3-2011

Social Network Analysis Plugin (SNAP) for Mesh Networks

Soumendra Nanda
BAE Systems

David Kotz
Dartmouth College

Recommended Citation

Soumendra Nanda and David Kotz. Social Network Analysis Plugin (SNAP) for Mesh Networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), March 2011. 10.1109/WCNC.2011.5779252

Social Network Analysis Plugin (SNAP) for Mesh Networks

Soumendra Nanda
BAE Systems
Burlington, MA 01803

David Kotz
Dartmouth College
Hanover, NH 03755

***Abstract*—In a network, bridging nodes are those nodes that from a topological perspective, are strategically located between highly connected regions of nodes. Thus, they have high values of the Bridging Centrality (BC) metric. We recently introduced the Localized Bridging Centrality (LBC) metric, which can identify such nodes via distributed computation, yet has an accuracy equal to that of the centralized BC metric. The LBC and BC metrics are based on the Social Network Analysis (SNA) metric "betweenness centrality". We now introduce a new SNA metric that is more suitable for use in wireless mesh networks: the Localized Load-aware Bridging Centrality (LLBC) metric. The LLBC metric improves upon LBC by detecting critical bridging nodes while taking into account the actual traffic flows present in a mesh network. We only use local information from surrounding nodes to compute the LLBC metric, thus our LLBC metric is designed for scalable distributed computation and distributed network analysis. We developed the SNA Plugin (SNAP) for the Optimized Link State Routing (OLSR) protocol to study the potential use of LBC and LLBC in improving multicast communications. We present some promising initial results for SNAP from real and emulated mesh networks. SNAP is open source and free for academic use.**

I. Introduction

We recently developed a new distributed management system for wireless mesh networks, called Mesh-Mon, that can help a team of system administrators (sysadmins) manage a wireless mobile ad hoc network (MANET) or a mesh network [1]. Mesh-Mon is designed to provide scalable monitoring of large unplanned mesh networks, by allowing mesh nodes to cooperate locally amongst themselves to monitor each other and detect faults and anomalies in a decentralized manner. To complement the distributed nature of mesh networks and of our management platform, we seek to develop new distributed metrics that assist in network analysis and enhance the design of network routing protocols.

A sysadmin is generally concerned about which nodes are more "critical" and require more scrutiny in the network. One technique to identify the nodes that are critical from a network topology management perspective is to identify all "articulation points" and "bridges" in the network, since, upon failure, these nodes will partition a network [2], [3]. When applied to wireless mesh networks, in our experience, we found that articulation points are rare in practice in mesh topologies (unless the network is sparse and weakly connected). Thus, this technique is less helpful when applied in the analysis of mesh networks. Furthermore, Depth First Search (DFS) of the entire network is an essential computation and it can only be implemented efficiently in a centralized manner.

While most network management issues are absolute in nature (such as dealing with faulty hardware or incorrectly configured devices), there are many situations when relative management decisions must be made. For example, consider the following questions: If the system administrator had to update a subset of nodes and reboot them, then in which order should he or she perform the update? or Which nodes are the most and least "important" in my network?

Centrality is a concept often used in social network analysis (SNA) to study relative properties of social networks. These social networks are typically modeled as graphs. Our approach is to apply techniques adapted from SNA to answer relativistic questions. In a wireless mesh network context, a system administrator should pay attention to "bridging nodes" since they are important from a robustness perspective (as they help bridge connected components together) and their failure will increase the risk of network partitions.

This paper makes two main contributions: the development and evaluation of a new SNA-based centrality metric: the *Localized Load-aware Bridging Centrality (LLBC)* metric, that builds upon the benefits of our *Localized Bridging Centrality (LBC)* metric [4]. Our second contribution is the development of an OLSR plugin to study the applicability of LBC, LLBC and EigenVector Centrality (EVC) in mobile networks and evaluation via simulations in an emulated 802.11 environment using the Extendable Mobile Ad-hoc Emulator (EMANE) [5]. Both LLBC and LBC provide functionality that is comparable to or better than the Bridging Centrality (BC) metric [6] at identifying bridging nodes, yet can be calculated quickly and in a distributed manner. BC is calculated in a centralized manner using the entire network graph and has an order of magnitude higher computational complexity. To calculate its own LBC value, each node only needs to know its 1-hop neighbor set and the degree of each of its neighbors. Additionally, for LLBC calculations, each node only requires the aggregate traffic summary of its direct neighbors.

II. Social-Network Analysis

Our initial motivation for this work was to discover metrics and develop tools that can help a system administrator manage a wireless mesh network or would allow an automated management system understand the state of a network. We use "centrality" metrics from social-network analysis to study the roles of individual nodes in the network and the relationship of these nodes to their neighbors. Social-network analysis is normally applied to the study of social networks of people and organizations. In our domain, we are interested in the positions and roles of individual mesh nodes and the relationships between them, which like social networks are often represented as graphs.

The most commonly used social centrality metrics are degree centrality, closeness centrality and eigenvector centrality (EVC) [7]. Several other definitions of centrality measures exist. We focus here on sociocentric betweenness centrality [8].

A. Sociocentric betweenness centrality

The betweenness centrality of a node is calculated as the fraction of shortest paths between all node pairs that pass through a node of interest. A node with a high betweenness centrality value is more likely to be located on the shortest paths between multiple node pairs in the network, and thus more information must travel through that node (assuming a uniform distribution of information across node pairs). Since all pairs of shortest paths must be computed the time complexity is $\theta(n^3)$ where n is the number of nodes in the entire network. Brandes presents a fast technique to compute betweenness centrality that runs in $O(VE)$ time and uses $O(V + E)$ space for undirected unweighted graphs with V nodes and E edges [9].

B. Egocentric betweenness centrality

A more computationally efficient approach is to calculate betweenness on the "egocentric" (or ego) network, rather than the global network topology. In social networks, egocentric networks are defined as networks of a single actor together with the actors that they are directly connected to, that is, their neighbors in the graph. Thus, for wireless mesh networks we calculate egocentric betweenness on the one-hop adjacency matrix of a node. This metric can be calculated in a fully distributed manner and the computational complexity is $\theta(k^2)$ where k is size of the 1-hop neighborhood and is one order of magnitude faster than computing the global betweenness score.

Sociocentric betweenness centrality is a key component of the bridging centrality metric, while our metrics LBC and LLBC are based on egocentric betweenness centrality.

C. Bridging Centrality (BC)

Bridging Centrality (BC) is a centrality metric introduced by Hwang et al. [6]. Bridging centrality can help discriminate bridging nodes, that is, nodes with higher information flow through them, and locations between highly connected regions (assuming a uniform distribution of flows).

The Bridging Centrality of a node is the product of its sociocentric betweenness centrality C_{Soc} and its bridging coefficient $\beta(v)$. The Bridging Centrality $BC(v)$ for a node v of interest is defined as:

$$BC(v) = C_{Soc}(v) \times \beta(v) \tag{1}$$

The bridging coefficient of a node describes how well the node is located between high-degree nodes. The bridging coefficient of a node v is defined as:

$$\beta(v) = \frac{\frac{1}{d(v)}}{\sum_{i \in N(v)} \frac{1}{d(i)}} \tag{2}$$

where $d(v)$ is the degree of node v, and $N(v)$ is the set of neighbors of node v.

According to the authors, betweenness centrality indicates the importance of a given node from an information-flow standpoint, but it does not consider the topological position of the node. On the other hand, the bridging coefficient measures only how well a node is located between highly-connected regions, but does not consider information flow. "Bridging nodes" should be positioned between clusters and also located on important positions from an information-flow standpoint. Thus, their BC metric is an attempt to combine these two distinct metrics by giving equal weight to both factors. Based on their empirical studies, the authors recommend labeling the top 25th percentile of nodes as ranked by BC as "bridging nodes," that is, nodes that are more bridge-like and lie between different connected modules [6].

We note that these bridging nodes are different from the *articulation points* of a graph that one can discover during topological analysis (via DFS), though some bridging nodes are articulation points. These bridging nodes provide the system administrator with a prioritized set of critical nodes to monitor from a robustness perspective (as they help bridge connected components together) and their failure may increase the risk of network partitions. BC can only be calculated in a centralized manner with global information.

D. Localized Bridging Centrality (LBC)

In previous work, we introduced our distributed equivalent of Bridging Centrality that we call Localized Bridging Centrality (LBC) [4]. As the name suggests, we define $LBC(v)$ of a node v using only local information, as the product of egocentric betweenness centrality $C_{Ego}(v)$ and its bridging coefficient $\beta(v)$. The definition of $\beta(v)$ is unchanged from Equation 2. LBC is thus defined as:

$$LBC(v) = C_{Ego}(v) \times \beta(v) \tag{3}$$

Marsden [10] and Everett and Borgatti [11] showed empirically that egocentric betweenness values have a strong positive correlation to sociocentric betweenness values (calculated on the complete network graph) for many different network examples. While these networks were derived from social networks, in many cases they are similar to wireless mesh networks. Our LBC results are thus nearly as accurate as BC results, while being easier to compute with only local information. Prior to us, Daly and Haahr applied egocentric betweenness centrality in SimBet, a distributed routing protocol in a mobile delay-tolerant network (DTN) [12]. Their approach too benefits from the strong correlation between egocentric and sociocentric betweenness, but is designed for a DTN routing protocol. Our work focuses on routing protocols like OLSR and distributed network management for a MANET.

The LBC metric can help the system administrator identify the bridging nodes in the mesh network, as well as clusters and their boundaries, but its distributed nature makes it suitable for use in routing protocol design as well.

E. Localized Load-aware Bridging Centrality (LLBC)

Betweenness centrality implicitly assumes that all paths between all node-pairs are equally utilized. Thus, both the BC and LBC metrics assume that a uniform distribution of traffic flows will exist between all node-pairs in the network. In a real mesh network used to provide last-mile Internet access, the distribution of traffic flows will almost certainly be non-uniform and gateway nodes will experience relatively higher traffic loads.

Taking the traffic load into consideration, we now introduce our new Localized Load-aware Bridging Centrality (LLBC) metric designed for distributed analysis of bridging nodes in wireless mesh networks. We compute the traffic load (measured in bytes) in each node locally as the sum of all bytes originating at the node (Out), destined for the node (In), and twice the number of bytes forwarded (Fwd) by that node. We count the forwarded bytes twice in the summation since they are both received and sent by the node. In effect, this metric represents the load on the node's network interface.

$$Load(v) = In(v) + Out(v) + 2 \times Fwd(v) \tag{4}$$

We use the measured traffic load to calculate the Load Coefficient (β_t) as the ratio of the traffic load of a given node to the sum of the traffic loads of its one-hop neighbors. As the load of a node increases (relative to that of its neighbors' loads), so do the chances of the node becoming a traffic bottleneck.

$$\beta_t(v) = \frac{Load(v)}{\sum_{i \in N(v)} Load(i)} \tag{5}$$

We define LLBC as the product of Ego-Betweenness and the Load Coefficient.

$$LLBC(v) = C_{Ego}(v) \times \beta_t(v) \tag{6}$$

Thus, the LLBC takes into account both the current traffic load and the relative position of nodes, and (like the LBC metric) can be calculated in a fully distributed manner. Over time, the measured traffic load at different nodes will change and nodes that reboot will have their counters reset to zero. Thus, it makes sense to periodically sample LLBC values and to consider the traffic load during the sampling period instead of cumulative values.

It is important to remember that centrality measures can only provide *relative* measures that can be used to compare nodes against each other at that instant of time for that specific network topology. This ranking allows a system administrator to prioritize management tasks on several nodes, such as deciding which nodes should be updated first and in which order, or to identify which nodes are most likely to cause partitions through failure or mobility. Both of our metrics: LBC and LLBC, are easier to compute than the BC metric. A similar load-based bridging centrality can be applied to the study of road networks and airline paths. For wireless networks with multiple interfaces the load should be weighted relative to the available capacity of that link.

III. Evaluation

We now present our results from the application of the BC, LBC and LLBC metrics on the topology of a wireless mesh network we deployed in our department. We verified all calculations using UCINET, a popular SNA tool [13]. Two or more nodes with the same centrality value were assigned the same rank.

A. LLBC vs. LBC vs. BC

We used actual topologies from a mesh network test bed (called Dart-Mesh) that we deployed on all three floors of our department building [1].

The mesh nodes use the Optimized Link State Routing (OLSR) [14] mesh routing protocol implemented on Linux by Tønnesen [15]. The rectangles represent mesh nodes and are identified by the last octet of their individual IP addresses. The diamond-shaped box numbered zero is a virtual node representing the Internet.

1) Real-world mesh network with one gateway : We applied our Localized Load-aware Bridging Centrality (LLBC) metric on the network shown in Figure 2. Node 50 was the sole Internet Gateway providing Internet connectivity to the whole mesh. The topology of the network did not change during this experiment, which was 10 minutes long. The BC, LBC and LLBC results are presented in Table I and the nodes are sorted in decreasing order of LLBC values. The Load metric is in bytes.

During this experiment, node 80 had a high traffic load since we connected one of our mobile clients to that node, then proceeded to download large video files to that client from the Internet using node 50 as our Internet gateway. According to the LBC results, which only consider the topology of the network, node 30 was a more important "bridging node" than node 50. Node 30 is an articulation point in this example. However, our LLBC results accurately show that node 50 was the most important bridging node by taking into consideration the traffic load on the network during our experiment.

2) Real-world mesh network example with two gateways: We next applied our LLBC metric on a similar network topology similar to the one used in the last experiment by converting node 20 into an Internet gateway. The topology of this network is shown in Figure 4, and now nodes 50 and 20 are the two Internet gateways. The BC, LBC and LLBC results are presented in Table II and the results are sorted in decreasing order of LLBC values.

Since there were two Internet gateways, traffic flowing to and from the Internet could go through either gateway, depending on the route selected by the routing protocol. LBC

TABLE I
RANKED CENTRALITY VALUES FOR THE NETWORK SHOWN IN FIGURE 2, SORTED BY $LLBC$ VALUES

Node	Degree	Load	C_{Ego}	β	β_t	BC	LBC	$LLBC$
50	6	30871080	2	0.176	1.232	0.353	0.352	1.949
30	7	274027	10	0.0726	0.0043	0.8712	0.726	0.0438
80	5	30679118	0	0.219	0.962	0	0	0
1	5	262501	0	0.219	0.0042	0	0	0
2	5	238071	0	0.219	0.0038	0	0	0
20	5	218143	0	0.219	0.0035	0	0	0
160	2	94005	0	0.777	0.2571	0	0	0
90	2	91602	0	0.777	0.2488	0	0	0

TABLE II
RANKED CENTRALITY VALUES FOR THE NETWORK SHOWN IN FIGURE 4, SORTED BY $LLBC$ VALUES

Node	Degree	Load	C_{Ego}	β	β_t	BC	LBC	$LLBC$
50	6	32989000	2	0.118	1.123	0.354	0.236	2.246
30	7	305327	10	0.0739	0.0049	0.8868	0.738	0.0489
20	6	1125000	2	0.118	0.0183	0.354	0.236	0.0367
80	5	16208854	0	0.219	0.3512	0	0	0
1	5	11011448	0	0.219	0.2144	0	0	0
2	5	722022	0	0.2282	0.01171	0	0	0
90	2	145226	0	0.7778	0.3358	0	0	0
160	2	127098	0	0.7778	0.2820	0	0	0

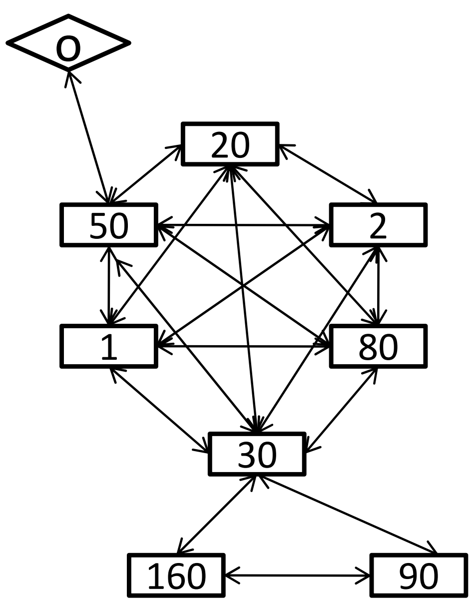

Fig. 1. Small single gateway mesh network

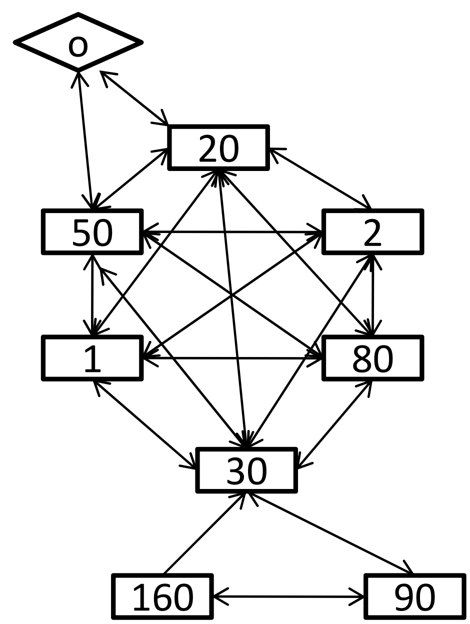

Fig. 2. Small mesh network with two gateways

picked node 30 as its top bridging node. While this node was indeed a critical node, there was little traffic flowing through this node, so it had little influence on the traffic flowing in the network or on the majority of the nodes, most of which were forwarding Internet-bound traffic through the two gateways.

LLBC picked node 50, in fact the most-heavily-used gateway node, as the most important bridging node and indicated that node 30 (a non-gateway node) was a more important bridging node than the gateway node 20, even though node 30 had only one fourth the traffic load of node 20 in absolute terms. The importance ranking generated by LLBC is insightful. In this scenario, if node 30 failed, then nodes 90 and 160 would be partitioned from the rest of the network. Whereas, if node 20 failed, there was still a potential backup path to the Internet through 50; the LBC rankings were unable to capture this subtle complexity present in this network. The BC ranking was identical to the LBC ranking, and thus not as helpful as the LLBC metric in this scenario. The distributed manner in which LLBC is calculated also complements a distributed network analysis system (such as Mesh-Mon).

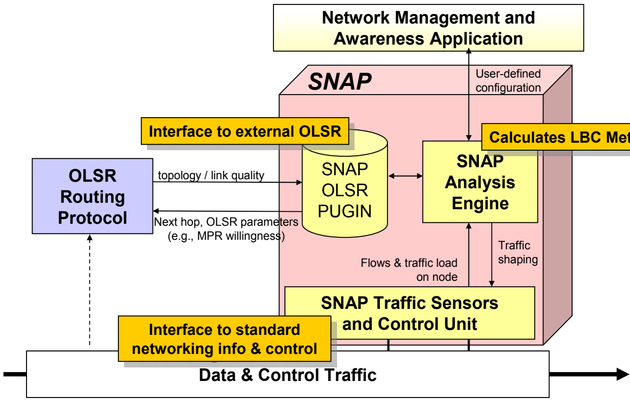

Fig. 3. SNA Plugin (SNAP) Architecture

IV. SNA Plugin (SNAP) for OLSR

To study the utility of LBC, LLBC and EVC in an OLSR-based MANET, we developed a Social Network Analysis Plugin (SNAP) as shown in Figure 6. While we use OLSR for our example, the same design can potentially benefit other MANET routing protocols.

OLSR is a unicast routing protocol but it floods all multicast traffic via Multi-Point Relays (MPRs) in the Basic Multicast Forwarding (BMF) plugin extension. We developed a simple distributed algorithm that ranks neighbors according to calculated LBC or LLBC scores and then each node locally adjusts its own advertised MPR-Willingess parameter up or down as per its relative ranking. Our initial hypothesis was that strong bridging nodes would serve as good MPRs for multicast communications. Our second hypothesis that we have not yet explored in depth is that LLBC can be used to enable better selection of load balanced paths in a mesh network (since LLBC can detect bottlenecks). Eigenvector Centrality (EVC) is also computed and reported for use in offline analysis in our plugin, but is not used to modify OLSR behavior at present. Recently, Gao et al. [16] explored the use of a new centrality measure for DTNs based on Poisson modelling of contacts using the egocentric network model to enhance multicast communications. Our approach does not buffer data and is designed for use in MANETs using OLSR, but both approaches use a similar idea of selecting better and fewer relays to improve multicast delivery of data and the use of egocentric network models.

A. Initial SNAP Results

We tested our SNAP plugin on a few emulated 802.11b topologies on EMANE while running a video multicast traffic application (generated by Mgen and NORM) using the BMF plugin version 1.7 for OLSR version 6.0. Our SNAP plugin recomputed LBC and LLBC values and changes to MPR_Willingness every 10 seconds.

We compared performance on the basis of a custom video utility metric. This metric takes into account a combination of the the latency of packets received and number of frames that have been dropped.

Our initial test was with a 6 node linear string with two sources at opposite ends and with each source also acting as destination. In this example, we found no difference in performance between the default BMF multicast and the LBC or LLBC influenced multicast. This is unsurprising since every node in the string must forward all multicast traffic that it receives.

We then tested our plugin on emulated scenarios with upto 23 mobile nodes (See Figure 8) with upto 6 multicast video sources and upto 19 destinations for 300 seconds. The scenarios use GPS logs and pathloss recordings from an outdoor experiment with OLSR nodes and our emulation test range provides performance similar to that recorded in those real experiments. Most of the nodes moved at a slow walking speed and two nodes moved in two vehicles at 10 MPH (along the purple lines in Figure. 8). We repeated each experiment three times and reported the average.

The average performance of our LBC and LLBC enhanced multicast strategy showed some improvements (See Table IV-A over the default behavior of BMF and in particular, the performance of SNAP-LBC was the best overall. Our analysis of the experiment logs indicated that SNAP and BMF were initially selecting the same MPRs for forwarding multicast traffic. We are uncertain if the heuristic used by SNAP-LBC was leading to an optimal MPR coverage (in our tested scenarios), but the results do provide early evidence for our first hypothesis.

The SNAP-LLBC tests showed more variance in the selection of MPRs because it reflected the changes in actual traffic flowing through individual nodes during the experiment. We suspect that this was due to the highest ranked nodes being overloaded or in some cases being clustered together and thus leading to too many redundant MPRs. We need to explore more topologies (real and simulated) and alternative MPR selection strategies, before we can conclude whether the use of higher ranked bridging nodes as MPRs is always preferable to the default strategy used by OLSR, but our initial results with SNAP using the LBC metric look very promising.

We are pleased to share the source code for our OLSR plugin for the benefit of the academic community to use in different mesh networks and to extend its functionality with further development.

V. Conclusion

In this paper we demonstrate the use of novel social network metrics to solve the problem of identifying important nodes in wireless mesh networks for system administrators. We introduce a new centrality metric called the Localized Load-aware Bridging Centrality (LLBC). Our evaluation of the LLBC and LBC metrics on a real mesh testbed running OLSR indicate its potential for use in routing and network analysis tools.

We demonstrated the usefulness of LLBC in identifying critical bridging nodes in a wireless mesh network from a network management perspective. Our initial results from our OLSR plugin shows that our SNA-based approach to selecting

TABLE III
VIDEO METRIC UTILITY SCORES FOR SNAP

	BMF	SNAP-LBC	SNAP-LLBC
6 node linear static	0.91	0.91	0.91
23 node mobile test	0.47	0.66	0.51

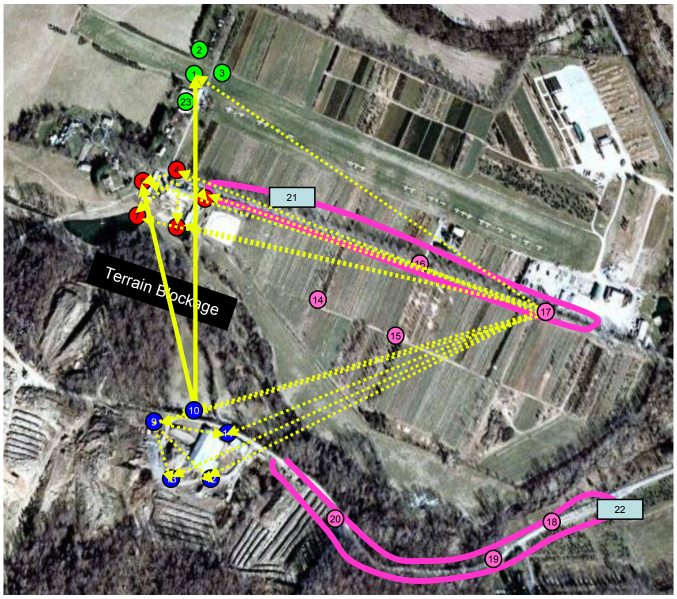

Fig. 4. 23 Mobile Node Test Snapshot

MPRs for multicast in OLSR when using the LBC metric is beneficial in certain topologies. We are in the process of testing the properties of our new metrics on larger mesh data sets (both simulated and from real deployments) and exploring its utility in other scenarios and application domains.

We acknowledge that further evaluations are needed on existing large-scale mesh networks or sensor networks with real traffic patterns and through further simulations using our SNA plugin for OLSR and with other routing protocols. We are also exploring other variants of LLBC and LBC that take into account link-quality measures, link capacities, and other real-world effects

While we focus on the distributed analysis of a wireless mesh network topology in this paper, our LBC and LLBC metrics have potential applications in other disciplines as well, such as for analysis of social networks, online collaboration tools, and identifying clusters and key components in complex biological structures or bottlenecks in transportation systems such as inter-state highways and flight plans.

ACKNOWLEDGMENT

We thank Charles Tao at BAE Systems for his help in coding the OLSR SNA plugin. This research program was supported by a gift from Intel Corporation, by Award number 2000-DT-CX-K001 from the U.S. Department of Homeland Security, by Grant number 2005-DD-BX-1091 from the Bureau of Justice Assistance and by Contract number N00014-10-C-098 from the Office of Naval Research (ONR). Points of view in this document are those of the authors, and do not necessarily represent reflect the views of any of the sponsors, the US Government or any of its agencies.

REFERENCES

[1] S. Nanda and D. Kotz, "Mesh-Mon: A multi-radio mesh monitoring and management system," *Computer Communications*, vol. 31, no. 8, pp. 1588–1601, 2008.

[2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, Mass.: MIT Press, 2001.

[3] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, p. 146, 1972.

[4] S. Nanda and D. Kotz, "Localized bridging centrality for distributed network analysis," in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN)*, 2008, pp. 1–6.

[5] CenGen Inc., "Extendable Mobile Ad-hoc Network Emulator," http://labs.cengen.com/emane/.

[6] W. Hwang, Y. Cho, A. Zhang, and M. Ramanathan, "Bridging Centrality: Identifying Bridging Nodes in Scale-free Networks," Department of Computer Science and Engineering, University at Buffalo, Tech. Rep. 2006-05, March 15 2006.

[7] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[8] L. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[9] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[10] P. Marsden, "Egocentric and sociocentric measures of network centrality," *Social Networks*, vol. 24, no. 4, pp. 407–422, 2002.

[11] M. Everett and S. Borgatti, "Ego network betweenness," *Social Networks*, vol. 27, no. 1, pp. 31–38, 2005.

[12] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Montreal, Quebec, Canada: ACM Press, 2007, pp. 32–40. DOI 10.1145/1288107.1288113

[13] S. Borgatti, M. Everett, and L. Freeman, "UCINET for Windows: Software for Social Network Analysis," *Harvard: Analytic Technologies*, 2002.

[14] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Oct. 2003. Available online: http://www.ietf.org/rfc/rfc3626.txt

[15] A. Tønnesen, "Implementing and extending the Optimized Link State Routing protocol," Master's thesis, Master's thesis, University of Oslo, Norway, 2004.

[16] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc)*, 2009, pp. 299–308.