APPLICATION TO DETERMINE OPTIMIZED PATH FOR NETWORK ROBUSTNESS

Kartikay Kaushik

Department of Electronics Engineering,
Indian Institute of Technology (Indian School of Mines), Dhanbad, India.

ABSTRACT

The recent trends of increasing unpredictability of traffic demand and the proliferation of networked devices have led to a demand for a traffic engineering application which views network robustness as a crucial factor. Robustness refers to the resilience of infrastructure networks against random and targeted failures, caused by traffic shifts, natural disasters and Denial of Service (DoS) attacks. In this paper, the author developed an application to assign the links of the network a criticality value and finds the least critical path or in other terms, the most robust path. The dynamics of the network are translated to graph metrics and the application optimizes these metrics to determine the most robust path in the network. The developed application can be further extended by incorporating its output as a feedback mechanism for another application developed for automation of the network system for robustness. The application was written in Python 2 and implemented on Ubuntu 14.04.4.

KEYWORDS

Networks, Performance prediction, Robustness, Criticality, Betweenness

1. INTRODUCTION

The study of network robustness has been an intriguing concept for scientists in the last few years. The diverse application of robustness in the fields of engineering, ecology, economics, medicine, and biology has been a motivation for scientific research across the world. With the explosion in demand for networked devices, the changes in network parameters have become highly unpredictable and thus it is important to accommodate these uncertainties while developing an application [1]. An important concept associated with a network is its maintenance. A large fraction of the cost associated with a network is spent in maintaining the network [2] [3]. Hence a relatively stable system is desirable to reduce the costs involved with its maintenance. The paper aims to minimise the probability of a network failure and the costs associated with it, by finding the most stable path between two given nodes in a network.

The concept of network robustness plays an important role in improving the grade of service provided to a customer by a network service provider [4]. A low probability of network failure leads to maximal system reliability[5]. The application, when supplied with the data of flow of traffic between source and destination pairs, provides the path as an output that leads to maximum stability against external factors. The path obtained can then be used by an SDN controller to route the traffic. Hence enabling the network to withstand the changes in the parameters and adapting by changing the flow of traffic between the nodes. The output of this application

David C. Wyld et al. (Eds) : IPPR, SOENG, DaMi, CSIT, AIS, CSE, CSIP, CCNET - 2018
pp. 159–167, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80512

(primary) can serve as a feedback to another application (secondary) that is developed for network automation. The secondary application can be developed to monitor and acquire data from the network periodically and the primary application then processes the data to give the most robust path. The secondary application can later enforce the obtained path in the network thus enabling a sense of automation.

2. BACKGROUND AND RELATED WORK

There has always been a trade-off between systems that achieve high performance and systems that are robust [6]. The former aims to deliver the best performance at the cost of high sensitivity and the latter aims to provide better stability at the cost of performance. But in the recent trends, it has been observed that network and telecommunication industries opt for more robust networks [7] [8]. It is necessary to understand the Percolation Theory and the impact of node failures on the integrity of the network [9]. In this paper, the primary approach is to define the metrics that affect the network robustness.

In this paper, the concept of using graph-theoretic metrics is inspired by Dekker and Colbert [10] who used a similar approach to explain robustness. The paper [10] defines "node connectivity" as a metric to measure robustness. Their approach involved understanding the behaviour of the network in case of node failures. The papers [10] [11] concluded that node similarity and optimal connectivity are the conditions required for a network to be robust and provided methods to evaluate the same.

Freeman [12] explains the concept of "Betweenness Centrality" for node and link. The paper explains Betweenness Centrality, for node k for flow from source node i to destination node j, as the ratio of shortest paths from node i to j that pass through k. The overall Betweenness Centrality of the node k is sum of the centralities over all source-destination pairs. Link Betweenness is defined similarly.

Tezghadam and Leon-Garcian [13] define the concepts of "link criticality" and "path criticality". The paper defined Link Criticality describes the impact of the failure of a particular link on the entire network and Path Criticality as the desirability of a path based on the criticality of the links. Also, it argues that in traffic management shortest paths need not be the best paths in all circumstances. Hence, the paper redefined the concept of Betweenness. Itstated that ifn_{ij} be the number of feasible paths between i and j and if n_{ikj} be the number of paths between i and j containing the link k. Betweenness for node k for source i and destination j is then n_{ikj}/n_{ij}. The overall betweenness of link k is the sum of the betweennesses for link k over all i and j. Hence, as Javier Mart´ınHern´andez∗ and Piet VanMieghem† stated in the paper [14], Betweenness Centrality is used as a metric to decide how critical a link in the network topology is.

3. ALGORITHM DESIGN

Tizghadam and Leon-Garcia [15] provided a theoretical basis for the design of the algorithm to calculate the metric, network criticality, to determine the robustness of the network. In this paper, the application was developed using this as a theoretical base [15].

3.1. Weight Assignment

In [15], the authors showed that if the weight increases, the goodness of that link increases. The factors increasing the goodness of the link such as available bandwidth are called as "beneficial QoS parameters" and those that decrease the goodness of the link are called the "detrimental QoS parameters" such as packet loss or length of the link. Each of these parameters is mapped to weights with an appropriate method as in [16]. In this paper, the weights are mapped by taking

the product of weights of beneficial QoS parameters and dividing it with the product of detrimental QoS parameters. Thus, the weights of each link are obtained.

3.2. Network Criticality

Newman [17] argued that a probabilistic interpretation of the betweenness is defined based on random walks in a graph. A random-walk starts from a source node i, chooses a neighbor at random with equal probabilities, and gets there using the link between the source and the neighbor. The random walk continues until it reaches a specified destination *d*, where it stops. The Betweenness b_{sk}(d) of a node (link) *k* for source-destination pair *s-d*is the expected number of times 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.

Consider a random walk from a source s to a destination d. The destination node is an absorbing statefor this random walk and the walk is stopped in destination.The probability of passing node *k* in next step is shown by$p_{sk}(d)$and defined as:

$$p_{sk}(d) = \begin{Bmatrix} 0 & if\ s = d \\ \frac{w_{sk}}{\sum_{q\in A(s)} w_{sq}} & otherwise \end{Bmatrix} \quad (1)$$

Where A(s) is the set of adjacent nodes of *s* and $w_{sk}$is the weight of link (*s, k*). The first condition in equation (1) 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$p_{dk}(d) = 0$.

Tizghadam and Leon-Garcia [15] thus defined node criticality for a weighted network simply as the random-walk betweenness of that node over the weight of the node.

$$n_k = \frac{b_k}{W_k} W_k = \sum_{j\in A(k)} w_{kj} \quad (2)$$

where $n_k b_k$, $W_k$are the criticality, betweenness, and weight of 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 (k,j) is shown by w_{kj}).

The authors [15] derived an expression for node betweenness by making a matrix $P_d$that would describe $p_{sk}(d)$ for destination d. The probability of entering k at q^{th}step is P_d^q. The authors treated d as a fixed point and the matrices are written under this assumption. General results are obtained by adding up for all destinations.

$$B_d = [b_{sk}]_d = \begin{Bmatrix} \sum_{q=0}^{\infty} P_d^q & if\ k \neq d \\ 0 & if\ k = d \end{Bmatrix} = \begin{Bmatrix} \left(I - P_d(d|d)\right)^{-1} & if\ k \neq d \\ 0 & if\ k = d \end{Bmatrix} \quad (3)$$

Where $B_d$is the betweenness matrix for destination d. As observed, the row and column d account to 0 always and hence their removal would not affect other entities. M(i|j) denotes Matrix M without row i and column j. Hence,

$$B_d(d|d) = \left(I - P_d(d|d)\right)^{-1} \quad (4)$$

If L is the Laplacian of the Matrix, in [15] the author obtained the equation for criticality by using the following equations

L = D – W where D = diag (W_1,W_2, W_3, W_4… W_n)and W is the weight matrix (5)

$P_d(d|d) = D(d|d)^{-1} \times W(d|d)$

I - $P_d(d|d)$ = I - $D(d|d)^{-1} \times W(d|d)$
I - $P_d(d|d) = D(d|d)^{-1} \times$ L(d|d) (6)

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

The reduced inverse of the Laplacian matrix is given as

$$\text{L(d|d)}^{-1}_{sk} = l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd} \quad (8)$$

Where $l^+_{sk}$is the entry of row s and column k of the Moore-Penrose inverse of L.

$(B_d(d|d))_{sk} = (l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}) \times W_k$

$\frac{[b_{sk}]_d}{W_k} = l^+_{sk} - l^+_{sd} - l^+_{dk} + l^+_{dd}$

For total betweenness of node k, the effect of all source-destination pairs is considered.

$\frac{b_k}{W_k} = \frac{1}{W_k}\sum_s \sum_d [b_{sk}]_d = \frac{1}{W_k}\sum_s \sum_d \frac{[b_{sk}]_d + [b_{dk}]_s}{2}$

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

$$\frac{b_k}{W_k} = \sum_s \sum_d \frac{l^+_{dd} - 2l^+_{sd} + l^+_{ss}}{2} = \frac{1}{2}\sum_s \sum_d T_{sd} = \frac{1}{2}T \quad (9)$$

T = $\sum_s \sum_d T_{sd} = \sum_s \sum_d (l^+_{ss} + l^+_{dd} - 2\, l^+_{sd})$

Hence, $b_k = W_k \frac{T}{2}$ for a node k or $b_{ij} = w_{ij}T$ for a link (i,j) (10)

T is known as the network criticality. Less the network criticality, lesser the sensitivity to changes. Hence more stable.

3.3. Optimization of Network Criticality

According to Tizghadam and Leon-Garcia [15], the equation for an optimal weight set W*, and using the concept of optimisation from [18] [19]

$$C\frac{\partial T}{\partial w_{ij}} + T \geq 0 \; \forall \; (i,j) \in E \quad (11)$$

$$\text{And } \frac{\partial T}{\partial w_{ij}} = -2n \left|\left| L^+_i - L^-_j \right|\right|^2 \quad (12)$$

Substituting (10) and (12) in (11)

$$\frac{\partial b_{ij}}{\partial w_{ij}} = w_{ij}\left[-2n \left|\left| L^+_i - L^-_j \right|\right|^2\right] + T \quad (13)$$

By substituting C = -1 in (11) and substituting (13)

$$\frac{\partial b_{ij}}{\partial w_{ij}} = w_{ij}\left[2n\left|\left|L_i^+ - L_j^-\right|\right|^2\right] \quad (14)$$

Equation (14) gives the value of optimised cost of each link.

3.4. Finding Most Robust Path

The cost of each link is determined from Equation (14). These costs are assigned as the new weights of the links. The least cost path can be found by using several techniques like Dijkstra's Algorithm [20]. This least cost path is also the most robust path [21].

3. APPLICATION DEVELOPMENT

The application was written in Python 2 in Ubuntu 14.04.4. The input of the application is taken in the form of an adjacency matrix wherein the elements of the matrix are the weights of the links. The Laplacian is obtained by Equation (5). The term D is obtained by constructing a diagonal matrix from the matrix obtained by taking the sum along each row. Following the equation (9), a matrix is made for each s-d pair. The sum of the elements of the matrix gives the network criticality.

From, Equation (10), the product of Network Criticality with weight matrix results in a betweenness matrix that gives betweenness for every source and destination pair. Equation (14) results in a cost matrix that gives the cost for every source and destination pair. The application takes source and destination node as an input from the user. By using Dijkstra's Algorithm, the least cost path is obtained between the given source and destination node. Hence, the most robust path is obtained between the given nodes.

4. EVALUATION

In this section, experiments are conducted on different network topologies and run the application for different source and destination nodes. The first line of input is the size of the network or the number of nodes present in the network. From the second line, the adjacency matrix of the graph is given as an input. The application gives the network criticality, betweenness matrix and cost matrix for the given graph. When supplied with the initial point and the output point, the output is the most robust path between the nodes.

4.1. Topology - 1

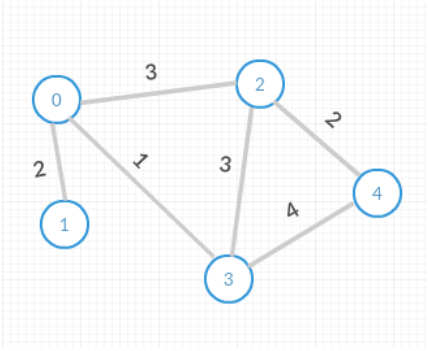

Figure 1. Topology – 1

The output for Figure 1 for initial point as 0 and destination point as 4 is given in Figure 2. The size of the matrix is 5 and the adjacency matrix is obtained from Figure 1.

```
mininet@mininet-vm:~/pox/sdn$ sudo python robnet.py
size: 5
0 2 3 1 0
2 0 0 0 0
3 0 0 3 2
1 0 3 0 4
0 0 2 4 0
('The network criticality is: ', 9.5737704918032769)
The betweenness matrix for the links is:
[[  0.          19.14754098  28.72131148   9.57377049   0.        ]
 [ 19.14754098   0.           0.           0.           0.        ]
 [ 28.72131148   0.           0.          28.72131148  19.14754098]
 [  9.57377049   0.          28.72131148   0.          38.29508197]
 [  0.           0.          19.14754098  38.29508197   0.        ]]
The cost matrix for the links:
[[ 0.          4.          2.02526203  1.15775329  0.        ]
 [ 4.          0.          0.          0.          0.        ]
 [ 2.02526203  0.          0.          0.73206127  0.86535877]
 [ 1.15775329  0.          0.73206127  0.          0.79333512]
 [ 0.          0.          0.86535877  0.79333512  0.        ]]
Initial point: 0
Destination point: 4
[0, 3, 4]
mininet@mininet-vm:~/pox/sdn$ _
```

Figure 2. Output for Topology – 1

Thus, from the above application, it has been observed that the path from 0 to 3 to 4 is the most stable path between 0 and 4.

4.1. Topology - 2

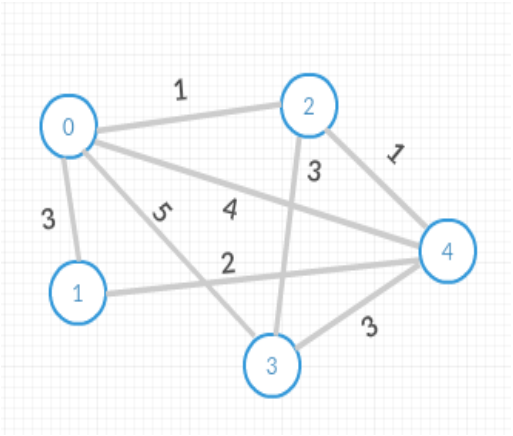

Figure 3. Topology – 2

The output for Figure 3 for initial point as 1 and destination point as 3 is given in Figure 4. The size of the matrix is 5 and the adjacency matrix is obtained from Figure 3.

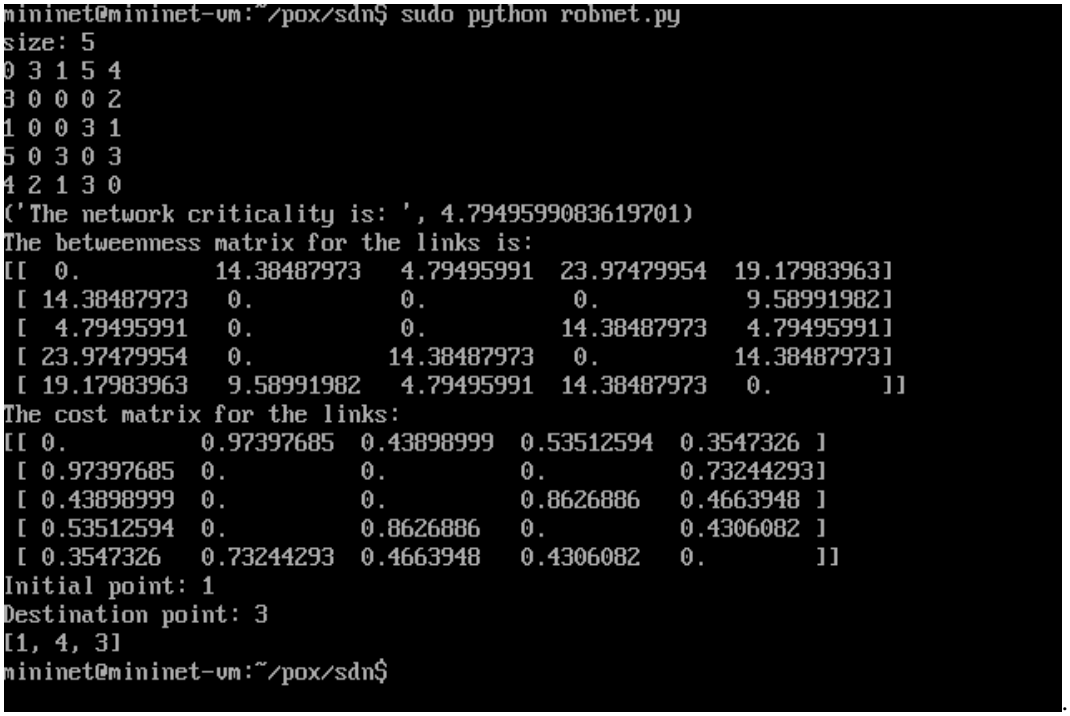

Figure 4. Output for Topology – 2

Thus, from the above application, it has been observed that the path from 1 to 4 to 3 is the most stable path between 1 and 3.

3. CONCLUSION AND FUTURE WORK

In this paper, an application was developed to determine an optimized path for network robustness. The developed application has been tested for different network topologies. It has been observed that the path obtained in the output had the least cost and hence most stable.

The work can be further extended towards automation of networks by dividing an application into three parts. They are data acquisition, data processing and policy enforcement. The data acquisition and policy enforcement stages involve interacting with the physical layer to receive data and implement policies. The application developed in this paper can be used in the data processing stage. The data acquired from the traffic is converted into graph-theoretic metrics. The data is processed and the path obtained from it is implemented. A continuous and periodic monitoring of the traffic and the updating of policies induce a sense of automation in the network. By reducing the human intervention, the quality of service can be highly increased. It can solve the problem of costs involved with the maintenance and the delay in response.

ACKNOWLEDGEMENTS

This research was supported in part by Department of Electronics Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad and Centre for Development of Advanced Computing, Pune.

REFERENCES

[1] David G. Messerschmitt, What the NII could be: A User Perspective, The Unpredictable Certainity

[2] E. Arcaute, R. Johari and S. Mannor, Network Formation: Bilateral Contracting and Myopic Dynamics.

[3] Amir Ranjbar, Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) Foundation Learning Guide, Ch 3.

[4] Gerald R. Ash, Robust Design of Dynamic Routing Networks, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Volume 5, 1991.

[5] Ioannis P. Chochliouros and George A. Heliotis, Optical Access Networks and Advanced Photonics: Technologies and Deployment Strategies, Page 37

[6] D. Applegate and E. Cohen. Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: understanding fundamental tradeoffs. In SIGCOMM, pages 313–324, 2003.

[7] R. Boutaba, W. Szeto, and Y. Iraqi. DORA: Efficient Routing for MPLS Traffic Engineering. Journal of Network and Systems Management, 10(3):309–325, September 2002.

[8] 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.

[9] D. Stauffer and A. Aharony. Introduction to Percolation Theory.Taylor and Francis.London, 1994.

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

[11] Ali Tizghadm, Alireza Bigdeli, Alberto Leon-Gracia and Hassan Naser, Joint Optimization resources and routes for minimum resistance from communication networks and power grids, Springer 2012.

[12] L. C. Freeman. Centrality in Networks: I. Conceptual Clarification. Social Networks, (1):215–39, 1978/79

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

[14] Classification of graph metrics Javier Mart´ın Hern´andez∗ and Piet VanMieghem†, November, 2011.

[15] A. Tizghadam and A. Leon-Garcia.Autonomic Traffic Engineering for Network Robustness.Vol 28, No. 1, Jan. 2010

[16] P. Van Mieghem and F. A. Kuipers. Concepts of Exact QoS Routing Algorithms .IEEE/ACM Transactions on Networking, 12(5):851–864, October 2004.

[17] M. Newman. A Measure of Betweenness Centrality Based on Random Walks. ArXiv cond-mat/0309045., 2003.

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

[19] S. Boyd and L. Vandenberghe. Convex Optimization.Cambridge University Press, 2004.

[20] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 2nd Edition.

[21] David Hock, Matthias Hartmann, Christian Schwartz, and Michael Menth, Effectiveness of Link Cost Optimization for IP Rerouting and IP Fast Reroute

AUTHOR

The author is pursuing B.Tech (Honours) degree in Electronics and Communication Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad and will be graduating in May 2018. His field of interest includes SDN, VLSI and computer architecture. His research experience includes an internship in DRDO and CDAC, Pune.