Chapter 7
Localized Bridging Centrality

Soumendra Nanda and David Kotz

Abstract Centrality is a concept often used in social network analysis (SNA) to study different properties of networks that are modeled as graphs. Bridging nodes are strategically important nodes in a network graph that are located in between highly-connected regions. We developed a new centrality metric called Localized Bridging Centrality (LBC) to allow a user to identify and rank bridging nodes. LBC is a distributed variant of the Bridging Centrality (BC) metric and both these metrics are used to identify and rank bridging nodes. LBC is capable of identifying bridging nodes with an accuracy comparable to that of the BC metric for most networks, but is an order of magnitude less computationally expensive. As the name suggests, we use only local information from surrounding nodes to compute the LBC metric. Thus, our LBC metric is more suitable for distributed or parallel computation than the BC metric. We applied our LBC metric on several examples, including a real wireless mesh network. Our results indicate that the LBC metric is as powerful as the BC metric at identifying bridging nodes. We recently designed a new SNA metric that is also 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 communications network. 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 and our initial results are promising. In this chapter, we present an introduction to SNA centrality metrics with a focus on our contributed metrics: LBC and LLBC. We also present some initial results from applying our metrics in real and emulated wireless mesh networks.

S. Nanda (✉)
BAE Systems, 6 New England Executive Park, Burlington, MA 01821, USA
e-mail: soumendra.nanda@baesystems.com

D. Kotz
Dartmouth College, 6211 Hinman, Hanover, NH 03755, USA
e-mail: kotz@cs.dartmouth.edu

M.T. Thai and P.M. Pardalos (eds.), *Handbook of Optimization in Complex Networks: Communication and Social Networks*, Springer Optimization and Its Applications 58,
DOI 10.1007/978-1-4614-0857-4_7,

7.1 Introduction

Wireless mesh networks are an emerging type of wireless network technology that are related to mobile ad hoc networks (MANETs). They are used to provide network access to wireless clients (such as laptops for e.g.), but unlike cellular wireless networks (where cell towers communicate with each other through a wired backbone), the mesh nodes communicate with each other through wireless channels and in a distributed and decentralized manner with the help of a MANET routing protocol. Optimized Link State Routing (OLSR) [16] is one such MANET routing protocol, but there are hundreds of others in the literature that could be used for mesh networks. Most wireless networks are represented as graphs with the mesh nodes as vertices and with the links between neighboring nodes as edges. Link quality values are often used as weights on these edges.

During the course of a PhD dissertation at Dartmouth College in Hanover, NH, we developed a new distributed management system for wireless mesh networks, called Mesh-Mon, that can help a team of system administrators (sysadmins) manage a stationary or mobile wireless mesh network [18, 20]. Mesh-Mon is designed to provide scalable monitoring of large unplanned wireless mesh networks, by allowing mesh nodes to cooperate locally to monitor each other and detect faults and anomalies in a decentralized manner. Most traditional network analysis tools for computer networks are centralized in nature. To complement the distributed nature of mesh networks and of our management platform, we sought to develop new distributed metrics and tools that may assist in network analysis and could enhance the design of future network routing protocols.

We provide below a list of questions asked from a sysadmin's point of view, that we initially set out to answer and consider relevant to mesh networks:

1. Which nodes should the system administrator be most concerned about from a robustness point of view? That is, the loss of which nodes would have a significant impact on the connectivity of the network?
2. How many nodes can fail before the network is partitioned into multiple parts?
3. Which nodes are the most "important" in the network?
4. If the sysadmin could or should add or move a node to enhance the network, which one should it be?
5. Similarly, if the sysadmin had to update a subset of nodes and reboot them, in which order should the update be performed in?

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 [10, 24]. 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 computational step 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 sequentially, 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 chapter provides an introduction to centrality metrics used in social network analysis and describes some of our own recently published contributions: the development and evaluation of two new SNA-based centrality metrics: the *Localized Load-aware Bridging Centrality (LLBC)* metric, and the *Localized Bridging Centrality (LBC)* metric [18–21]. 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) by CenGen Inc [8]. Both LLBC and LBC provide functionality that is comparable to or better than the Bridging Centrality (BC) metric [15] 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.

7.2 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.

Many social-network analysis techniques and metrics are based on graph theory. Humans tend to form clusters of communities within social networks. Similarly, mesh networks may have a complex network structure with groups of nodes that share a common relationship or structure that is worth identifying. Although we refer to wireless mesh networks as our primary domain for the application of our techniques, they are general enough to apply to other domains that use graphs to represent complex networks.

The connectivity relationship between different mesh nodes can be characterized in many different ways, such as direct or indirect, and weak or strong. The position a node occupies in a network can play a role in the node's ability to control or impact the flow of information in the network. Centrality indices help characterize the position of a node in different ways.

The most common centrality metrics used in social network analysis are degree centrality, closeness centrality, eigenvector centrality (EVC) [2] and sociocentric betweenness centrality [13]. Several other definitions of centrality measures exist. A popular software package for experimenting with Social Network Analysis metrics is provided by Analytic Tech [5], but there are many other tools available. We focus on sociocentric betweenness centrality and use it to develop our own centrality measures later in this text.

We believe that techniques borrowed and enhanced from the domain of social network analysis can help in providing answers to some of the questions we pose. We aim to 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 actors, usually people and their relationships with other people. In our domain, we are interested in the positions and roles of individual mesh nodes and the relationships between them.

7.2.1 Degree Centrality

One simple way to characterize an individual node in a topological graph is by its *degree*. The *degree* of a node in a graph in the mesh context is the number of links the node shares with its neighbors and which are available for routing purposes. A well-connected mesh network is a healthy network. If a node has many neighbors then the failure of a single neighbor should not affect the routing health of the regional network adversely. A node with a high degree can be considered as being well connected and a node with a relatively low degree can be considered weakly connected. The degree of an individual node and the minimum, maximum and average degree over all the nodes are standard characterization metrics in graph theory.

If the global topology is available at a central location, then all the nodes can be quickly ranked according to their degree. However, this degree-based ranking does not convey a good picture of the nature of connectivity in the network since

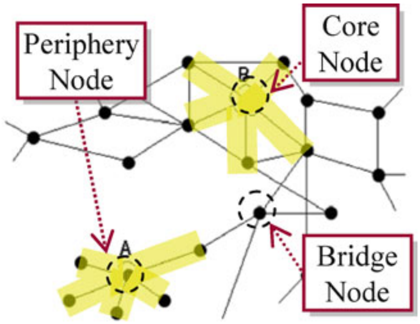

Fig. 7.1 Limitations of degree centrality. Note that nodes A and B each have degree 5 but are likely to have different impact on a network. (Adapted from [1])

all links are rarely identical. For instance, different links may have varying capacity levels and different latencies. In addition, the existence of neighbor links and their respective qualities fluctuate over time. In a wireless network, a link with a poor-quality connection has lower effective capacity and a link using a lower bit-rate may have a higher latency (Fig. 7.1).

Furthermore, two nodes may have the same degree, but they may have very different characteristics due to their relative positions [1]. There are other centrality metrics, such as eigenvector centrality, that can help distinguish between nodes A and B that have the same degree centrality.

7.2.2 Eigenvector Centrality

Eigenvector Centrality (EVC) is a concept often used in social-network analysis and was first proposed by Bonacich [2, 3]. Eigenvector Centrality is defined in a circular manner. The centrality of a node is proportional to the sum of the centrality values of all its neighboring nodes. In the social-network context, an important node (or person) is characterized by its connectivity to other important nodes (or people). A node with a high centrality value is a well-connected node and has a dominant influence on the surrounding network. Similarly, nodes with low centrality values are less similar to the majority of nodes in the topology and may exhibit similar characteristics and behavior and share common weaknesses. Google uses a similar centrality ranking technique (called Pagerank) to rank the relevance of hyper-linked pages in search results [7].

Eigenvector centrality is calculated using the adjacency matrix to find central nodes in the network. Let v_i be the ith element of the vector $\mathbf{v}$, representing the centrality measure of node i, where $N(i)$ is the set of neighbors of node i and let A be the $n \times n$ adjacency matrix of the undirected network graph. Eigenvector centrality is defined using the following formulas:

$$v_i \propto \sum_{j \in N(i)} v_j \tag{7.1}$$

which can be rewritten as

$$v_i \propto \sum_{j=1}^{n} A_{ij} v_j \tag{7.2}$$

which can be rewritten in the form

$$A\mathbf{v} = \lambda \mathbf{v} \tag{7.3}$$

Since A is an n x n matrix, it has n eigenvectors (one for each node in the network) and n corresponding eigenvalues. One way to compute the eigenvalues of a square matrix is to find the roots of the characteristic polynomial of the matrix. It is important to use symmetric positive real values in the matrix used for calculations [4].

The principle eigenvector is the eigenvector with the highest eigenvalue. The principle eigenvector is recommended for use in rank calculations [2]. After the principle eigenvector is found, its entries are sorted from highest to lowest values to determine a ranking of nodes. The most central node has the highest rank and most peripheral node has the lowest rank.

This metric is often used in the study of the spread of epidemics in human networks. In the mesh context, a node with a high eigenvector centrality represents a strongly connected node. A worm or virus propagated from the most central node could spread to all reachable nodes in the most efficient manner as opposed to one that was spreading from a node on the extreme periphery. Thus, the central node is a prime target for preventive inoculation or for prioritized software update.

In any network, and especially in an ad hoc or mesh network where nodes must cooperate with each other to route packets, the connectivity of a node depends on the connectivity of its neighbors and EVC can help capture this property. The main drawback of eigenvector centrality is that it can only be calculated in a centralized manner.

7.2.3 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 flow 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 [6].

7.2.4 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, an egocentric network is defined as network of a single actor (or node) together with the actors (also referred to as alter-egos) that this actor are directly connected to, that is, the node's neighbors in the graph and all the corresponding links between these nodes. For application in wireless networks, the ego network is exactly the same as the 1-hop adjacency matrix representation of the connectivity graph centred around any given node of interest.

Thus, for wireless mesh networks we calculate egocentric betweenness on the one-hop adjacency matrix of a node and not the entire network (as would be the case for calculating betweenness in the sociocentric betweenness metric). This egocentric betweenness 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 thus this computation is one order of magnitude faster than computing the global betweenness centrality 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.

The simple structure of an ego network allows for fast computation of the egocentric betweenness centrality (also referred to as ego betweenness). Everett and Borgatti developed the following fast technique (and illustrative example) to calculate ego betweenness [12]. If the ego network is represented as a $n \times n$ symmetric matrix A where 1 represents a link between node ii and node j and 0 represents the absence of a link, then the betweenness of the ego node is the some of the reciprocals of the entries of $A^2[1-A]_{i,j}$. This value has to be divided by two if the graph is symmetric.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.4}$$

For a faster implementation, first note which entries in $A^2[1-A]_{i,j}$ will be non-zero (for a symmetric matrix, we only need to consider the zero entries above the leading diagonal) and calculate $A^2[1-A]_{i,j}$ only for those entries.

$$A^2[1-A]_{i,j} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & 2 & 1 \\ * & * & * & * & 1 \\ * & * & * & * & 1 \\ * & * & * & * & * \end{bmatrix} \tag{7.5}$$

The ego betweenness is the sum of the reciprocals of the entries of interest and in this example it is $\frac{1}{2} + 1 + 1 + 1 = 3.5$.

7.2.5 Bridging Centrality

Bridging Centrality (BC) is a centrality metric introduced by Hwang et al. [15]. 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 $\text{BC}(v)$ for a node v of interest is defined as:

$$\text{BC}(v) = C_{\text{Soc}}(v) \times \beta(v) \tag{7.6}$$

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:

$$\beta(v) = \frac{\frac{1}{d(v)}}{\sum_{i \in N(v)} \frac{1}{d(i)}} \tag{7.7}$$

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 [15].

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.

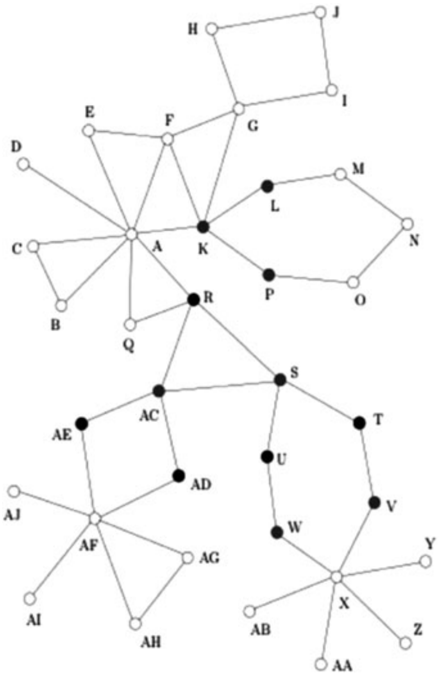

Fig. 7.2 Top bridging nodes as identified and ranked by Localized Bridging Centralityn with higher ranked nodes *shaded darker*. (Adapted from [15])

7.2.6 Localized Bridging Centrality

In previous work, we introduced our distributed equivalent of Bridging Centrality that we call Localized Bridging Centrality (LBC) [19]. As the name suggests, we define LBC(v) of a node v using only local information, as the product of egocentric betweenness centrality $C_{\text{Ego}}(v)$ and its bridging coefficient $\beta(v)$. The definition of $\beta(v)$ is unchanged from (7.7). LBC is thus defined as (Fig. 7.2):

$$\text{LBC}(v) = C_{\text{Ego}}(v) \times \beta(v) \tag{7.8}$$

Marsden [17] and Everett and Borgatti [12] 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 also benefit from similar correlations and are thus nearly as accurate as BC results, while being easier to compute with only local information. As you can see in Sect. 7.2.5, LBC is able to identify key bridge nodes and articulation points.

Prior to us, Daly and Haahr applied egocentric betweenness centrality to develop SimBet, a distributed routing protocol in a mobile delay-tolerant network (DTN) [11]. 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 real-time network routing protocols like OLSR and for 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. While individual nodes can calculate their own LBC metric in a fully distributed manner, to determine the global rank of each node, a central node must aggregate all LBC values or all nodes must use a distributed consensus-based ranking algorithm.

7.2.7 Localized Load-Aware Bridging Centrality

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 developed our new Localized Load-aware Bridging Centrality (LLBC) metric designed for distributed analysis of bridging nodes in wireless mesh networks [21]. 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.

$$\mathrm{Load}(v) = \mathrm{In}(v) + \mathrm{Out}(v) + 2 \times \mathrm{Fwd}(v) \tag{7.9}$$

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.

$$\beta_{\mathrm{t}}(v) = \frac{\mathrm{Load}(v)}{\sum_{i \in N(v)} \mathrm{Load}(i)} \tag{7.10}$$

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

$$\mathrm{LLBC}(v) = C_{\mathrm{Ego}}(v) \times \beta_{\mathrm{t}}(v) \tag{7.11}$$

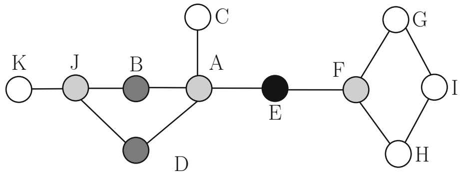

Fig. 7.3 A small synthetic network example with its top six high bridging score (via LBC) nodes *shaded* (Adapted from [15])

Thus, the LLBC metric 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 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.

7.3 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 [5]. Two or more nodes with the same centrality value were assigned the same rank.

7.3.1 Synthetic Network Example

We first tested our LBC metric using a synthetic network, presented in Fig. 7.3 (this network was also used by Hwang et al. [15]). The rankings produced by Bridging Centrality and Localized Bridging Centrality shown in Table 7.1 are nearly identical, although we note that the BC and LBC values are clearly not identical, nor are the betweenness measures used. Since both BC and LBC are used as a "relative" measure of how nodes differ from each other, the induced ranking is more important than the magnitude of the BC or LBC value and thus in this example our LBC metric is exactly equivalent to the BC metric.

Table 7.1 Top six centrality values for the network shown in Fig. 7.3, including sociocentric betweenness (C_{Soc}), egocentric betweenness (C_{Ego}), bridging coefficient (β), bridging centrality (*BC*) and localized bridging centrality (*LBC*)

Node	Degree	C_{Soc}	C_{Ego}	β	BC	LBC	Rank of BC	Rank of LBC
E	2	0.533	1	0.857	0.457	0.857	1	1
B	2	0.155	1	0.857	0.133	0.857	2	1
D	2	0.155	1	0.857	0.133	0.857	2	1
F	3	0.477	3	0.222	0.106	0.666	4	4
A	4	0.655	6	0.100	0.065	0.600	5	5
J	3	0.211	3	0.166	0.035	0.499	6	6

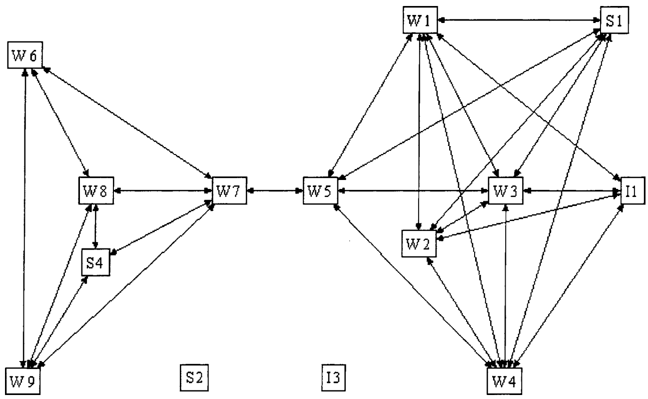

Fig. 7.4 Bank wiring room games example (Adapted from [17])

7.3.2 *Social Network Example: Bank Wiring Room*

This example (presented in Fig. 7.4) represents game-playing relationships in a bank wiring room and is popular in social-network studies. Marsden [17] presented this example to show how sociocentric and egocentric betweenness measures correlate. Again, the relative ranking of nodes calculated by BC and LBC as shown in Table 7.2 are identical. Visible inspection of Fig. 7.4 shows that nodes W5 and W7 are bridging nodes, and the tie between them is a bridge between two connected components.

7.3.3 *Wireless Mesh Network Examples*

We present actual topologies from a mesh network test bed (called Dart-Mesh) that we deployed on all three floors of our department building [20].The mesh nodes use the Optimized Link State Routing (OLSR) [9] mesh routing protocol implemented on Linux by Tønnesen [25]. In Fig. 7.5, the rectangles represent mesh nodes and

Table 7.2 Centrality values for the network shown in Fig. 7.4 sorted by BC values

Node	Degree	C_{Soc}	C_{Ego}	β	BC	LBC	Rank of BC	Rank of LBC
W5	5	30.00	4.00	0.222	6.667	0.889	1	1
W7	5	28.33	4.33	0.179	5.074	0.775	2	2
W1	6	3.75	0.83	0.140	0.528	0.117	3	3
W3	6	3.75	0.83	0.140	0.528	0.117	3	3
W4	6	3.75	0.83	0.140	0.528	0.117	3	3
S1	5	1.50	0.25	0.222	0.333	0.055	6	6
W8	4	0.33	0.33	0.223	0.073	0.073	7	7
W9	4	0.33	0.33	0.223	0.073	0.073	7	7
W2	5	0.25	0.25	0.210	0.052	0.052	9	9
W6	3	0	0	0.476	0	0	10	10
S4	3	0	0	0.476	0	0	10	10
I1	4	0	0	0.357	0	0	10	10
I3	0	0	0	0	0	0	10	10
S2	0	0	0	0	0	0	10	10

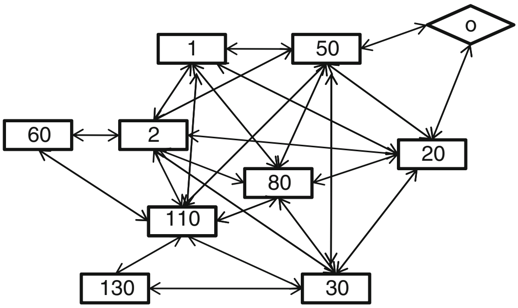

Fig. 7.5 A small OLSR mesh network that was deployed at Dartmouth College

Table 7.3 Ranked centrality values for Fig. 7.5 sorted by BC values

Node	Degree	C_{Soc}	C_{Ego}	β	BC	LBC	Rank of BC	Rank of LBC
110	7	6.367	5.75	0.078	0.496	0.448	1	1
50	7	4.733	3.40	0.096	0.454	0.326	2	3
30	6	3.367	2.75	0.132	0.444	0.363	3	2
2	7	4.067	3.4 0	0.096	0.391	0.326	4	3
20	6	2.867	2.25	0.126	0.361	0.283	5	5
80	6	0.400	0.40	0.173	0.069	0.069	6	6
1	5	0.200	0.25	0.262	0.052	0.065	7	7
60	2	0	0	1.615	0	0	8	8
130	2	0	0	1.650	0	0	8	8
0	2	0	0	1.615	0	0	8	8

are identified by the last octet of their individual IP addresses. The diamond-shaped box numbered zero is a virtual node representing the Internet. Nodes connected to the Internet are the Internet Gateways.

In the example in Fig. 7.5, while the two rankings produced (shown in Table 7.3) by the two metrics are not identical, they are quite similar. The top-five ranked nodes

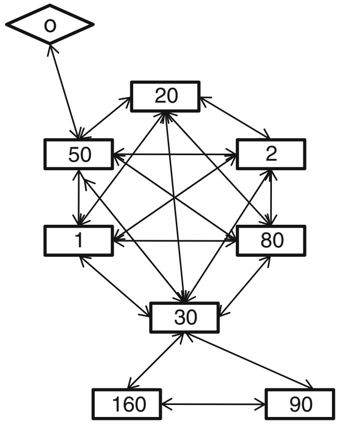

Fig. 7.6 A small real-world mesh network with one gateway at Node 50. Both Node 30 and node 50 are articulation points

are common to both metrics. If we remove any of these bridging nodes, then at least one of the other bridging nodes will become an articulation point. If that node is now removed, the network will be partitioned. For example, if node 110 (the top-ranked-node) fails, then both nodes 30 and 2 become articulation points. However, if you examine the original network graph in Fig. 7.5, you will find that it is bi-connected and has no articulation points. Thus, LBC can help detect nodes that may not be articulation points, but with certain perturbations in the network are the most likely candidates to become articulation points. Our LBC metric allows the system administrator to gather this information in a distributed manner using fewer computational resources than the BC metric.

7.3.4 LLBC vs. LBC vs. BC

7.3.4.1 Real-World Mesh Network with One Gateway

We now present examples where we considered how the volume and direction of traffic flowing in the network affected our metrics. We applied our Localized Load-aware Bridging Centrality (LLBC) metric on the network shown in Fig. 7.6. 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 min long. The BC, LBC, and LLBC results are presented in Table 7.4 and the nodes are sorted in decreasing order of LLBC values. The Load metric is given here 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

Table 7.4 Ranked centrality values for the network shown in Fig. 7.6, 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

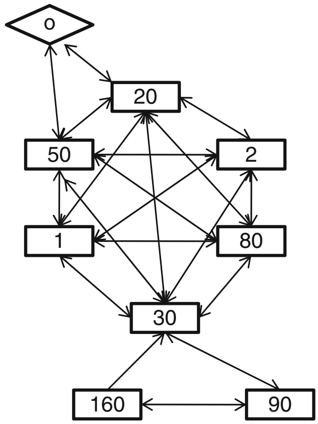

Fig. 7.7 A small mesh network with two gateways. Node 50 and node 20 are the two Internet gateways but Node 30 is an articulation point in the network

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.

7.3.4.2 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 Fig. 7.7, and now nodes 50 and 20 are the two Internet gateways. The BC, LBC and LLBC results are presented in Table 7.5 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 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.

Table 7.5 Ranked centrality values for the network shown in Fig. 7.7, 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

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 analysis engine, such as the one used in Mesh-Mon.

7.4 SNA Plugin (SNAP) for OLSR

To study the practical utility of LBC, LLBC, and EVC in an OLSR MANET for analysis and performance improvement, we developed a Social Network Analysis Plugin (SNAP) as shown in Fig. 7.8 [21]. While we use OLSR for our example, the same design can potentially benefit other MANET routing protocols.

OLSR is a unicast protocol but 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 1-hop neighbors according to calculated LBC or LLBC scores and then each node locally adjusts its own advertised MPR-Willingess parameter slightly up or down as per its relative ranking.

The MPR-Willingness parameter is used by OLSR running on each node to decide if it should become an MPR and can range from 0 (never become an MPR) to 7 (always be an MPR). Having too many MPRs leads to excessive flooding and waste of spatial resources while having too few will lead to insufficient coverage and poor distribution of topological information in the network. We did not modify any of OLSR's internals. We only modify one parameter that can influence how

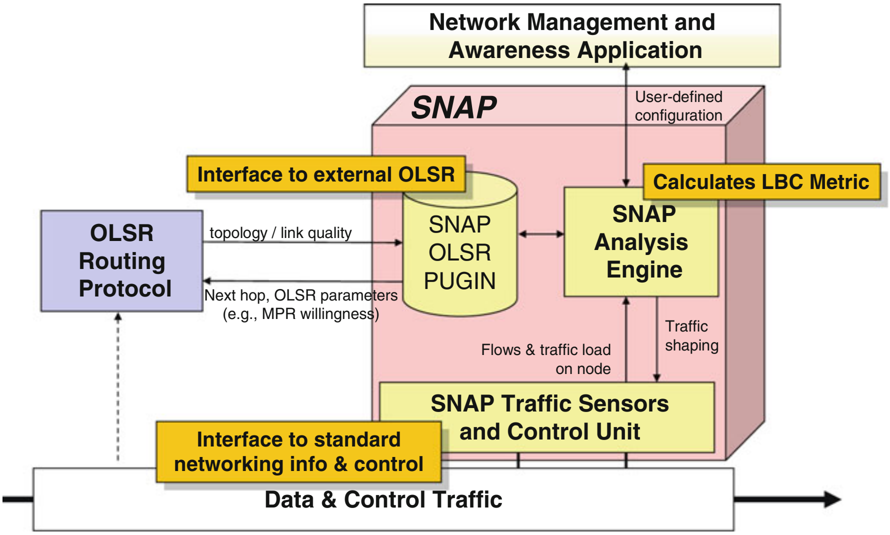

Fig. 7.8 SNA Plugin (SNAP) architecture

OLSR chooses the MPRs that it prefers to use for routing, topology distribution and multicast of packets.

Our SNA Plugin (SNAP) architecture uses a simple design and executes the following series of five sequential operations in a continuous loop every 10 s:

1. Acquire local network topology state from OLSR Routing Protocol
2. Calculate LBC metric (within the SNAP Analysis Engine)
3. Distribute local LBC metric to all 1-hop neighbors (single broadcast)
4. Compare local LBC metric value with that of all received LBC values from ego network
5. Adjust MPR Willingness locally based on relative differences in LBC scores in the ego network

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. Recently, Gao et al. [14] explored the use of a new centrality measure for Delay Tolerant Networks (DTNs) based on Poisson modelling of contacts using the egocentric network model to enhance multicast communications. 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.

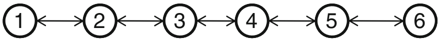

Fig. 7.9 Six node linear string topology. Any node can communicate only with its linked neighbors, but not with other nodes

SNAP-LBC and SNAP-LLBC results below show how our metrics compare against the default OLSR-BMF setup. SNAP-Degree results demonstrate the impact of substituting LBC (or LLBC) metric in our original SNAP algorithm described above with *degree centrality*. Degree centrality is a much simpler SNA metric (conceptually and computationally) than LBC or LLBC, since it is just the number of neighbors that a given node is connected to directly.

7.4.1 Initial SNAP Results

We tested our SNAP plugin on a few emulated 802.11b topologies while running a video multicast traffic application generated by the open-source Multi-Generator (MGEN) [22] software tool (developed by Naval Research Laboratory) and by using the Basic Multicast Forwarding (BMF) plugin version 1.7 for OLSR version 0.6.1 (available for download at www.olsr.org). Our SNAP plugin recomputed LBC and LLBC values and made changes to the local MPR_Willingness parameter every 10 seconds. Each source uses the MGEN tool to generate video traffic by sending packets of 1,024 bytes at 24 packets/second. For all video traffic, we added additional reliability using NACK-Oriented Reliable Multicast (NORM) protocols with Forward Error Correction with 2x redundancy (block 4, parity 4 settings) [23].

We compared performance on the basis of a custom video utility metric that was developed by an external third party. This metric takes into account a combination of the latency of packets received and number of frames that have been dropped. The maximum possible score is one indicating perfect video reception and minimum is zero. The video metric captures the notion that a video receiver is satisfied with a 10 seconds chunk of transmitted video if it receives 90% of the packets in that chunk within 4 seconds of end-to-end delay. The entire scenario is broken into 10 seconds of time slices and a time slice video utility score is computed for each slice for all intended destination receivers. All these time slice utilities are combined into an average scenario score for the whole experiment.

7.4.1.1 Six Node Static Linear Topology

Our first test for SNAP-enhanced version of OLSR was with a six node linear string topology with two sources for video traffic at opposite ends (node 1 and node 6 in Fig. 7.9) and with each source also acting as destination for the other source.

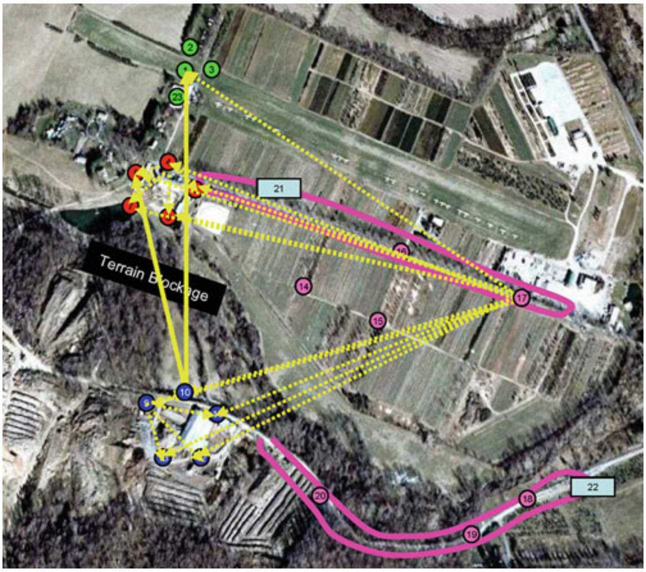

Fig. 7.10 Snapshot from the 23 node mobile test with four multicast video sources and 19 video destinations

In this example, we found no difference in performance between the default BMF multicast and the LBC or LLBC influenced multicast. We repeated each experiment three times and reported the average. The results match our expectations, since every node in the string must forward all multicast traffic that it receives and there is no alternate path for the multicast traffic and no room for optimization or improvement.

7.4.1.2 Twenty Three Node Real World Topology

We then tested our plugin on emulated scenarios with upto 23 mobile nodes (See Fig. 7.10) with upto four multicast video sources and upto 19 destinations for 300 s. 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 thicker curved shaded paths in Fig. 7.10). Traffic was sent to local clusters as well as clusters of nodes far away to ensure that we had multi-hop communications. We repeated each experiment three times and reported the average.

The average performance of our LBC and LLBC enhanced multicast strategy showed some significant improvements (See Table 7.6) over the default behavior of BMF and in particular, the performance of SNAP-LLBC was the best overall. We tested the performance of SNAP-LBC, SNAP-LLBC, and SNAP-degree at higher

Table 7.6 Video metric utility scores for SNAP

	OLSR-BMF	SNAP-LBC	SNAP-LLBC	SNAP-Degree
6 node linear static (1x load)	0.91	0.91	0.91	0.91
23 node mobile test (0.1x load)	1.0	1.0	1.0	1.0
23 node mobile test (0.5x load)	1.0	1.0	1.0	1.0
23 node mobile test (0.75x load)	0.27	0.81	0.92	0.36
23 node mobile test (1x load)	0.21	0.23	0.28	0.22
23 node mobile test (1.2x load)	0	0	0	0

and lower traffic loads, and found that when the load (1x = 24 packets per second) was reduced to 0.75x we found very large gains for SNAP-LBC and SNAP-LLBC, while SNAP-Degree and the defaults did relatively worse. When we increased the load to 1.2x, all protocols tested gave zero utility results.At the lowest loads, all protocols did equally well.

Our analysis of the individual experiment logs indicated that SNAP and BMF were initially selecting the same MPRs for forwarding multicast traffic. Upon further analysis and at higher loads, we found SNAP-LBC and SNAP-LLBC did better because they were more selective in their choice of MPRs and thus flooded less traffic than the other strategies that used more MPRs and sent traffic more often. In a busy saturated network, sending too much traffic leads to more collisions and that is where SNAP showed the most gains. We are uncertain if the heuristic used by SNAP-LBC or SNAP-LLBC was leading to an optimal MPR coverage (in our tested scenarios) but the results do provide ample evidence for our first hypothesis that bridging nodes have the right characteristics to be useful as MPRs for multicast traffic.

These results collectively helps better understand the benefits of choosing LBC and LLBC over other potential SNA centrality metrics for this application. We did not do studies with eigenvector centrality or sociocentric betweenness centrality measures because these two metrics can only be centrally calculated at a much higher computational expense and require complete topology information, whereas we require localized computations in a distributed environment for better efficiency.

We need to explore more topologies (real and simulated) and other alternative MPR selection strategies, before we can conclude whether the use of LBC or LLBC is always preferable to the default multicast strategy used by OLSR and to identify any potential drawbacks, but our initial results with SNAP using the LBC and LLBC metrics look very promising.

7.5 Conclusion

In this chapter, we demonstrated the use of novel social network metrics to solve the problem of identifying important nodes in wireless mesh networks for system administrators. The same tools can be used for other applications in any complex network represented as a graph. We introduced 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 indicated their 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 MPRs for multicast in OLSR when using the LLBC metric is beneficial in certain topologies and levels of traffic load. 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 to validate our results. 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 chapter, 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.

Acknowledgements We thank Charles Tao at BAE Systems for his help in developing our OLSR plugin. We thank our families for their love, support, and patience. 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 or reflect the views of any of the sponsors, the US Government or any of its agencies.

References

1. Begnum, K., Burgess, M.: Principle Components and Importance Ranking of Distributed Anomalies. Machine Learning **58**(2), 217–230 (2005)
2. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology **2**(1), 113–120 (1972)
3. Bonacich, P.: Power and Centrality: A Family of Measures. The American Journal of Sociology **92**(5), 1170–1182 (1987)
4. Bonacich, P., Lloyd, P.: Eigenvector-like measures of centrality for asymmetric relations. Social Networks **23**(3), 191–201 (2001)
5. Borgatti, S., Everett, M., Freeman, L.: UCINET for Windows: Software for Social Network Analysis. Harvard: Analytic Technologies (2002)
6. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology **25**(2), 163–177 (2001)
7. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30**(1-7), 107–117 (1998)
8. CenGen Inc.: Extendable MobileAd-hoc Network Emulator. http://labs.cengen.com/emane/

9. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental) (2003). URL http://www.ietf.org/rfc/rfc3626.txt
10. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, second edn. Cambridge, Mass.: MIT Press (2001)
11. Daly, E.M., Haahr, M.: 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), pp. 32–40. ACM Press, Montreal, Quebec, Canada (2007). DOI 10.1145/1288107.1288113
12. Everett, M., Borgatti, S.: Ego network betweenness. Social Networks **27**(1), 31–38 (2005)
13. Freeman, L.: A Set of Measures of Centrality Based on Betweenness. Sociometry **40**(1), 35–41 (1977)
14. Gao, W., Li, Q., Zhao, B., Cao, G.: 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), pp. 299–308 (2009)
15. Hwang, W., Cho, Y., Zhang, A., Ramanathan, M.: Bridging Centrality: Identifying Bridging Nodes in Scale-free Networks. Tech. Rep. 2006-05, Department of Computer Science and Engineering, University at Buffalo (2006)
16. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized link state routing protocol for ad hoc networks. In: Proceedings of the IEEE International Multi Topic Conference on Technology for the 21st Century (INMIC), pp. 62–68 (2001). DOI 10.1109/INMIC.2001.995315
17. Marsden, P.: Egocentric and sociocentric measures of network centrality. Social Networks **24**(4), 407–422 (2002)
18. Nanda, S.: Mesh-mon: A monitoring and management system for wireless mesh networks. Ph.D. thesis, Dartmouth College, Hanover, NH (2008)
19. Nanda, S., Kotz, D.: Localized bridging centrality for distributed network analysis. In: Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN), pp. 1–6 (2008)
20. Nanda, S., Kotz, D.: Mesh-Mon: A multi-radio mesh monitoring and management system. Computer Communications **31**(8), 1588–1601 (2008)
21. Nanda, S., Kotz, D.: Social Network Analysis Plugin (SNAP) for Wireless Mesh Networks. In: Proceedings of the 11th IEEE Wireless Communications and Networking Conference (WCNC) (2011)
22. Naval Research Laboratory: Multi-Generator (MGEN). http://cs.itd.nrl.navy.mil/work/mgen/
23. Naval Research Laboratory: NACK-Oriented Reliable Multicast (NORM). http://cs.itd.nrl.navy.mil/work/norm/
24. Tarjan, R.: Depth-First Search and Linear Graph Algorithms. SIAM Journal on Computing **1**, 146 (1972)
25. Tønnesen, A.: Implementing and extending the Optimized Link State Routing protocol. Master's thesis, Master's thesis, University of Oslo, Norway (2004)