Circuit-Elimination based Link-State Routing in Mobile Ad-hoc Networks

Liu, Yaoda; Schwefel, Hans-Peter; Li, Frank; Nickelsen, Anders

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Circuit-Elimination based Link-State Routing in Mobile Ad-hoc Networks

Yaoda Liu†, Hans-Peter Schwefel†, Frank Y. Li‡, and Anders Nickelsen †
†Center for Teleinfrastruktur, Aalborg University, Denmark
‡Dept. of Information and Communication Technology, University of Agder, Norway
†{yl, hps, an}@es.aau.dk, ‡{frank.li}@hia.no

Abstract— **Circuit-elimination based connected dominating set formation is an efficient technique for reducing routing overhead in mobile ad hoc networks. In this paper, we propose a new message dissemination algorithm which utilizes such techniques to reduce the number of nodes that generate or forward link state advertisements in link state routing protocols. Simulation results with both static and dynamic network topologies demonstrate the potential of the proposed algorithm to reduce routing overhead, compared with a benchmark link state routing protocol, OLSR.**

I. Introduction

Routing in wireless multi-hop networks has attracted massive effort, due to its necessity and challenges, which include topology dynamics and bandwidth limitations. Distance Vector and Link State (LS) routing are two main categories appearing in the literature. In this paper, we delimit our focus to link state routing protocols as they in general adapt faster to topology changes than distance vector routing protocols. Proactive LS routing protocols typically disseminate topology information periodically or in an event-driven manner. In either case, a large amount of routing overhead traffic may be flooded through the network. Hence, it is a critical task for LS routing to reduce this routing overhead.

For routing in wireless multi-hop networks, the basic requirement is the capability to handle network dynamics. Both mobility and wireless channel characteristics introduce dynamics, which may affect the throughput, reliability, and hence the cost of the link. Mobility introduces even more dramatic dynamics, i.e. causing links to disappear or newly appear. Delayed discovery of link state changes may cause route breakage or inefficient route selection, and hence delays should be minimized. On the other hand, the reduction of those delays may lead to increased routing overhead, which needs to be avoided in bandwidth limited wireless scenarios. Therefore the optimization goal is to reduce the routing overhead as much as possible while maintaining a highest possible level of routing performance.

Techniques like Multi-point Relay (MPR) [1] and Connected Dominating Set (CDS) formation [2] have been proposed to reduce the number of generated messages [1] and the number of forwarding events per message [1], [2], [3]. In this study, by utilizing the circuit-elimination based connected dominating set formation algorithm [3], we propose a new link state routing protocol, which encompasses a few variants for message generating and forwarding, in order to efficiently reduce flooding overhead. The proposed protocol applies to both static and dynamic ad-hoc networks.

The rest of the paper is organized as follows. We first provide some background information about link state routing in general and in wireless multi-hop networks in particular and discuss the potential impact of existing optimization techniques on link state routing in Sec. II. The proposed circuit elimination based link state routing protocol is introduced in Sec. III, followed by the detailed simulation-based performance evaluation in Sec. IV. Section V concludes the paper.

II. LS Routing in Wireless Multi-hop Networks

In this section, we first discuss the principle of link state routing in general and the challenges for link state routing in wireless multi-hop networks. A survey of techniques used in link state routing in wireless multi-hop networks and a qualitative analysis of them are provided at the end.

A. *Principle of Link State Routing*

In link state routing, a router maintains a list of links and their associated link costs for the whole network. This list acts as input to a route calculation algorithm, such as Dijkstra's algorithm. In practice, this link list is obtained and maintained in two steps: First, a router should discover the neighboring routers to which it is directly connected, and should determine the costs of those links. Secondly, each router broadcasts the identities (two ends of the link) and the costs of its adjacent links to all other routers. Broadcasting the adjacent-link list, called link-state Update (LSU) [4], by all routers provides each router with the complete network topology graph.

Based on the network topology graph, two basic types of routing mechanisms can be deployed: hop-by-hop routing and end-to-end routing (also known as source routing). With source routing, the source specifies in the packet header a complete or partial sequence of routers through which this packet is going to transfer, and an intermediate router forwards a received message according to this sequence of routers. With hop-by-hop routing, a router forwards a received message according to its local route table, which is of course calculated based on its view of link states. In practice, hop-by-hop routing is preferred in many scenarios due to the following advantages:

- No extra header is required to specify the route calculated at the source,
- The route table is much simpler: for each destination, only one entry containing the next hop router is required. While source routing requires a router to store a sequence of routers for each destination.

With dynamic scenarios, a router may have obsolete link lists and routers may have different views on link states. This information mismatch can lead to the following problems in hop-by-hop routing:

1) Identifying a non-existing route: this is mainly caused due to the delay in discovery of disappearing adjacent links, which are assumed to lead to the next hop according to the router's view on the link list.
2) Failure in finding the best possible route: Two possible consequence can result in that case; finding a suboptimal route if more than one route exists, or being unable to find the only possible route.
3) Routing a packet in a loop: as the routing decision is made in a distributed manner, this problem can occur due to possibly mismatching topology views at different nodes.

B. Overview of Optimization Techniques

The overall routing overhead is affected by three factors: the number of initiated Link State Updates (LSUs), the number of relay events per LSU, and the size of the LSUs. We can modify a link state routing protocol in both the space and the time domains. In the space domain, since not all links are necessary for routing purposes, although the complete knowledge would potentially bring better routing performance, a subset of nodes can be selected to *generate* broadcast messages throughout the whole network. Another possible direction in the space domain is to minimize the number of routers that are responsible for *relaying* these link state broadcast messages. Meanwhile in the time domain, intervals between HELLO messages or link state broadcasts can be increased to reduce the routing overhead.

In this paper, we focus on the space domain optimization. The most efficient solution for finding a link set sufficient for routing is to utilize a spanning tree and only advertise the links in the spanning tree. With this solution, the minimum number of links to be advertised is achieved, which is equal to $n - 1$, where n is the number of routers. Moreover, given a spanning tree, only non-leaf nodes need to generate and forward link state broadcast messages. However, constructing a spanning tree is challenging given the dynamic topology and bandwidth scarcity features in mobile ad hoc networks.

In practice, two solutions are commonly used in this area, CDS formation and MPR selection. The common feature of these techniques is that a router, based on its 2-hop local topology graph, determines the responsibility of itself or its neighbors. Given certain responsibilities, a router conducts either link state broadcast or link state broadcast relay, or both, and the overall advertised link set may also be reduced. Specifically, given a CDS, only routers in the connected dominating set may generate and relay link state broadcast messages, and links between non-CDS nodes may not be advertised at all. With MPR, only routers that have been selected as multi-point relays by at least one neighboring router may generate link state broadcasts, and a router only relays a link state broadcast received from a neighboring router that has selected itself as an MPR; furthermore, a link state broadcast may contain only MPR links.

C. Impact of Optimization Techniques on Routing

The utilization of CDS and MPR diminishes the routing overhead caused by link state broadcast transmissions through a reduced set of nodes for link state advertisement, and a reduced number of transmission events for any single link state broadcast. However, it is important to note that wireless channels are error-prone, hence a certain level of redundancy is necessary and beneficial. This implies that the utilization of CDS and MPR may have negative impact on routing performance. Here we provide a qualitative analysis of their impact on the routing performance as measured by end-to-end metrics such as path length, packet delivery ratio, delay, and jitter.

The utilization of CDS or MPR in link state routing reduces the redundancy in the sense that 1) the number of link state broadcast transmissions is reduced, 2) the number of possible receptions of the same link state broadcast may also be reduced, 3) the number of links advertised, which can be potentially used by the routing mechanism, may be reduced. These effects will degrade the resilience against channel errors, which can cause all three problems discussed in Sec. II-A. Relating to the routing performance, these problems may imply longer paths, lower packet delivery ratio, and longer delay.

Note that the utilization of CDS formation does not introduce extra delay into the link state routing procedure, if the algorithm is localized, i.e. a router determines its own responsibility purely based on its local topology graph. One the other hand, the utilization of multi-point relaying may introduce extra delay into the link state routing procedure as, decisions by neighboring nodes need to be communicated and this notification introduces extra delay, although in most cases rather small.

On the other hand, the reduction of routing overhead by the utilization of CDS and MPR can potentially improve the routing performance. Keep in mind that wireless links in wireless multi-hop networks are resource constrained. The reduction of routing overhead can actually reduce the network load and leave more resources to the data traffic, which may improve the end-to-end performance.

III. CIRCUIT ELIMINATION BASED LINK STATE ROUTING

In this section, we first introduce the circuit elimination based connected dominating set formation, and then describe the Circuit Elimination based Link State Routing (CELSR) protocol.

A. Circuit Elimination based CDS formation

The algorithm proposed in [3] is adopted as the method for connected dominating set formation. In this method, each node maintains the following data structure:

- The *local topology graph* of a node is the subgraph that contains all the nodes in its 2-hop neighborhood, and all links incident on nodes in its 1-hop neighborhood.
- The *reduced local topology graph* is the resulting subgraph of the local topology graph after the execution of the circuit elimination algorithm.

To decide whether a node is a CDS node or not, the node first removes one edge from its reduced local topology graph for each circuit according to a specific selection criterion, for instance highest node id or lowest node degree. The node then determines its CDS status by checking whether its degree in the reduced local topology graph is larger than one or not. In other words, *a node identifies itself to belong to the CDS, if its degree is larger than one in its reduced local topology graph*.

B. Introduction of CELSR

The proposed CELSR protocol consists of the following three components: 2-hop neighbor discovery, topology discovery, and routing mechanisms.

In CELSR, the *2-hop neighbor discovery* procedure is done by the exchange of HELLO messages containing all adjacent links of the sending node. Such messages are periodically broadcast by each node to its 1-hop neighbors. With this approach, a node can obtain the knowledge of its 2-hop neighborhood.

Topology discovery is the key component of CELSR, which makes it different from other existing link state routing protocols. A CDS constructed by the circuit elimination algorithm can be utilized in three ways. First, only CDS nodes can be used in determining the nodes to *initiate* LSU messages containing all adjacent links of the generating node. Since all nodes with CDS status form a virtual backbone with all other nodes being adjacent, the union of adjacent links of all CDS nodes, called advertised link set, is sufficient to derive routes for any source-destination pair. Second, only CDS nodes can be used to determine the nodes to *forward* LSU messages as in [3]. If all CDS nodes forward all LSU messages, the messages can reach all nodes in the network, and consequently all nodes can collect the complete advertised link set. Third, the advertised link list can be reduced further, i.e. only those links adjacent to the generating node in its *reduced local topology graph* are attached in the generated LSU messages, which also provide a link set sufficient for routing. Among those three options, the first two can be utilized independently, while the third one can only be applied when used together with the first option. The first two options are expected to reduce the number of LSU transmission events, while the last option can reduce the size of generated LSU messages.

If the first option but not the third option is applied, we have two design alternatives for the *routing algorithm* to make, i.e. either hierarchical or flat routing. With hierarchical routing, non-CDS nodes will choose a CDS node, e.g. the only neighbor in its reduced local topology graph, as the relay point for all its data traffic and CDS nodes forward message according to the routing table. However, with flat routing, CDS and non-CDS nodes act in the same way, i.e. calculate the next hop for all destinations and route messages accordingly. The difference between these two approaches can be illustrated in Fig. 1 (for simplicity, all adjacent links are attached in LSU messages in this example although a similar example can be easily shown for the cases when the third option is also applied). With flat routing, all links except those between non-CDS nodes are utilized in the calculation of the routing table. With hierarchical routing, among all adjacent links of a non-CDS node, only one, i.e. the one in the reduced local topology graph (together with all links between CDS nodes), is utilized. Clearly the flat routing approach leads to shorter path from non-CDS nodes to other nodes than the hierarchical routing approach (not for path from CDS-nodes to CDS-nodes) as shown in Fig. 1. Hence we focus on the flat routing approach in the evaluation section. However, we are aware that the hierarchical routing may be preferable in heterogeneous networks where devices differ in computational capability, memory space, or battery power.

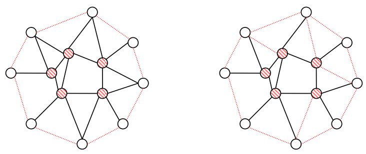

Fig. 1. In this figure, CDS nodes are shown as circles with red line pattern. The left-hand and right-hand figures show the link set (shown as solid black line) used in the routing table calculation in flat and hierarchical routing, respectively.

If both the first and the third options are applied, the routing mechanism is always hierarchical as the advertised link set contains only one link adjacent to any non-CDS nodes.

IV. Performance Evaluation

Simulation based evaluation is conducted in this section. We first investigate the routing overhead caused by two variants of CELSR and Optimized Link State Routing (OLSR, as the benchmark protocol) using Matlab. The choice of OLSR as the benchmark protocol is made based on its popularity and acceptance in the MANET and mesh networking communities. Detailed network simulations are conducted using ns2 to evaluate the routing performance.

A. Routing Overhead of CELSR

During this set of simulations in Matlab, we simulate 200 static networks of 50 nodes with various average node degrees at 4, 14, 24, 34, 44 respectively, modeled by unit-disk graphs. Two variants of CELSR, depending on the criterium used in the circuit elimination, are considered. That is, CELSR-id removes edges from end-nodes with lowest IDs, and CELSR-dgr removes edges from end-nodes with lowest degrees. For CELSR, the number of generating nodes [1] and forwarding nodes in a single simulation run (corresponding to one specific static networks topology) are equal to the size of the CDS resulting from the circuit elimination algorithm. The number of adjacent links of the CDS nodes within their reduced local topology graph is averaged over the set of CDS nodes to obtain the average LSU size.

The overall routing overhead is then defined as the average number of LSU link-layer transmissions conducted by all nodes per LSU update period. It can calculated as the product of the number of link state broadcast initiated by all nodes and the average number of transmission events for a link state broadcast. Another important metric is the average number of links advertised in a link state updates.

For OLSR, the number of LSU generating nodes for a single simulation run is equal to the number of nodes that have been selected as MPRs by at least one neighbor. The number of forwarding nodes and the number of links per LSU message are

[1] The number of generating nodes is linearly related to the number of LSU messages generated given a a constant LSU generation period.

obtained from the simulation. The obtained average values for the first two performance metrics, as well as their product as a measure of the overall routing overhead, over 200 simulation runs are depicted in Fig. 2. It is clearly shown that CELSR-dgr can reduce the number of generating nodes significantly compared to OLSR, and CELSR-id can also do so with relatively high node degree, e.g., more than 10 in the figure. However, the average number of forwarding nodes by CELSR-id is larger than that achieved by OLSR, while OLSR and CELSR-dgr achieve comparable results.

Concerning the number of links in a LSU message as shown in Fig. 3, CELSR in most cases attaches more links than OLSR, especially in the cases of high node degrees. In the given scenario, the average number of links in a LSU message generated by CELSR are comparable to those of OLSR when the node degree is smaller than 15. With the increase of average node degree, OLSR leads to much smaller LSU messages, especially when compared to CELSR-dgr. The reason is that with CELSR-dgr, a node with higher degree is more likely to become a CDS node, consequently a generating node.

This set of simulations clearly shows the advantage of CELSR-dgr, i.e. resulting in significant reduction of transmissions of LSU messages. CELSR-id can also achieve this improvement in networks with relatively large degrees. However, CELSR-dgr leads to larger LSU messages than those of OLSR, as the number of links described in the message is much larger, especially in dense networks. Hence, in relatively sparse networks, CELSR-dgr is recommended, while in relatively dense networks, the choice may depends on the characteristics of the Layer-2 technology, specifically on how packet sizes influence frame numbers and frame transmission properties. For example, in networks with high bit error rates, short packets may have higher probability of successful reception, in which case OLSR may be preferred. Otherwise, CELSR-dgr can still be preferred.

Although CELSR-dgr outperforms CELSR-id in most cases, we consider CELSR-id in the network simulations presented below for two reasons: First, CELSR-dgr requires each node to have the degree information of 2-hop neighbors. Hence, each node has to attach its degree in the HELLO message. Collecting the degree info for the complete two-hop neighborhood takes one more HELLO interval than for the collecting of link states only. Second, the implementation complexity of CELSR-id is lower than that of CELSR-dgr.

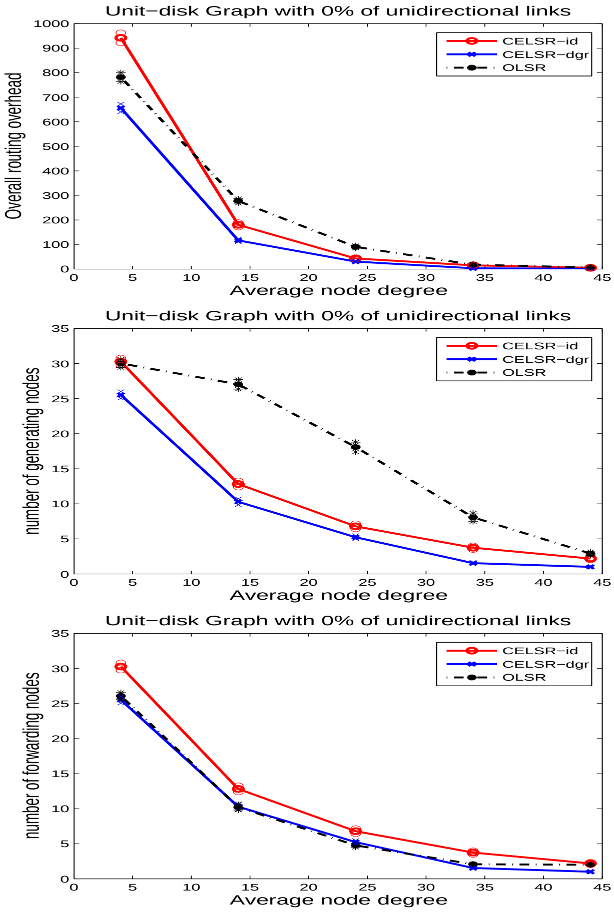

Fig. 2. The number of transmissions caused by link state updates: The average number of nodes generating LSU messages and the average number of nodes forwarding a LSU message, and the overall number of transmission.

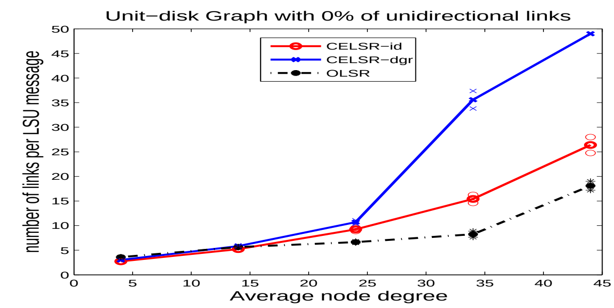

Fig. 3. Average number of links in a link state update

B. Routing Performance

In the above evaluations, we have shown that the introduction of circuit elimination based on connected dominating set formation can reduce the routing overhead significantly in static networks compared to OLSR. Hereafter, to demonstrate the applicability of CELSR in dynamic scenarios, we conduct network simulations in highly dynamic networks, i.e. nodes move with relatively high speed. The following four variants of the CELSR protocol were implemented in NS2 based on the NRLOLSR implementation [5]:

1) CELSR: Only CDS nodes generate and relay LSU messages.
2) CEMPRLSR: CDS and MPR are used to decide which nodes to generate and relay LSU messages, respectively.
3) MPRCELSR: MPR and CDS are used to decide which nodes to generate and relay LSU messages, respectively.
4) OLSR: This is the original OLSR. MPR is used to decide which nodes to generate and relay LSU messages.

We simulate extensively the CELSR routing protocol with the above four enhanced LS dissemination variants, in order to compare CELSR with OLSR in dynamic networks. In a geographical area of 900m $\times$ 900m, nodes move with a maximum speed of 10 m/s during the simulation time of 250 seconds according to the random waypoint mobility model. With the above mobility pattern, 15 mobility traces are generated for

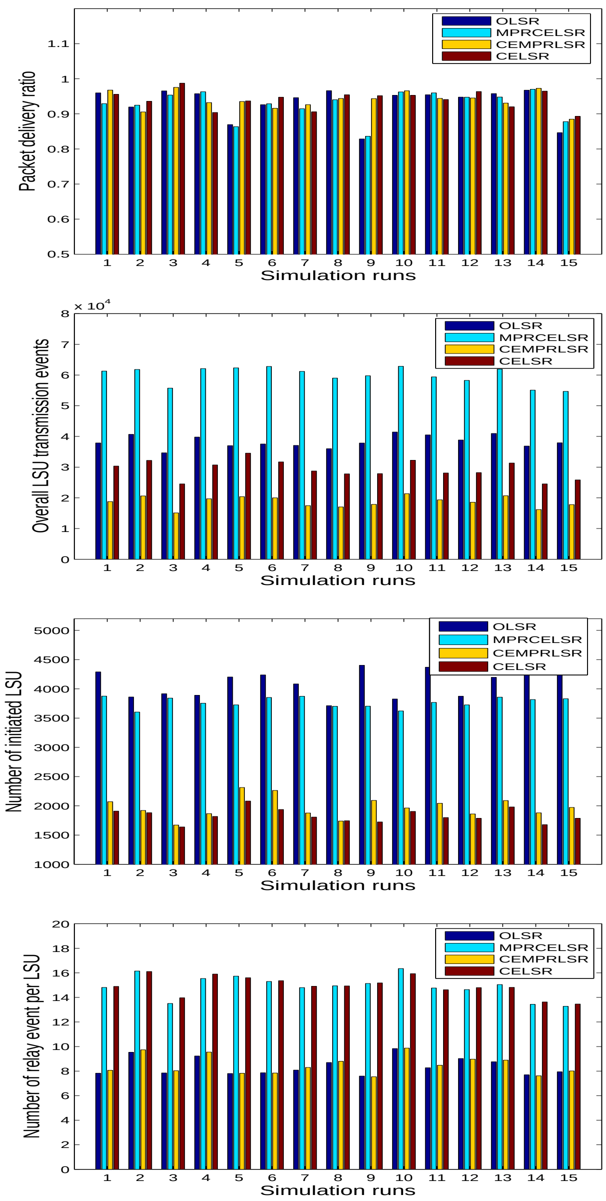

Fig. 4. Routing performance measured by packet delivery ratio, and routing overhead caused by LSU transmissions in networks of 50 nodes in a geographical area of 900 meter by 900 meters; simulation duration is 250 seconds for each of the 15 independent replications.

simulations of all protocols. The same traffic trace, in which one source-destination pair is sending UDP data traffic, is used for all simulation runs. IEEE 802.11 MAC is used as the underline MAC protocol. We measure the packet delivery ratio, the overall number of LSU message transmissions, the number of initiated LSU messages, and the average number of relay events per LSU message, as plotted in Fig. 4. Comparable performance in packet delivery ratios, around 90%, are observed for all four variants. First, this shows the applicability of the circuit elimination algorithm in link state routing for dynamic networks, i.e. the algorithm is capable of reacting to topology changes in dynamic networks. Second, this also proves that the nodes to generate and relay LSU messages can be decided by different techniques, as the two variants using CDS and MPR to decide which node to generate and which node to relay respectively achieve similar packet delivery ratio as OLSR. Third, as expected, the delay introduced by MPR notification does not lead to significant performance degradation. Concerning routing overhead, the two variants using CDS to determine the nodes generating LSU messages introduce much less LSU messages than the other two using MPR, which is similar to those results in Fig. 2. They however lead to more relay events per LSU message than the two using MPR to determine whether to relay a message. Furthermore, CEMPRLSR has introduced least routing overhead in this set of simulations.

In summary, all the investigated variants work in dynamic networks; MPR and CDS formation can be used to decide generating or relaying node either independently or jointly; Although CDS (id based) causes more relaying events per LSU than MPR, it introduces less routing overhead due to the significant reduction of generated link state update messages.

V. Concluding Remarks

In this paper, we have proposed to use circuit elimination based CDS formation techniques for link state routing. The simulation results show the applicability of a set of variants of the proposed CELSR protocol in both static and dynamic multi-hop networks. Another contribution of this work is that we demonstrate the feasibility to use both CDS and MPR, either individually or jointly for deciding generating and forwarding nodes for link state broadcasts. While CDS leads to a lower number of generating nodes in general, the number of forwarding nodes may also be reduced using certain variants of the circuit elimination algorithm. As a result, a carefully designed combination of these techniques can reduce the routing overhead caused by link state broadcast significantly.

Acknowledgment

This work has been supported by the European Commission under the framework of the IST-HIDENETS (http://www.hidenets.aau.dk/) project and the IST-ADHOCSYS (www.adhocsys.org/) project.

References

[1] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", IETF RFC 3626, Oct. 2003.

[2] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *In Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Aug. 7-14 1999.

[3] Y. Liu, H.-P. Schwefel, "Algorithms for Efficient Broadcasting in Wireless Multi-hop Networks", *In Proc. of IEEE Globecom 2006.*

[4] R. Perlman, Interconnections: Bridges, Routers, Switches, and Internetworking protocols, 2nd Edition, Addison-Wesley Professional Computing Series, Reading, MA, 1999.

[5] The Navy Research Lab Implementation of OLSR for NS2, http://cs.itd.nrl.navy.mil/work/proteantools/ns2extensions.php.