metrics in modeling the performance of communication networks. For instance, recent attempts to uncover a hidden stable network structure in delay-tolerant networks (DTNs) using ideas from graph theory and social networks resulted in introducing the SimBet routing algorithm [3], where a local ego-centric betweenness measure is used to find the nodes with higher centrality (relay nodes). Messages are forwarded toward the relay nodes to increase the possibility of finding the potential carrier to the final destination. A recent work [4] investigates the effect of link weight distribution on the distribution of link betweenness centrality in a data network. It is shown that for a variety of weight distributions (e.g., uniform, exponential, and power law), the distribution of link betweenness centrality remains power law. Using this result, the authors in [4] consider a polynomial link weight distribution and provide conditions under which most of the traffic of a network flows along a minimum spanning tree (MST). Finally, in [5] a new metric, routing betweenness centrality (RBC), is proposed that generalizes previous betweenness measures by considering network flows created by arbitrary loop-free routing strategies. It is anticipated that RBC will be useful for prediction of congestion in communication networks, design and examination of routing strategies, and optimizing network layout for balancing the traffic load in a network.

Resistance Distance and Network Criticality

If we view the weight of a link as a measure of capacity, the ratio of link (node) betweenness (i.e., traffic flow) to link (node) weight makes intuitive sense as a measure akin to utilization. It can be shown that for any undirected graph, the ratio of the random-walk betweenness of a node to the node weight (the sum of the weights of all incident links) is the same for all nodes in the graph. This ratio, which we define as *network criticality*, is therefore a single metric that characterizes a weighted graph and its corresponding communication network.

There is a nice interpretation of network criticality in the context of electrical circuits. Resistance distance [6] is a well-known measure in electrical circuits. Suppose there is a unit current source at node s and that node d is grounded. Of particular interest is the equivalent resistance (point-to-point resistance distance) seen between s and d (denoted τ_{sd}) which is equal to the voltage at node s (divided by the unit current) [2, 6]. It can be shown that the network criticality is equal to the total resistance distance (i.e., $\sum_{s,d}\tau_{sd}$) of the network if we interpret the network as a resistive electrical circuit with link conductances equal to the corresponding link weights in the original network. For the comparison of graphs with different numbers of nodes, we usually use the normalized total resistance distance (or network criticality)

$$\hat{\tau} = \frac{1}{n(n-1)}\sum_{s,d}\tau_{sd}.$$

It can be shown that network criticality ($\hat{\tau}$) is proportional to the trace of the generalized inverse standard Laplacian, and is a monotone decreasing and strictly convex function of link weights [7, 8].

Network criticality provides a natural measure for the robustness of a network to changes in traffic, topology, and community of interest. This becomes evident by interpreting network criticality as the total resistance of a circuit. For two circuits with the same number of nodes, the one with the smaller total resistance is clearly better connected than the one with higher total resistance, and hence better positioned to accommodate higher current flows. Therefore, network criticality provides a useful way of characterizing the robustness of a graph and its associated network.

We can extend the definition of network criticality to account for the effect of non-uniform external traffic demands by simply replacing the random-walk betweenness with its traffic-aware version. It can be shown that the resulting traffic-aware node criticality (TANOC) is a linear combination of point-to-point resistance distances, and it is convex with respect to the link weights. TANOC is not the same for all nodes, so we use the average of TANOC over all the nodes as a global convex graph/network measure in non-uniform traffic scenarios, and we refer to it as traffic-aware network criticality (TANC) [1].

Resistance Distance in Networking Literature

The technique of effective resistance (resistance distance) has seen growing popularity in control theory and communication networks. In [9] an algorithm for distributed clock synchronization in multihop wireless networks is analyzed that relies on minimum variance estimation. The authors show that the calculation of variance can be converted to finding the resistance distance of an equivalent electrical circuit. The minimization of total resistance distance as a function of link weights is discussed in [8], and a numerical algorithm to find the optimal weights is proposed. The clock synchronization problem can be solved effectively with this algorithm. Another recent work [10] extends the notion of effective resistance to cases where the weight of a link is a matrix. The article studies a problem from sensor networks that involves estimating a number of node variables from measurements of the noisy differences between them. It is shown that the covariance matrix of the measurement error equals the extended matrix form of the effective resistance. Moreover, [10] calculates the extended effective resistance of some d-dimensional ($1 \leq d \leq 3$) dense and sparse networks, and suggests an optimal network configuration in sensor (and ad hoc) networks to minimize the measurement error. Finally, the problem of finding robust network topologies is addressed in [11] by investigating the following question: given an initial graph topology, how should we add k edges so that the resulting graph topology has the maximal number of spanning trees among all possible topologies? It is shown that in the optimal network the point-to-point resistance distance of every two pairs should have the same value. Based on this fact, a relaxed convex optimization problem is proposed to maximize the number of spanning trees.

Interpretations of Network Criticality

In this section we shed more light on the importance of network criticality in the study of communication networks by providing some of its interpretations.

Average Travel Cost — Suppose that there are costs associated with traversing links along a path, and consider the effect of network criticality on average cost incurred by a message during its walk from source s to destination d. It is shown in [1, references therein] that the average incurred cost is the product of network criticality and the total cost of all link weights. Therefore, if we set a fixed maximum budget for the cost of assigning weights to links, the average travel cost is minimized when network criticality ($\hat{\tau}$) is minimized. In the special case where all links have unit cost, the average travel cost is equal to the average hop length (or average travel time). Hence $\hat{\tau}$ quantifies average path length.

Average Link Betweenness Sensitivity — We are interested in identifying situations where slight changes in the betweenness of a link or node may cause dramatic changes in the betweenness values elsewhere in a network. Thus, a reasonable goal in designing robust network control algorithms is to minimize the changes in the betweenness of different links. We have shown [1] that $\hat{\tau}$ is equal to the average of random-walk link betweenness sensitivities (derivative of a link betweenness with respect to its weight). Therefore, one needs to control the value of $\hat{\tau}$ to keep the change in the average link betweenness sensitivity below a prespecified level.

Connectivity — Traditionally, graph connectivity is quantified with the node (link) connectivity, which gives the number of nodes (links) that must be removed in order to disconnect the graph. But node (link) connectivity is insensitive to the link weights, so it may not be a suitable metric to study the connectivity features of a weighted graph. Algebraic connectivity is another well-known network connectivity metric and is defined as the second smallest eigenvalue (denoted λ_2) of the standard Laplacian matrix (L_w) of an undirected graph. Note that the eigenvalues of the standard Laplacian matrix of an undirected graph are nonnegative real numbers and the smallest eigenvalue is always zero. Algebraic connectivity (λ_2) is positive if and only if the graph is connected. More generally, the number of times zero appears as an eigenvalue in the standard Laplacian equals the number of connected components in the graph. As a result, in connected undirected networks (i.e., networks with one connected component) the smallest eigenvalue is zero, and the rest (including λ_2) are real and positive.

In an unweighted graph, algebraic connectivity is a lower bound for node (link) connectivity; consequently, a higher value of λ_2 corresponds to higher connectivity. For weighted graphs it can be shown that the derivative of algebraic connectivity with respect to link weights is always nonnegative (i.e., a higher value of link weights results in better connectivity). Furthermore, the distribution of the graph's Markov process converges to the stationary vector at a rate determined by the algebraic connectivity. It can be shown that $\hat{\tau}$ is upper bounded by the reciprocal of λ_2 [1]. This means that highly connected graphs have a smaller upper bound for network criticality; therefore, the connectivity of a network can be controlled by $\hat{\tau}$.

Network criticality carries more information about the structure of the network than λ_2. To see this we run an experiment on an extended linear graph (ELG) and flat grid graph (FGG) (Fig. 1a and 1b), and we compare the behavior of $\hat{\tau}$ and λ_2 for different network sizes. In this experiment we assume all the link weights are equal to 1 (unweighted). Figure 1c shows that the criticality of ELG grows much faster than that of FGG. Figure 1d reveals that the flat grid has better connectivity, but the speed of decrease in the connectivity of the graph is much slower than the increase in $\hat{\tau}$. The observation in this experiment is that while the changes in algebraic connectivity of ELG and FGG are relatively similar, there is a huge change in the behavior of $\hat{\tau}$, which means that $\hat{\tau}$ captures some robustness properties of the graph that cannot be found in λ_2. This experiment also reveals that increasing the topological dimension of a network will increase its robustness. Note that FGG expands in two dimensions while ELG grows in one dimension. A more detailed comparison of algebraic connectivity and network criticality for unweighted graphs is discussed in [12].

Congestion — $\hat{\tau}$ can determine the onset of congestion in a communication network. Let λ be the average total input rate of the network, and let the weight of each link be the capacity of the link. It can be shown that the maximum acceptable value of λ before the capacity of some link in the network is exceeded (congestion) is upper bounded by the reciprocal of network criticality [1]. One concludes that to delay the onset of congestion to the maximum extent possible, we need to minimize $\hat{\tau}$.

Average Network Utilization — Let the link weights be the capacity of the links in the network and a traffic demand matrix is given; then the average network utilization is proportional to the difference of TANC and network criticality [1].

Network Design and Control

The interpretations introduced in the previous section suggest that $\hat{\tau}$ (and TANC) can play a useful role in the study of communication networks. In this section we present specific examples that show how $\hat{\tau}$ can be used to design optimal networks and synthesize network controls.

Figure 2 shows a general block diagram for controlling a network using two control loops. A slow loop is concerned with changes in provisioned link capacity and in topology and so is concerned with the optimal assignment of weights in a

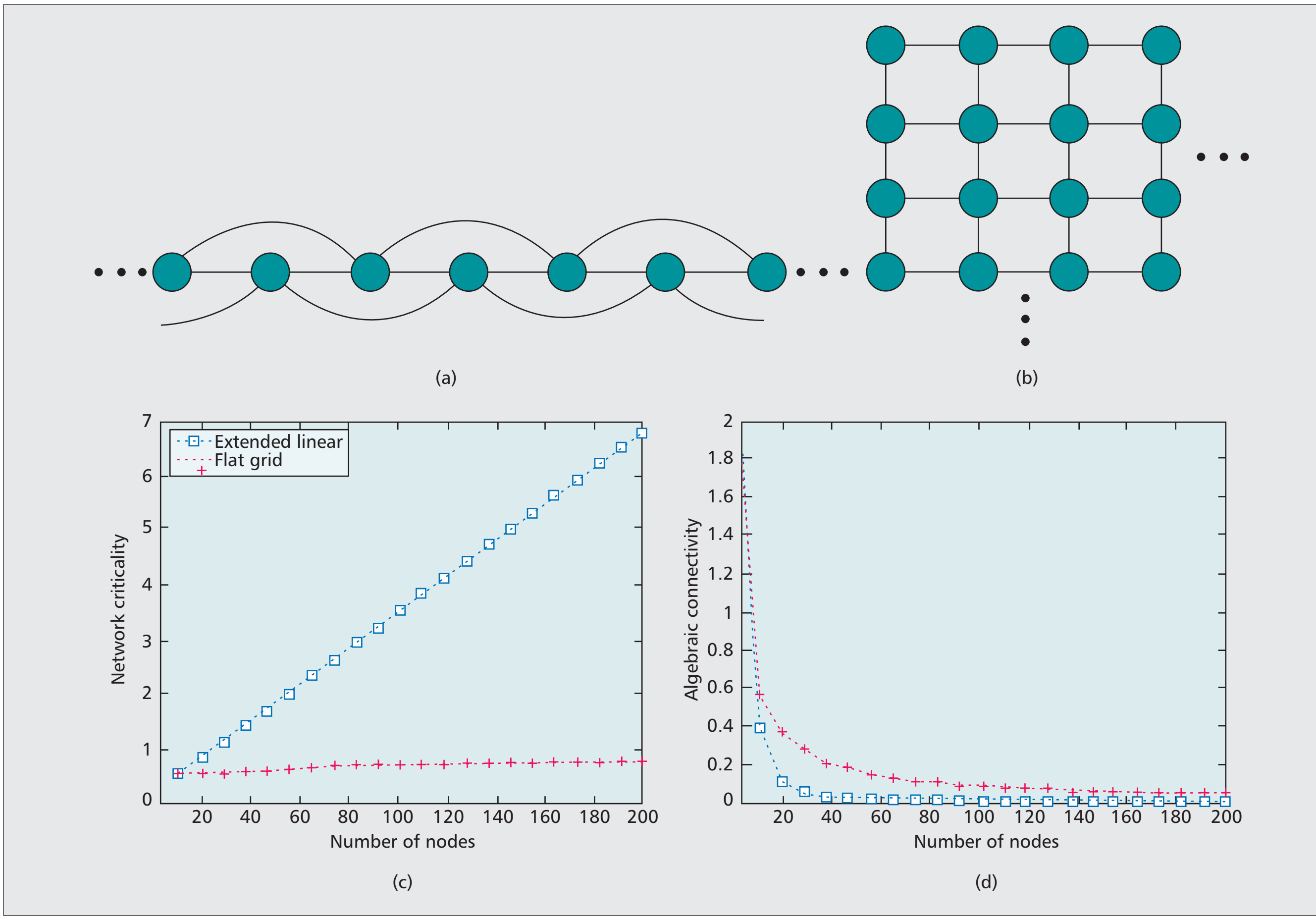


Figure 1. *Behavior of network criticality ($\hat{\tau}$) and algebraic connectivity (λ_2) for ELG and FGG: a) ELG; b) FGG; c) variations of $\hat{\tau}$; d) variations of λ_2.*

graph model for the network. On the other hand, a fast loop regulates short-term performance through path selection and routing decisions that are determined by applying incremental changes in weight values. Network criticality can play a key role in these control loops by serving as the objective function that guides network control and design decisions. In the remainder of this section we focus on the optimization problems that arise in the control architecture.

Slow-Loop and Network Design

Motivated by the properties and interpretations of $\hat{\tau}$ and TANC we use these as the objective function in this work. $\hat{\tau}$ is a decreasing and strictly convex function of link weights, so it can be readily minimized subject to appropriate constraints. For example, we can address the capacity allocation problem where we have a maximum budget C for purchasing link weights (interpreted as link capacities in this case) that have cost z_{ij} per unit of weight for link (i, j) (i.e., $\sum_{i,j} z_{ij} w_{ij} \leq C$). We can then pose the network criticality optimization problem:

$$\begin{aligned} &\textit{Minimize} \quad \hat{\tau} \\ &\textit{Subject to} \quad \sum\nolimits_{(i,j)\in E} w_{ij} z_{ij} \leq C \quad ,C \textit{ is fixed} \\ &\qquad w_{ij} \geq 0 \, \forall (i,j) \in E \end{aligned} \tag{1}$$

The complete analysis of this problem is given in [7, 8]. Here we simply state the main result, which is very useful for designing algorithms for the fast and slow loops of our architecture. Let w^*_{ij} be the optimal weight of link (i, j) and $\hat{\tau}^*$ be the minimum value of $\hat{\tau}$; then for any suboptimal solution of the convex optimization problem ($\hat{\tau}$), the deviation from optimal solution (optimality gap) has the upper bound

$$\frac{\hat{\tau} - \hat{\tau}^*}{\hat{\tau}} \leq 1 + \frac{\hat{\tau}}{C \min_{(i,j)\in E} \dfrac{1}{z_{ij}} \dfrac{\partial \hat{\tau}}{\partial w_{ij}}}. \tag{2}$$

Later we show how Eq. 2 is used to develop path selection algorithms for the fast loop portion of the control architecture.

The traffic-aware version of the optimization in Eq. 1 proceeds as follows. We replace $\hat{\tau}$ with TANC, and wherever appropriate add flow conservation constraints. Link flow variables are necessary when we aim to jointly optimize routing and resource allocation.

Next we discuss solutions to optimization Eq. 1 for various network scenarios and topologies. More details on fast and slow control loops can be found in [13].

Parking Lot Network — Consider the parking lot network in Fig. 3a. First suppose that there is traffic only between two specific nodes a and b. Suppose $a = 4$ and $b = 9$, the costs of all the links are set to 1, and the budget is $C = 2000$ (w_{ij} and w_{ji} are considered separately); then the optimization problem assigns equal weights along the shortest path from a and b as indicated in Fig. 3a by thick lines. Next suppose that all pairs in the network have the same volume of traffic between them. The optimum weight assignment, shown in the second column of the table in Fig. 3b, then assigns higher weights to the bottleneck links (3, 5), (5, 8), and (8, 10). Finally, consider a nonuniform traffic matrix given by

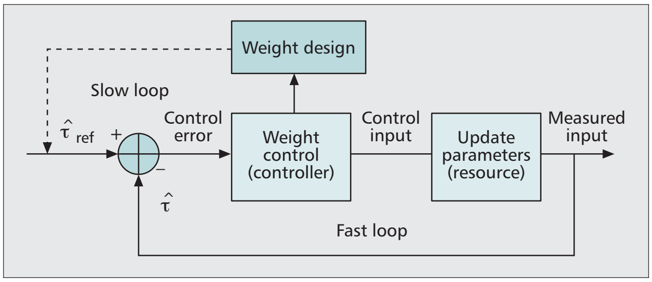


Figure 2. *Block diagram of two-loop weight control system.*

$$\gamma_{ij} = \frac{1}{(i+j)^2} \forall i, j \in N, \ i \neq j.$$

In this case the nodes with lower indices have higher traffic volumes flowing between them. The third column of the table in Fig. 3b 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 over all *s-d* pairs subject to a cost constraint. The fourth column of the table in Fig. 3b shows the weight assignment that achieves the minimax value of point-to-point resistance distance (the value is 0.0455) for the network in Fig. 3a.

Kleinrock's Minimum Delay Capacity Assignment — We now compare the minimum network criticality weight assignment to Kleinrock's minimum average delay capacity assignment and Meister's extension (see [1, references therein] for a more detailed discussion on Kleinrock's and Meister's methods). We use the example network in Fig. 4a, which is used by Kleinrock and Meister. We suppose that all link costs are equal. Kleinrock's method finds the link capacities that minimize the average delay of the network when a routing algorithm is already in place. 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. Figure 4c compares the capacity assigned to the links using Kleinrock's, Meister's, and the minimum network criticality methods. The capacity assignments for links 3, 5, and 7 show that the methods yield very different solutions. Figure 4d shows the individual link delays for the three methods; again, links 3, 5, and 7 show wide disparity between Kleinrock's and Meister's methods, with network criticality somewhere in between. Figure 4b shows the average network delay for the three methods; not surprisingly, the network criticality method strikes a balance between the the other two methods.

Hypercube with 2^n nodes (H_n) — Consider a hypercube network of order n (H_n). A hypercube network is a potential choice for use in the core of a data center assuming that there is a load balancing mechanism to uniformly distribute the traffic that flows from the edge into the core of the data center. It can be shown that for uniform traffic, the total resistance of the hypercube is minimized when all the link weights are equal. The optimal average network criticality (average resistance distance) of a hypercube is obtained in [1], where it is shown that the ratio of the optimal network criticality for H_n to the optimal network criticality of a complete graph with the same size reaches a maximum for small values of n (to be more precise, for $n = 4$ we have the maximum ratio, which is approximately 1.22). When the network size increases the ratio decreases, and for sufficiently high values of n the ratio approaches 1. However, the optimal network criticality of a hypercube is always significantly higher than that of a complete graph of the same size.

Fast Loop Path Selection and Routing

The fast loop in the proposed control architecture acts on a shorter time horizon and is responsible for dynamic resource allocation such as path selection. The main goal here is to find appropriate paths to place the flow requests so that the change in $\hat{\tau}$ is minimized.

End-to-End Path Selection Methods — We now design a path selection method using the upper bound for the optimality gap in Eq. 2. For simplicity assume that $z_{ij} = 1$ for all links; then the upper bound becomes

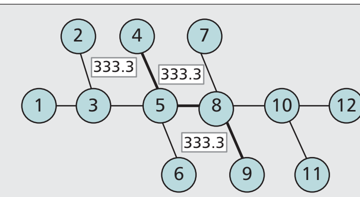
$$1 + \frac{\hat{\tau}/C}{\min_{(i,j)\in E} \frac{\partial \hat{\tau}}{\partial w_{ij}}} = 1 - \frac{\hat{\tau}/C}{\max_{(i,j)\in E} \left| \frac{\partial \hat{\tau}}{\partial w_{ij}} \right|},$$

where we used the fact that $\hat{\tau}$ is a decreasing function of the link weight. The above expression shows that the upper bound is determined by the link with the highest sensitivity of $\hat{\tau}$ with respect to weight. This suggests that the path selection algorithm should avoid links with high sensitivity. We have explored a variety of path selection algorithms and found that the preferred algorithm in terms of balancing performance and implementation is one that selects the shortest path (using Dijkstra's or another algorithm) with the link metric given by

$$1 - \frac{\hat{\tau}/C}{\left| \frac{\partial \hat{\tau}}{\partial w_{ij}} \right|}.$$

The length of a path is then

$$h - (\hat{\tau}/C) \sum \frac{1}{\left| \frac{\partial \hat{\tau}}{\partial w_{ij}} \right|},$$

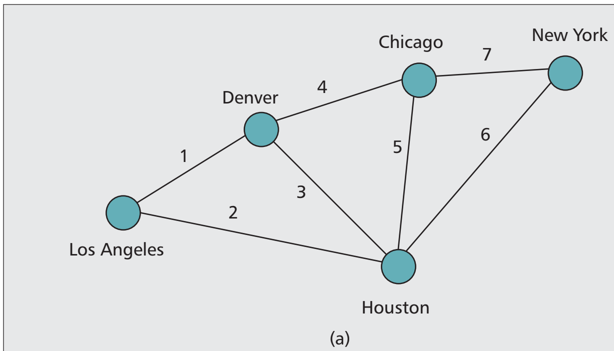


(a)

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

(b)

Figure 3. *Optimal weights for parking-lot: a) parking-lot topology; b) optimal weight in different scenarios.*



Method	Average network delay	Network criticality
Kleinrock	44.72	1.06
Meister	55.01	0.80
Criticality method	49.30	0.56

(b)

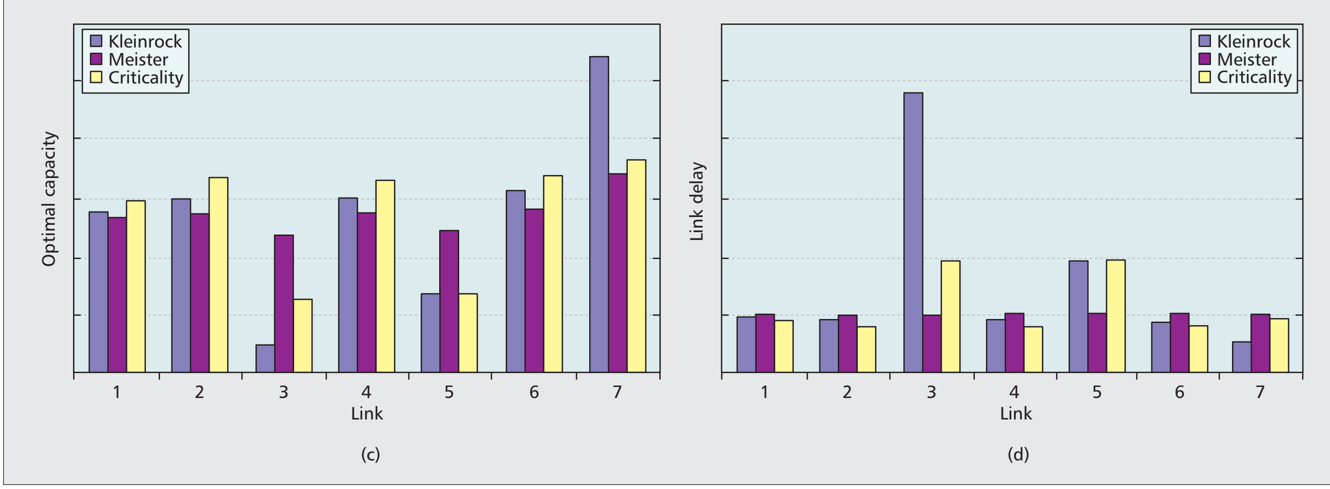


Figure 4. *Kleinrock's network and optimal capacity assignment: a) network topology; b) network delay; c) optimal link capacities; d) link delays.*

where summation is over all the links in the path and h is the number of hops in the path. Therefore, this path selection algorithm can be viewed as modifying the minimum hop algorithm using a correction term based on link sensitivities. We refer to this algorithm as weighted random-walk path criticality routing (WRW-PCR).

We evaluated the performance of the path selection algorithm using an anonymized version of a real network (22 nodes, 68 links) when a link's available capacity is considered as the link weight. We use the algorithm to select paths in an multiprotocol label switching (MPLS) network. First we study the effect of link failure on a network where paths have been selected using our algorithm. We fail 10 links in the network and measure the utilization of all links after rerouting of the paths affected by link failures. For comparison we repeat the experiment for a network in which paths are selected using a Constrained Shortest Path First (CSPF) algorithm, where the constraint in our case is the minimum bandwidth required per link. Figure 5a shows that failures induce fewer overloaded links for our algorithm than for CSPF.

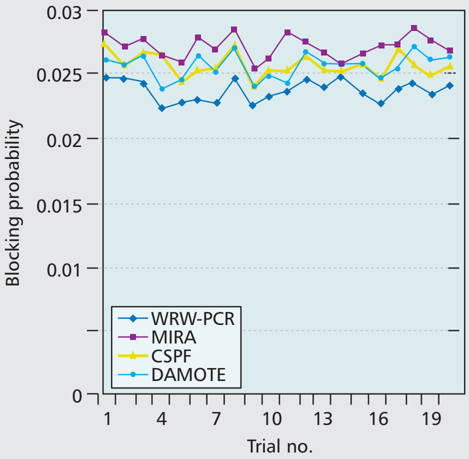
In a second experiment we compare the blocking probability of different path selection methods. We assume that requests for paths between random *s-d* pairs arrive according to a Poisson process with an average rate λ, that holding times are exponentially distributed with mean μ, and that bandwidth requests are uniformly distributed between 1 and 3 units. We set $\lambda/\mu = 1800$ in our experiments (this provides a *heavily loaded* scenario in our test). We generate 10,000 requests and count the number of requests rejected. Figure 5b compares the performance of our proposed method (WRW-PCR) with the Minimum Interference Routing Algorithm (MIRA), a well-known MPLS path setup algorithm, Decentralized Agent for Online MPLS Traffic Engineering (DAMOTE), and CSPF (a nice review of MIRA, DAMOTE, CSPF, and other traffic engineering algorithms is discussed in [14]). The test is performed 20 times.

One can see that the proposed method consistently has the best performance.

Hop-by-Hop Methods — In some communication networks routing is achieved through hop-by-hop methods. This approach is particularly useful when the network topology is not stable. Minimizing network criticality is a useful strategy in this case, but the algorithm must be implemented in a distributed manner using local computations at the nodes. As an example, consider routing in ad hoc networks when only some of the geographic locations of the nodes are known. There are effective methods for geographic routing when all the node locations are known, but in most practical situations this is not the case. In recent proposals the geographic information of only some specific nodes (boundary nodes) is given. The geographic routing algorithm uses a set of virtual x-y coordinates, which are obtained by averaging the coordinates of neighboring nodes while the coordinates of boundary nodes remain fixed. By developing nodal equations it can be shown that this averaging problem solves Dirichlet's equation with boundary conditions [15]. Therefore, it is not surprising that this geographic routing is related to the concept of resistance distance. In fact, it can be shown that the value of geographic coordinates is determined by a convex combination of the boundary nodes. The location information is propagated among the internal nodes, first getting into the neighbors of the boundary nodes and then propagating through the graph according to the resistance distance between the neighbors of the boundary nodes and the rest of the internal nodes.

Link utilization	CSPF	WRW-PCR
>60%	17	4
>70%	5	2
>80%	3	1
>90%	2	0

(a)



(b)

Figure 5. *Result of applying WRW-PCR, MIRA, and CSPF to the network under test: a) failure test (link utilization); b) blocking probability test.*

Network Problems Involving Directed Weighted Graphs

All the approaches discussed so far have assumed weighted undirected graph models. 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), so we cannot model such a network with an undirected graph.

While the analogy between resistance distance and random walks does not hold in directed graphs, we can still find the hitting times and commute times for a directed graph, and the interpretation of average travel cost (or equivalently average commute time) for $\hat{\tau}$ 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 [16], that is, the trace of generalized inverse of the combinatorial Laplacian matrix of a directed graph (L) defined earlier. We propose to use the average travel time as the objective of our optimization problem in the case of directed graphs.

Conclusion

In this article we propose to use the concept of resistance distance and betweenness centrality from graph theory to model the behavior of a communication network. We model a network as a weighted graph and introduce a two-loop control architecture to design and manage the changes in network weights with minimizing total resistance distance as the main objective.

The proposed framework can easily be used in networks modeled by undirected graphs, but many communication networks are direction-aware, particularly when we have wireless mobile nodes with an interference effect. The algorithms presented in our two-loop architecture are mostly useful in undirected cases. We provide some initial thoughts on modeling these directed networks, but further work is required. In particular, the mathematical properties of the objective function needs to be investigated so that suitable algorithms for direction-aware networks can be developed for the fast loop and slow loop of the proposed network control architecture.

References

[1] A. Tizghadam and A. Leon-Garcia, "Robust Network Planning in Nonuniform Traffic Scenarios," to appear, *Comp. Commun.*

[2] U. Brandes and T. Erlebach, Eds., *Network Analysis: Methodological Foundations*, LNCS, vol. 3418, Springer-Verlag, 2005.

[3] E. M. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," *Proc. 8th ACM MobiHoc*, Montreal, Canada, Sept. 2007, pp. 32–40.

[4] H. Wang, J. M. Hernandez, and P. Van Mieghem, "Betweenness Centrality in a Weighted Network," *Physical Rev. E*, vol. 77, 2008, paper no. 046105.

[5] S. Dolev, Y. Elovici, and R. Puzis, "Routing Betweenness Centrality," *J. ACM*, vol. 57, no. 4, Apr. 2010.

[6] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*, Mathematical Assn. America, 1984.

[7] A. Tizghadam and A. Leon-Garcia, "A Graph Theoretical Approach to Traffic Engineering and Network Control Problem," *Proc. 21st Int'l. Teletraffic Cong.* , Paris, France, Sept. 2009.

[8] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing Effective Resistance of a Graph," *SIAM Rev.*, vol. 50, no. 1, Feb. 2008, pp. 37–66.

[9] A. Giridhar and P. R. Kumar, "Distributed Clock Synchronization over Wireless Networks: Algorithms and Analysis," *Proc. 45th IEEE Conf. Decision & Control*, San Diego, CA, Dec. 2006, pp. 4915–20.

[10] P. Barooah and J. P. Hespanha, "Error Scaling Laws for Optimal Estimation from Relative Measurements," *IEEE Trans. Info. Theory*, vol. 55, no. 12, Dec. 2009, pp. 5661–73.

[11] A. Giridhar and P. R. Kumar, "Efficient and Robust Communication Topologies for Distributed Decision Making in Networked Systems," *Proc. 47th IEEE Conf. Decision & Control*, Shanghai, China, Dec. 2009, pp. 3751–56.

[12] A. Bigdeli, A. Tizghadam, and A Leon-Garcia, "Comparison of Network Criticality, Algebraic Connectivity, and other Graph Metrics," *1st Wksp. Simplifying Complex Net.*, Venice, Italy, July 2009.

[13] A. Tizghadam and A. Leon-Garcia, "Autonomic Traffic Engineering for Network Robustness," *IEEE JSAC*, vol. 28, no. 1, Jan. 2010, pp. 39–50.

[14] G. Leduc *et al.*, "An Open Source Traffic Engineering Toolbox," *Comp. Commun.*, vol. 29, no. 5, Mar. 2006, pp. 593–610.

[15] A. Jadbabaie, "On Geographic Routing without Location Information," *Proc. 43rd IEEE Conf. Decision & Control*, Atlantis, Bahamas, vol. 5, Dec. 2004, pp. 4764–69.

[16] A. Tizghadam and A. Leon-Garcia, "On Random-Walks in Direction-Aware Network Problems," *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 38, no. 2, Sept. 2010, pp. 9–11.

Biographies

ALI TIZGHADAM [M] (ali.tizghadam@utoronto.ca) is currently a postdoctoral research fellow at the University of Toronto. He is managing the Network Architecture Laborabory (NAL) in the Communications group. He is also a seasonal instructor at the University of Ontario Institute of Technology (UOIT). He received his M.A.Sc. and Ph.D. in electrical and computer engineering from the University of Tehran (1994) and University of Toronto (2009), respectively. Between his M.A.Sc. and Ph.D. studies he worked in the industry for about 10 years, where he gained an abundance of experience in telecommunications, especially in local exchange switches and access networks. His major research interests are in applications of network science in communications, green communications, autonomic network control and management, network optimization methods, and smart grid.

ALBERTO LEON-GARCIA [F] (alberto.leongarcia@utoronto.ca) received the 2006 Thomas Eadie Medal in Communications from the Royal Society of Canada for his work on the design of multigeneration networks. He holds the Jeffrey Skoll Chair in Computer Networks and Innovation at the University of Toronto. He was founder and CTO of AcceLight Networks in Ottawa, which developed leading edge optical switching equipment. He holds several patents and has published research extensively in the areas of switch architecture and traffic management. He is recognized as an innovator in networking education. In 1986 he led the development of the University of Toronto-Northern Telecom Network Engineering Program. He also led the development of the Master of Engineering in Telecommunications program, and the communications and networking options in the undergraduate computer engineering program. He is the author of the leading textbooks *Probability and Random Processes for Electrical Engineering* and *Communication Networks: Fundamental Concepts and Key Architecture.*