On Robust Traffic Engineering in Core Networks

Ali Tizghadam
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: ali.tizghadam@utoronto.ca

Alberto Leon-Garcia
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: alberto.leongarcia@utoronto.ca

***Abstract*—This paper reports on a probabilistic method for traffic engineering (specifically routing and resource allocation) in backbone networks, where the transport is the main service and robustness to the unexpected changes in network parameters is required. We analyze the network using the probabilistic betweenness of the network nodes (or links). The theoretical results lead to the definition of ”criticality” for nodes and links. Link criticality is used as the main metric to model the risk of taking a specific path from a source to a destination node. Different paths will be ranked based on their criticality measure, and the best path will be selected to route the flow along the core network. The choice of the path is in the direction of preserving the robustness of the network to the unforeseen changes in topology and traffic demands. The proposed method is useful in situations like MPLS and Ethernet networks where path assignment is required.**

***Index Terms*—Robustness, Graph-Theory, Betweenness, Congestion, Traffic Engineering.**

I. Introduction

Robustness is a well studied subject in network literature. Intuitively robustness is the resilience against possible changes in different parameters of the network due to uncertainties. In order to establish the discussion, we give our definition of robustness. There are three major types of changes that may affect the performance of the network:

1. Network topology and connectivity. This includes changes in capacity of the links.
2. Community of interest.
3. Traffic Matrix.

Throughout this paper, we call a ”routing algorithm” or a ”resource allocation method” robust if it can resist against uncertainties which are the result of changes in topology, traffic or community of interest. Our stress in this paper is in traffic surges that might guide network to the congestion mode and our main resource in this study is the link bandwidth.

In contrast to a previous work [1], we used a probabilistic approach to design a robust routing plan for the core network that allows the network service provider to manage the assignment of flows to the paths primarily at the edge of the core network. Using this approach we were able to investigate the problem analytically using metrics from graph-theory and discover some useful aspects of the robustness problem in networks.

The rest of this paper is organized as follows. Section II reviews previous works on network design and robustness problem in networks. In section III the analytical results are discussed. The proposed robust routing plan is introduced in section IV, followed by the simulation results and validation for the case of MPLS networks in section V. The paper is concluded in section VI.

II. Previous work

A wealth of literature is available for network robustness and different aspects of it. In [2] some facts from graph-theory are reviewed which are important for the development of robust network topologies.

Robustness to the traffic uncertainty is investigated from different standpoints. In [1] a framework for robust routing in core network is proposed based on the idea of ”link criticality” and ”path criticality”. Originally, the shortest path betweenness centrality [3], a metric from graph theory, is used to measure the criticality of a node. Suppose that we are measuring the centrality of node k. The betweenness centrality is defined as the share of times a node i needs a node k in order to reach a node j via the shortest path. Link betweenness can also be defined in the same way. A modified version of the link betweenness is used in [1] to define link criticality. Suppose n_{ij} is the number of paths between source-destination pair (s,d) and n_{ikj} is the number of paths between i, j containing the specific link k . Then betweenness of node k for i (source) and j (destination) is given by $\frac{n_{ikj}}{n_{ij}}$. The overall betweenness of a link is defined as the sum of all betweennesses for link k when i, j are changing. This gives an indication of how critical the link is in the network topology. Based on this interpretation of the betweenness centrality, the Path-Criticality Routing (PCR) is proposed in [1] to find the least biased paths (paths with minimum sensitivity to the changes in traffic demands or topology of the network) to run the flow. In [2] a number of graph properties are listed which are useful for determining the robustness of a network against changes in network topology. There is also an abundance of literature for oblivious routing ([4], [5]).

The proposed work in this paper is along the line of our previous research on quantifying the robustness by link and path criticality [1]. Here, we introduce a probabilistic approach to the definition of criticality for links and nodes, and provide the details of our analytical results that lead us to the idea of Random-Walk based Path Criticality Routing (RW-PCR).

III. Analytical Results on Robustness

In this section we provide the details of theoretical results that motivate the approach in this paper. We start by introducing the notions and metrics. In this paper, we consider the probabilistic definition of the node (link) betweenness as the main metric to quantify the criticality. In [6] a probabilistic interpretation of the betweenness is defined based on random walks. The betweenness of a node (link) k for source-destination pair (s,d) is the expected number of times that a random walk passes node k in its journey from source s to destination d. the total betweenness of node k is the sum of this quantity over all possible (s,d) pairs. Now, we define the node criticality for a weighted network simply as the random-walk betweenness of that node over the weight of the node.

$$\eta_k = \frac{b_k}{W_k} \tag{1}$$
$$W_k = \sum_j w_{kj}$$

where $\eta_k,\ b_k,\ W_k$ are the criticality, betweenness, and weight of the node k (or weighted degree of the node) respectively. W_k is equal to the sum of all link weights incident to node k (weight of link (j,k) is shown by w_{jk}). Similarly, the criticality of a link (i,j) (η_{ij}) is defined as the betweenness of the link over its weight:

$$\eta_{ij} = \frac{b_{ij}}{w_{ij}} \tag{2}$$

The main goal to introduce criticality of the nodes (and links in a similar way) is to be able to sort different networks based on their robustness to the changes in traffic demand, topology, and community of interest (source-destination pairs).

To this end, we consider a network which is specified by a graph G=(N,L). Based on our definition of the betweenness, we can quantify the criticality as follows. Each node has a certain probability to send its data to the adjacent nodes. Let's assume a random walk at node s wants to go to node d as its final destination. The destination node is an absorbing state for this random walk and the walk stops as soon as it reaches the destination. Starting from node s the probability of passing node k in next step is given by p_{skd}.

$$p_{skd} = \{^{0 \quad if\ s=d}_{\frac{w_{sk}}{\sum_{q\in A(s)} w_{sq}} \quad otherwise} \tag{3}$$

where $A(s)$ is the set of adjacent nodes of s and w_{sk} is the weight of link (s, d), if any. The first condition in equation 3 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 $p_{dkd} = 0$. Clearly, equation 3 defines a Markovian system. The betweenness of node (link) k for the source-destination pair (s, d) is denoted by b_{skd} and defined as the expected number of times that a random walk from s to d traverses k.

Note that the path from i to k could be of length 0 to infinity. If we specify the probability values p_{skd} for destination d with matrix P_d, then for all $k \neq d$, the probability of entering node k at q^{th} step for different values of s and k can be obtained from corresponding entries of the matrix ${P_d}^q$ and in case of $k = d$ it would be 0. In our calculations, we treat the destination d as a fixed point and write all matrices based on this assumption. At the end we obtain the general results for our metrics by adding up the results for different destinations. In other words, the matrix P_d can be viewed as routing matrix to destination d when the random walk starts from node s. One can write this relationship in matrix form as follows:

$$B_d = [b_{skd}] = \{^{\sum_{q=0}^{\infty} P_d^q \quad if\ k\neq d}_{0 \quad otherwise}$$

B_d is the betweenness matrix for destination d. Using well-known matrix relation $\sum_{q=0}^{\infty} P_d^q = (I - P_d)^{-1}$, this equation can be simplified as:

$$B_d = [b_{skd}] = \{^{(I-P_d)^{-1} \quad if\ k\neq d}_{0 \quad otherwise} \tag{4}$$

By examining equation 4 one can easily see that the removal of column and row d from betweenness and probability matrices does not affect the other entries. We use $M(i|j)$ to denote the reduced matrix resulted from removing i^{th} row and j^{th} column of matrix M. The equation 4 can be written as:

$$B_d(d|d) = (I - P_d(d|d))^{-1} \tag{5}$$

Let $W = [w_{ij}]$ be the weight matrix of the graph, D be the diagonal matrix of weighted degrees or graph nodes, and L be the Laplacian of the graph. We know that:

$$\begin{aligned} L &= D - W \\ D &= diag(W_1, W_2, ..., W_n), \quad W_i = \sum_{k=1}^{n} w_{ik} \\ P_d(d|d) &= D^{-1}(d|d) \times W(d|d) \end{aligned}$$

The last equation is the direct result of equation 3. Now we have:

$$\begin{aligned} I - P_d(d|d) &= I - D^{-1}(d|d) \times W(d|d) \\ &= D^{-1}(d|d) \times (D(d|d) - W(d|d)) \\ \Rightarrow I - P(d|d) &= D^{-1}(d|d) \times L(d|d) \end{aligned} \tag{6}$$

Replacing equation 6 in 5 results in:

$$B_d(d|d) = L^{-1}(d|d) \times D(d|d) \tag{7}$$

Note that the graph $G(N, L)$ is assumed to be connected which means that the rank of graph Laplacian L is $(n-1)$. As a result, the inverse of reduced Laplacian $L(d|d)$ exists and equation 7 has a unique solution.

Now we need to write the equation 7 in terms of the Laplacian of the original graph. Without loss of generality, we rename the nodes so that the removed node becomes the last node of the graph (node n). Now, in order to write $L^{-1}(n|n)$ in terms of L, we use the Moore-Penrose generalized inverse matrix of L ([7]). The Moore-Penrose inverse of $L(n|n)$ and the $L^{-1}(n|n)$ are equal since $L(n|n)$ is an $(n-1) \times (n-1)$ matrix with rank $n-1$. In other words, $L(n|n)$ is full-rank and its inverse is the same as its Moore-Penrose inverse. To obtain L from $L(n|n)$, we first add a column to $L(n|n)$ to

get: $Q = [L(n|n)\ z_{n-1}]$.
The column-vector z_{n-1} has to be chosen in a way to make the sum of every row of the matrix Q equal to zero. We use the following formula from [7] which is a recursive formula to obtain the Moore-Penrose inverse of a matrix when a column is added to the original matrix. Let $A \in \mathbb{F}^{p\times q}$ be a $p\times q$ matrix and $b \in \mathbb{F}^{p}$ be a $p \times 1$ column vector.

$$\begin{pmatrix} A & b_p \end{pmatrix}^+ = \begin{pmatrix} A^+(I - b_p\zeta_p) \\ \zeta_p \end{pmatrix} \tag{8}$$

where ζ_p is a $1 \times p$ row vector such that

$$\zeta_p = \{ \begin{smallmatrix} (b_p - AA^+b_p)^+ & if\ b_p \neq AA^+b_p \\ \frac{b_p^*(AA^*)^+}{1+b_p^*(AA^*)^+b_p} & b_p = AA^+b_p \end{smallmatrix}. \tag{9}$$

where $*$ means *conjugate transpose*.
To satisfy the requirement of Laplacian matrix we need to have

$$[L(n|n)\ z_{n-1}]\overrightarrow{1}_{n-1} = 0 \tag{10}$$

where $\overrightarrow{1}_{n-1}$ is a $(n-1)\times 1$ vector of all ones: $\overrightarrow{1}_{n-1} = [1\ 1\ 1\ ...\ 1]^t$.
From 10 one can easily see that:

$$\begin{aligned} L(n|n)\overrightarrow{1}_{n-1} + z_{n-1} = 0 \\ z_{n-1} = -L(n|n)\overrightarrow{1}_{n-1} \end{aligned} \tag{11}$$

Now from 8 by replacing $A = L(n|n)$ and using 11, one can see:

$$\begin{aligned} Q^+ &= \begin{pmatrix} L(n|n) & z_{n-1} \end{pmatrix}^+ \\ &= \begin{pmatrix} L^+(n|n) - L^+(n|n)z_{n-1}\zeta_{n-1} \\ \zeta_{n-1} \end{pmatrix} \\ &= \begin{pmatrix} L^+(n|n) + L^+(n|n)L(n|n)\overrightarrow{1}_{n-1}\zeta_{n-1} \\ \zeta_{n-1} \end{pmatrix} \\ &= \begin{pmatrix} L(n|n)^+ + \overrightarrow{1}_{n-1}\zeta_{n-1} \\ \zeta_{n-1} \end{pmatrix} \\ &= \begin{pmatrix} L(n|n)^+ \\ 0 \end{pmatrix} + \overrightarrow{1}_{n-1}\zeta_{n-1} \end{aligned}$$

This expression for Q^+ can be expanded as:

$$Q^+ = \begin{pmatrix} L(n|n) & z_{n-1} \end{pmatrix}^+ = \begin{pmatrix} L^+(n|n) \\ 0 \end{pmatrix} + \begin{pmatrix} \zeta_{n-1} \\ \zeta_{n-1} \\ . \\ . \\ . \\ \zeta_{n-1} \end{pmatrix} \tag{12}$$

Equation 12 asserts that:

$$\begin{aligned} q^+_{sk} &= (L^+(n|n))_{sk} + (\zeta_{n-1})_k \quad if \quad s \neq n \\ q^+_{nk} &= 0 + (\zeta_{n-1})_k \quad if \quad s = n \end{aligned}$$

Subtracting these two equations show that:

$$\Rightarrow (L^+(n|n))_{sk} = q^+_{sk} - q^+_{nk} \tag{13}$$

With the same approach , we add the n^{th} row to Q to obtain the $n \times n$ Laplacian matrix L: L $= \begin{bmatrix} Q \\ d \end{bmatrix}$ With similar reasoning and using equation 8 for the transpose of this matrix (which is equal to matrix L as the Laplacian is symmetric), one can obtain:

$$\Rightarrow q^+_{sk} = l^+_{sk} - l^+_{sn} \tag{14}$$

Using equations 13, 14 we can find our desired result.

$$(L^+(n|n))_{sk} = l^+_{sk} - l^+_{sn} - l^+_{nk} + l^+_{nn} \tag{15}$$

but $L^+(n|n) = L^{-1}(n|n)$, so:

$$(L^{-1}(d|d))_{sk} = l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}$$

In this equation we replaced node n with d to make it more general. Now, according to the equation 7, we can obtain the betweenness of the node k for source-destination pair (s, d):

$$\begin{aligned} B_d(d|d) &= L^{-1}(d|d) \times D(d|d) \\ (B_d(d|d))_{sk} &= (l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}) \times W_k \\ \frac{b_{skd}}{W_k} &= l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd} \end{aligned}$$

To obtain the total betweenness of node k, we need to consider the effect of all source-destination pairs.

$$\frac{b_k}{W_k} = \frac{1}{W_k}\sum_s\sum_d b_{skd} = \frac{1}{W_k}\sum_s\sum_d \frac{b_{skd} + b_{dks}}{2}$$

$$\frac{b_k}{W_k} = \sum_s\sum_d \frac{l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd} + l^+_{dk} - l^+_{ds} - l^+_{sk} + l^+_{ss}}{2}$$

$$\frac{b_k}{W_k} = \sum_s\sum_d \frac{l^+_{dd} - l^+_{sd} - l^+_{ds} + l^+_{ss}}{2}$$

$$\begin{aligned} \frac{b_k}{W_k} &= \frac{1}{2}\sum_s\sum_d(l^+_{ss} + l^+_{dd} - 2l^+_{sd}) = \frac{1}{2}\tau \\ \tau &= \sum_s\sum_d(l^+_{ss} + l^+_{dd} - 2l^+_{sd}) \end{aligned} \tag{16}$$

To obtain equation 16 we used the fact that Laplacian matrix (and its Moore-Penrose inverse) is symmetric.
A similar result can be derived for a link of the graph as well. For a link (i,j), one can write the betweenness of the link based on the betweenness of its two end nodes:

$$\begin{aligned} b_{ij} &= \frac{1}{2}(\frac{w_{ij}}{\sum_k wik}b_i + \frac{w_{ji}}{\sum_k wkj}b_j) \\ &= \frac{1}{2}(\frac{w_{ij}}{W_i}\tau W_i + \frac{w_{ji}}{W_j}\tau W_j) \\ &= \frac{1}{2}(w_{ij} + w_{ji})\tau = w_{ij}\tau \end{aligned}$$

or equivalently

$$\eta_{ij} = \frac{b_{ij}}{w_{ij}} = \tau \tag{17}$$

Equations 16, 17 give exact formulas to measure the generalized criticality of a node or a link of a graph. If we consider the capacity of a link as its weight, then the left hand side of equation 16 is the same as equation 1.

Observation 3.1: From equations 16 and 17 one can see that the node/link criticality is independent of the choice of node/link.

Corollary 3.2: Let T be the average time that a random-walk is in the system for all source-destination pairs, and B be the average node betweenness of all nodes. Then:

$$B = (n-1)T \tag{18}$$

Proof: The average time that a random-walk starting at node s is in the system before reaches to its destination node d is equal to

$$\begin{aligned} T_{sd} &= \sum_{s,d} b_{skd} \\ &= \sum_{k=1}^{n} (l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}) W_k \end{aligned}$$

Now, the average time in system considering all possible source-destination pairs would be

$$\begin{aligned} T &= \frac{1}{n(n-1)} \sum_{s,d} T_{sd} \\ &= \frac{1}{n(n-1)} \sum_{s,d} \sum_{k=1}^{n} (l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}) W_k \\ &= \frac{1}{n(n-1)} \sum_{k=1}^{n} (W_k \sum_{s,d} (l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd})) \\ &= \frac{1}{n(n-1)} \sum_{k=1}^{n} (W_k \times \frac{b_k}{W_k}) \\ &= \frac{1}{n(n-1)} \sum_{k=1}^{n} b_k = \frac{B}{n-1} \end{aligned}$$

This is exactly what we want in equation 18.

Corollary 3.3: The normalized betweenness of each node i of the graph is $\frac{b_i}{\sum_{k=1}^{n} b_k}$. It can be shown that this quantity is equal to the stationary probability of that node in a Markov chain built on the weights of the graph.

Proof: Equation 17 can be used to simplify the normalized betweenness of a node.

$$\begin{aligned} Normalized\ b_i &= \frac{b_i}{\sum_{k=1}^{n} b_k} \\ &= \frac{\frac{1}{2}\tau W_i}{\frac{1}{2}\sum_{k=1}^{n} \tau W_k} \\ &= \frac{W_k}{W} = \pi_i \end{aligned} \tag{19}$$

In these equations: $W_k = \sum_{j=1}^{n} w_{ij}$, $W = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}$.

Now we are ready to state the main result of this research. Let λ be the average input rate at any individual node 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_{max} be the average load on the node which has the maximum betweenness among all the nodes, and consider the capacity of this node as c^* . x_{max} can be approximated by the total average rate of this node times the average time that a demand is in the system.

$$\begin{aligned} x_{max} &= n\lambda\pi_{max} T \\ &= n\lambda \frac{b_{max}}{\sum_i b_i} T \\ &= n\lambda \frac{b_{max}}{nB} \frac{B}{n-1} \\ x_{max} &= \frac{\lambda}{n-1} b_{max} \end{aligned} \tag{20}$$

To obtain 20 we used corollary 3.2 and 3.3. Hence:

$$\begin{aligned} x_{max} &<= c^* \\ \frac{\lambda}{n-1} b_{max} &<= c^* \\ \lambda &\leq \frac{n-1}{\frac{b_{max}}{c^*}} \\ \lambda &\leq \frac{n-1}{\max_{n \in N} \eta} \qquad (21) \\ \lambda &\leq \frac{n-1}{\eta} \qquad (22) \end{aligned}$$

we used observation 3.1 to get the equation 22 . This result can be summarized in the following theorem.

Theorem 3.4: To maximize the acceptable input load of a network, one needs to minimize the node/link criticality of the network.

This result has many consequences. In this paper we are looking for possible ways to engineer the traffic of the transport network. Hence, we use theorem 3.4 to design a heuristic method for routing and flow assignment problem. This is the subject of the next section.

IV. Design of a robust routing scheme

An application of the analytical results extracted in previous section is the design of a robust routing scheme which is required for every core network in order to be able to cope with unpredicted changes in traffic and topology. In this section we briefly outline an approach to the design of a robust routing algorithm.

A. Random Walk Path Criticality Routing (RW-PCR)

In this part our goal is to find a robust routing plan for the core network that allows the network service provider to manage the assignment of flows to the paths primarily at the edge of the core network. To achieve the goal first we need to identify the important factors affecting the routing plan and flow assignment. We have already mentioned that these factors are topology, community of interest (source-destination pairs), capacity of the links, and traffic matrix.

In order to have a robust routing plan we need to recognize the effect of link and node failure. References [8], [3], [6] provide us with useful metrics to measure the sensitivity of the network to node or link failures. Capacity of a network is another key issue in flow assignment problem. Clearly the paths with more capacity are desired since the low capacity paths are prone to congestion. Hence an intelligent routing plan should avoid routing the flows onto the low capacity paths and should request for capacity increases for those paths if possible. Finally traffic demand directly affects the routing plan. The traffic demand profile may change from time to time (e.g. week-day traffic profile). Traffic changes might be predictable and periodical or chaotic. We need to find a routing scheme which is robust to the predicted traffic patterns and unpredicted ones to the extent possible.

B. Link Criticality Index

We need some metrics to estimate the effect of the aforementioned characteristics. Link Criticality is already defined in equation 2. we chose to have the link betweenness over "available capacity" as our main metric, and called it Link Criticality Index (LCI). Indeed the LCI is the same as criticality in 2 when the link weight is chosen to be the available bandwidth of the link. LCI captures the effects that we would like to quantify. One can easily see that betweenness centrality captures the effect of load. The higher the link betweenness, the more the chance of congestion. On the other hand, the available capacity has inverse effect on the congestion. In other words, the more the available capacity, the less the chance of congestion. In order to capture the effect of source-destination pairs (community of interest) we use the equation 14 to obtain the betweenness, but the summation is only on the active source-destination pairs. This approximation makes our approach away from optimal solution, but still gives a near-optimal estimation of the paths.

The link criticality index (LCI) could be obtained by having the betweenness and available capacity of the link.

$$I(x) = 0 \quad if \quad x > 0 \quad otherwise \quad 0$$
$$LCI(i,j) = \frac{b_{ij}}{c_r(i,j)} \times \frac{1}{I(c_r(i,j) - \gamma_{ij})}$$

In above equations $I(x)$ is the indicator function, $LCI(i,j$ is the total criticality of the link (i,j), $c_r(i,j)$ is the available capacity of link (i,j), and $\gamma(i,j))$ is the present demand on link (i,j). The indicator function is added in the denominator to guarantee that if the demand is more than the available capacity of the link, the demand is not accepted in this link (the link criticality would be effectively infinite in this situation). Definition of the link criticality is clearly showing that the criticality of a link is increasing if more load is carried through this link.

When the LCI of all the links are known, the criticality of a path, Path Criticality Index (PCI), will be the defined as the maximum of the LCI of the links belonging to the path.

$$PCI(p) = max(LCI(s,i_1), LCI(i_1,i_2),\ ...,\ LCI(i_q,d))$$

where p is a nominal path from node s to node d and q is the order or the number of links of the path. Note that the use of maximum in calculation of PCI is motivated by equation 21.

C. RW-PCR Algorithm

The basic idea of our routing algorithm is to accommodate new requests for connections along the paths with low PCI. The basic idea of the PCR algorithm is as follows. We label each and every link of the graph with its LCI as the cost (note that this cost is different than the weight set w_{ij}) and use Dijkstra's algorithm to obtain the shortest path(s) from a source s to a destination d using the assigned cost for the links. When a demand for source-destination pair $s-d$ arrives, the shortest path obtained in this way would be considered as a candidate to be assigned to the demand. A simple call-admission control is applied here by considering a threshold tr for the criticality of the path. If the PCI is more than this threshold, then the flow would be considered too risky for the network and be rejected (blocked), otherwise the path is used as the route and the demand flow is assigned to this path. The available bandwidth of all the links on this path is updated and the LCI's are also modified accordingly. These steps are shown in flowchart of Fig. 1.

To approximate the time complexity of the algorithm, we note that the running time to get the Moore-Penrose inverse is $O(mn^{\frac{1}{2}})$ [7], where m and n are the number of links and nodes in the graph respectively. The main part of the RW-PCR can be obtained in $O(nlog(n))$ as it is just a shortest path algorithm with link costs. Hence the complexity of the algorithm would be $O(mn^{\frac{3}{2}}log(n))$.

Fig. 1. Flowchart of RW-PCR Algorithm

V. Evaluation

In order to investigate the effectiveness of our RW-PCR algorithm, we ran a set of simulations on the network of Fig. 2. The numbers on some of the links show the bandwidth of the link, and the bandwidth is assumed to be 100 units for all the links whose bandwidth are not shown in Fig. 2. We apply RW-PCR to create LSPs (Label Switch Path) assuming that MPLS is used in the network to create the paths. In the first experiment the requests for LSP arrive with Poisson distribution and stay for ever (no departures). In our tests the bandwidth requests for paths (LSPs) are taken to be

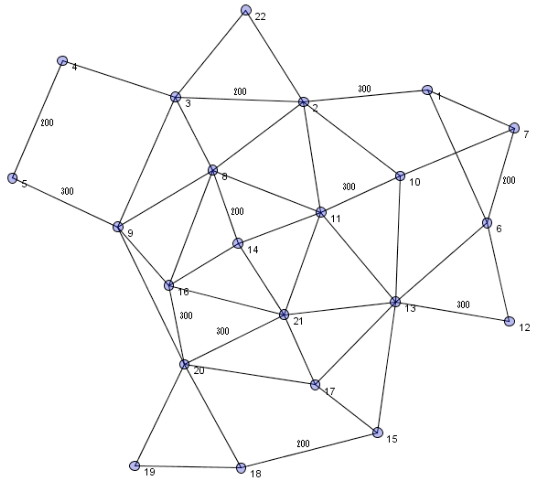

Fig. 2. The Test Network

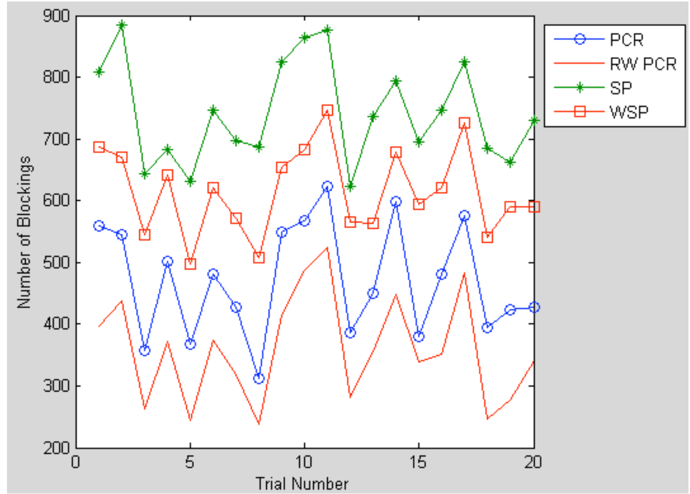

Fig. 4. Dynamic Case: Result of Applying PCR, Random-Walk PCR, SP, and WSP to the Network under Test

uniformly distributed between 1 to 3 units. In Fig. 3 we show the number of rejected calls for this case and compare the performance to that of original PCR, shortest path (SP), and widest shortest path (WSP). The test is performed 20 times and each time with 2000 path requests. We measured the number of blocked requests. In another experiment we examined the

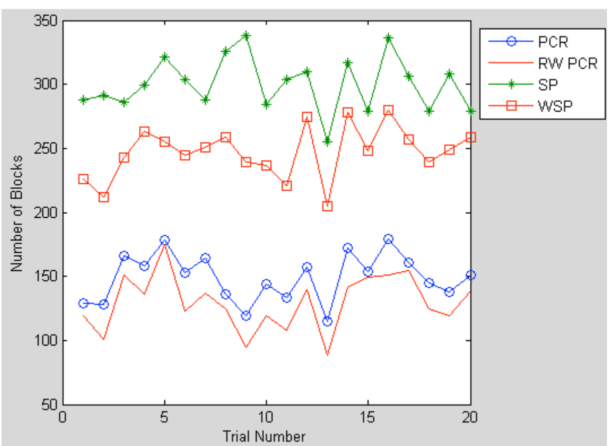

Fig. 3. Static Case: Result of Applying PCR, Random-Walk PCR, SP, and WSP to the Network under Test

behavior of the algorithms in the presence of dynamic traffic. Fig. 4 shows the number of the path requests rejected in 20 experiments for the following scenario. Path requests arrive between each source-destination point (which is chosen at random) according to a Poisson process with an average rate λ, and the holding times are exponentially distributed with mean μ. We set $\frac{\lambda}{\mu} = 1800$ in our experiments. We generate 7000 requests and measure the rejections or blocking for each one of the algorithms. The results are illustrated in Fig. 4.
In both static and dynamic cases, one can easily see that the RW-PCR has the best performance, the original PCR is in second place and WSP and SP in next positions.

VI. Conclusion

In this paper we analyzed the robustness of a network to the unexpected changes in different parameters and proposed an approach for path setup and routing of flows in transport networks.
The essence of our work is based on determining a criticality index for each link/path showing how critical that link/path is to the changes in the topology and traffic demand of a network. We gave an analytical expression for the link and node criticality, and then proposed a heuristic for flow assignment based on it. Our algorithm identifies the least critical paths for allocation of new traffic flow requests. The results from applying the proposed algorithm to networks that are difficult to handle by existing approaches are very encouraging.
There are many issues that remain to be investigated in the new approach. We need to investigate more on the effect of the threshold parameters. As another research challenge we need to look into the back up paths and the efficient algorithms to find them again with the goal of having less critical paths and back up paths.

References

[1] A. Tizghadam and A. Leon-Garcia. A robust routing plan to optimize throughput in core networks. *ITC20, Elsvier*, pages 117–128, 2007.
[2] A. H. Dekker and B. D. Colbert. Network robustness and graph topology. *Australasian Computer Science Conference*, 26:359–368, Jan. 2004.
[3] L. C. Freeman. Centrality in networks: I. conceptual clarification. *Social Networks*, (1):215–239, 1978/79.
[4] M. S. Kodialam and et al. Maximum throughput routing of traffic in the hose model. *INFOCOM*, pages 1217–1225, 2006.
[5] N. G. Duffield and et al. A flexible model for resource management in virtual private network. *ACM SIGCOMM*, Aug. 1999.
[6] M. E. J. Newman. A measure of betweenness centrality based on random walks. *arXiv cond-mat/0309045.*, 2003.
[7] D. S. Bernstein. *Matrix Mathematics*. Pronston University Press, 2005.
[8] L. S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.