Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 418–425

Procedia Computer Science

The 2nd International Conference on Ambient Systems, Networks and Technologies

OLSR-based Topology Discovery for Mobile Situational Awareness Systems

Anand Dersingh, Z.M. Faizul Islam, Shahram Shah Heydari, Md. Mustafizur Rahman

University of Ontario Institute of Technology, 2000 Simcoe Street North, Oshawa, Ontario, Canada

Abstract

In this paper, we propose a non-intrusive approach for connectivity visualization of OLSR-based MANET topology based on local topology databases available in an OLSR node. Two different scenarios are considered: a central (full view) topology from a command and control location, or a nodal (partial) view from an ad-hoc node. A simulation based analysis is conducted to calculate total number of active links at a particular time in full and nodal topology views. Also the error rate of network topology discovery based on total undiscovered link both mobile and static scenario is considered and reported in this paper.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.

Keywords: Mobile Ad-hoc Networks, Topology Discovery, OLSR, Situatinal Awareness

1. Introduction

1.1 Background

Mobile Ad-hoc Networks (MANET) play an increasingly important role in various types of ambient networks, sensor networks, vehicular networks, and tactical communication networks (e.g. search and rescue teams). In absence of an underlying routing infrastructure, reliability and performance of such networks varies significantly with time because of the fact that nodes are mobile and the topology of the network is constantly changing. Therefore, having a real-time update of the status of nodes in such networks is extremely important. Network situational awareness systems are used for providing a real-time view of the current status of all network assets for a command and control centre [1]. An important task of such system is to create a network Common Operating Picture (COP) to deliver a view of the status of network components in real time, including parameters such as logical view of interconnectivity and dependencies, logical view of assets and their configuration, geographical information of the assets, and potential threats [1]. This task is particularly difficult in tactical Mobile Ad-hoc Networks (MANETs) because of harsh environment (noise, link instability, and hostile nodes), dynamic topology due to node mobility, limited bandwidth due to wireless communication, limited energy resources for battery powered devices, and limited security against eavesdropping [2]. Most current mobile ad-hoc network visualization

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas. doi:10.1016/j.procs.2011.07.054

tools fall into one of the two categories: position plotting tools (often based on GPS data), or graph visualization tools that display the status of connectivity between nodes based on routing table data [3]. Our focus in this research is on the connectivity visualization based on the topology and routing data available from the routing protocol at each node. Network topology discovery is the main step in connectivity visualization. A network is modeled as a graph of nodes with connections established between pair of nodes. It can be assumed that in a wireless MANET, all authorized nodes will establish connections to all other nodes within the range of their transceivers. However some networks may have security policies that limit such access based on the trust level of the nodes [4]. Furthermore, the transmitting range of a node may not be necessarily the same as its receiving range. Such issues are typically dealt with at the layers below the network layer and as a result their impact is already included in the routing tables, i.e. no direct route will be considered between two nodes that are not authorized to be connected directly.

In developing topology discovery algorithms based on routing information, the following considerations must be taken into account:

- Active vs. Passive: Situational Awareness systems can use an active approach to network discovery, e.g. by sending queries to network nodes; or take a passive approach by merely listening to the messages, in particular routing updates and Link State Advertisements (LSA) that are exchanged between network nodes. In this work we are only interested in passive network discovery.
- Proactive vs. Reactive, Distance Vector vs. Link State Routing Protocols: MANET routing schemes are often categorized as proactive or reactive, depending on whether the routing is performed using a pre-calculated routing table, or on-demand.
- Nodal database vs. Protocol messages: A passive Situational Awareness System has two sources of information to help it builds the network topology and status map: the data available locally at the monitoring node (e.g. routing database, neighbour adjacency information etc.), and the link state updates. In the former case the topology discovery algorithm relies on the information gathered by the routing protocol to build the full status map and is thus limited to the extent of databases maintained by each particular routing protocol, while in the latter the algorithm replicates some of the workings of the routing protocol to build the topology map from the link state messages. Depending on the routing protocol, some status messages may not affect the routing databases while could be useful for topology mapping.
- Centralized vs. Nodal Visualization: By nodal visualization we refer to the viewpoint of an individual node of the entire network, based on the routing messages received by this node. Depending on the routing protocol, this view could be partial or complete at different time instances. In centralized visualization we assume that the monitoring node has access to all messages in the network and/or all nodal routing and topology databases. Obviously a centralized network discovery converges faster that nodal network discovery; however it may require active participation, i.e. sending queries to the nodes to collect status information.

1.2 Related works

Koyama et. al. [8] developed a visualization system called MANET-Viewer in order to observe ad hoc network behaviour. The system is a centralized approach by which there is a dedicated node (Personal Computer with wireless cards) for data collection. The process of collecting data is performed by using flooding model used by a data collection application on the dedicated node. The authors claim that the performance of this tool depends on node information gathering which varies depends on the stage of the network. Experimental results show that the node' information collection time increases as a number of hops increases. However, there is no experiment regarding a number of nodes in the network. Also, this tool does not provide any form of location information except that a GPS coordinate can be entered manually.

Visualization of MANET simulations based on NS or OPNET has also been studied in the literature. Typically NS-2 executes a network scenario and produces trace files containing time stamps of network events. Nam [9] and iNSpect [10] are network visualizers that playback NS-2 trace files. Nam was built for wired network visualization rather than wireless network, and has some limitations in drawing wireless traffic. The iNSpect program was built to visualize wireless network events such as packet hops, wireless node links, and packet delivery success. iNSpect can only visualize wireless traffic. Belue et. al. [11] designed and developed an enhanced framework for visualizing military and non-military wired and wireless networks. The main idea of this framework is to animate events contained in a simulation trace file. The acceptable trace files are NS-2 and OPNET. GTNets [12] is another network simulator that combines a full-featured network simulation environment with graphical viewing of the simulation. It shows the network topology along with network traffic. However, the visualizer component of GTNets is tightly coupled to the simulator and cannot be separated nor it can work with other simulators' outputs.

MLab is a MANET test bed from National Institute of Standards and Technology (NIST) that addresses the balance between desktop simulations and outdoor field tests [13]. The motivation of the mLab came from the fact that simulation results may not accurately reflect in a real-world scenario, hence, a lab environment is necessary but it comes with expenses of actual hardware. mLab allows users to develop and test their MANET protocols and applications in a laboratory environment and takes advantage of simulated systems to bring the setup environment closer to an actual deployment. To use mLab, it requires a set of PCs or laptops by which each of them must have an Ethernet adapter and a wireless adapter. [14] is an extension of mLab to include a collection of host-based kernel network traffic monitoring modules. The kernel modules (Linux) monitor the network traffic traversing the host's wireless adapter and write a number of network traffic metrics into the "/proc" virtual directory which in turns provides a system status. Hence, this extension helps in collecting more traffic information as close to the real traffic in an actual deployment of a node.

The impact of a partial view of the network on data delivery in an OLSR network has been studied in [7]. This work was primarily focused on increasing data delivery rate by tuning various OLSR parameters (which we will discuss in Section 2) and assessed the increase in control overhead.

In this work we present the framework for mobile ad-hoc network status and topology visualization based only on standard OLSR protocol databases, and without any modification to standard OLSR operation [5]. In contrast to the work in [8], we develop an approach which can be used either in a centralized command and control environment based on node queries, or in a non-intrusive manner based on the routing and topology information available at each node (which we call the nodal view). We show the efficiency and accuracy of using OLSR databases to build a map of the network, and use simulations to study the performance of centralized versus nodal visualization. In particular, we present results to show how quickly and with what degree of accuracy a nodal view of the network can approach the centralized full view.

The rest of this paper is organized in the following manner: In Section 2 we provide necessary background information on topology information in the OLSR protocol. In Section 3 the framework of our approach is presented. Simulation results and analysis are presented in Section 4, and conclusions and future objectives are presented in Section 5.

2. OLSR Topology Information

Optimized Link State Routing Protocol (OLSR) [5] is a routing protocol developed for mobile ad hoc networks (MANETs). It is a table driven and proactive protocol. OLSR is considered optimized since it minimizes the overhead (number of messages) in discovering nodes. OLSR achieves its optimization by using the concept of Multipoint Relay (MPR) nodes. Conceptually OLSR topology discovery involves two phases: neighbourhood discovery and topology flooding.

The concept of two-phase topology discovery is similar to OSPF [6] but the operations are not the same. In the first phase, neighbour nodes are discovered by using Hello messages. The exchange of Hello messages in OLSR also involves electing MPR nodes. MPR nodes are nodes responsible for broadcasting topology control (TC) messages (topology information) which would be flooded throughout the network in the second phase. With this approach, only a subset of nodes in the network will be used to generate broadcast topology messages, in contrast to OSPF where all nodes are responsible for forwarding topology information.

2.1. MPR Selection in OLSR

Each node calculates its own set of MPRs so that it can reach all its symmetric 2-hop neighbours via one of its MPRs. MPR calculation is based on willingness announced by neighbours using Hello messages. Willingness is one of the fields in a Hello message, which specifies the willingness of a node to carry and forward traffic on behalf of other nodes. According to the standard OLSR [5], willingness may be set to integer value between 0 and 7. The willingness value of WILL_NEVER (integer value of 0) means a node does not wish to carry traffic for other nodes and it will not be included in the MPR set. The willingness value of WILL_ALWAYS (integer value of 7) means a node is willing to or has resources to forward traffic for other nodes. Therefore, for a given node all the neighbour nodes with willingness equal to WILL_ALWAYS will always be included in the set of MPRs [5]. This is the beginning of the process and the node keeps adding its neighbours to the MPR set based on the reach ability to the 2-hop nodes which is based on link set, neighbour set, and 2-hop neighbour set until all the 2-hop neighbours are covered by at least one of the MPRs. The detail of the algorithm is described in section 8.3 of the RFC3626 [5].

2.2. Steps for Topology Discovery in OLSR

The results of exchanging Hello message are records of nodes 2-hop neighbourhood information, the MPR nodes and the MPR selectors. One may view that, for a given node, exchanging Hello messages of OLSR reveals network topology of the area the node belongs within 2-hop radius. The next step in discovering topology is flooding TC messages. Only MPR nodes are allowed to generate and forward TC messages. The information embedded in TC messages generated by an MPR includes at least the existing links between itself and its MPR selectors. The non-MPR nodes do receive TC messages from their MPRs and process them. However, non-MPR nodes do not forward the received TC messages. This feature of OLSR reduces the number of messages exchanged in topology discovery.

However, according to the standard OLSR specifications [5], OLSR does not use any form of reliable mechanism in guaranteeing the reception of the topology information in contrast to OSPF that requires nodes to acknowledge when they receive topology information. Unreliable message transport in conjunction with nodes' movement and random nature of the wireless signal may cause inaccuracy of topology information. In other words, topology information in a node may not reflect the current actual network topology accurately. Standard OLSR does allow some flexibility in order to make OLSR more redundant by providing two tuneable parameters. Details of these parameters can be found in [5] and [7].

3. **Proposed Framework**

Two different approaches are possible here: centralized visualization, in which a command and centre node collects OLSR topology information from all nodes and builds the network topology based on that information, and nodal visualization in which each node builds its own view of the network according to its own OLSR information. In former case, either the command and control centre itself is part of the OLSR network and thus receives all TC messages; or it has to exchange messages with the individual ad-hoc nodes to retrieve their databases, in which case an additional control traffic overhead exists. In the latter case, the node relies on its local OLSR database and therefore the topology discovery is non-intrusive. However this approach may provide a partial (nodal) view. Both approaches may be employed concurrently in certain scenarios, e.g. a tactical MANET, where network situation awareness is crucial both to the field nodes and to the central command.

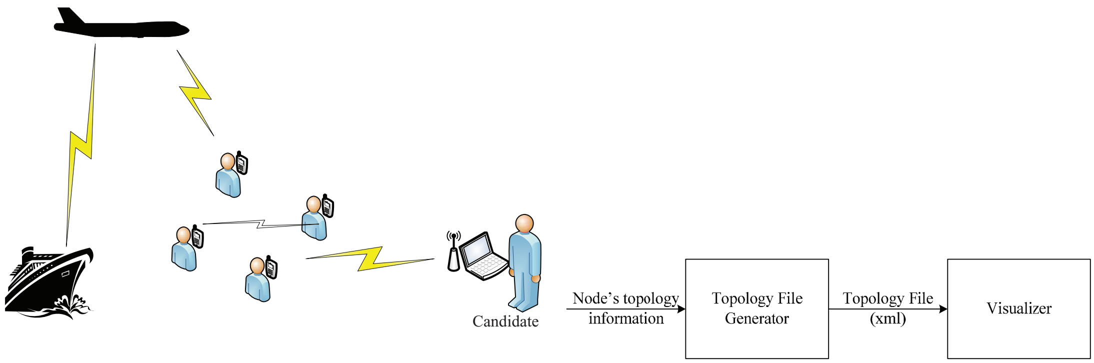

Figure 1: (a) OLSR MANET Scenario; (b) Visualization Processes of the Candidate Node

Figure 1a shows an OLSR MANET scenario by which the candidate node will be any node that performs visualization related tasks (either a nodal view or as a central command node). An application process is required on the candidate node. The purpose of the application is to collect the node's topology information and to display the network topology graphically. There are two modules within this application (as shown in Figure 1b): Topology File Generator and Visualizer. The purpose of the topology file generator is to generate a topology file and to update it in real-time.

There are two possible approaches in implementing topology file generator. One approach will only use the OLSR databases. In the nodal view approach, the generator module has access to the neighbourhood database and topology database (routing layer) of the candidate node. The neighbourhood database contains a set of tuples (neighbour_main_addr, 2_hop_addr, and time stamp) that contains address of a node's first and second-hop neighbour nodes. The topology database contains a set of the last hop before each destination nodes in the network. The information read from these two databases reveals the nodal view of the network topology. It is obvious that if the eccentricity of a node is less than four hops, the OLSR neighbourhood and topology databases at that node provide sufficient information to identify all network nodes. If a node is four hops or more from a destination, then the 1^{st}-hop, 2^{nd}-hop and last hop tuple will not be sufficient for identifying all intermediate nodes to that destination. If a centralized approach is used, i.e. the topology generator has access to databases on all nodes, then a complete view of all nodes can be constructed using the 1^{st}- and 2^{nd}-hop neighbor databases only.

One problem with relying on OLSR topology databases is that only links that are on the shortest path trees will be discovered. In the centralized visualization approach this would not cause a problem because, assuming that in a typical wireless MANET that nodes always connect to their nearest nodes, a link between a node and its neighbour is included in at least two shortest path trees and therefore, if the central visualizer has access to all databases, it can find all links as well. In a nodal visualization approach, however, any link that is not on the shortest path tree of the visualizing node will remain undetected if we only rely on OLSR databases. Furthermore, OLSR TC messages have the option of only reporting the links between an MPR and its selectors, therefore potentially omitting links between MPR selectors, In such case only a centralized approach that uses 1-hop neighbour databases from all nodes can guarantee the full topology map.

The topology information constructed in either approach can be parsed to and stored in an XML file. Note that this XML file will be updated in real-time in order to maintain an up-to-date network view.

4. Simulation Methodology and Results

A simulation environment was set up for implementing MANET using NS-3 which is a discrete-event network simulator. We created different scenarios for network sizes of 4, 6, 10 and 15 nodes, and each scenario was once simulated with 2-D random walk mobility model and then with constant position (non-mobile) for comparison. Also IEEE 802.11a is used as wireless MAC Layer, where each node was capable of establishing ad-hoc connection with all other nodes within its range. As a routing protocol a table driven proactive OLSR routing protocol was used. For the wireless channel we use Yans wifi channel propagation model [15]. In our simulation, constant speed propagation delay model and Friss propagation loss model were used to represent the propagation delay and the loss of signal power in a free space as distance increased.

For modeling 802.11 physical layer we also defined the Tx power = -0.1615 dBm and the data rate was set to 4 kb/s. For random walk mobility model, a closed area of 1000 x 1000 sq. meters was assumed where the nodes moved randomly within the area. In non-mobile case, the nodes were placed on a grid within the same area.

We used NS-3 logging feature [16] to log all OLSR events. A custom script was created and used to parse the log file and re-generate events related to topology database in real time and to send them to a topology calculation module. In order to create the full_view OLSR events for all nodes were processed while creating nodal_view required only events on node under consideration. The script checked the log file for events that contain information about active links between a node and its 1st- and 2nd-hop neighbours, as well as the last hop before a destination.

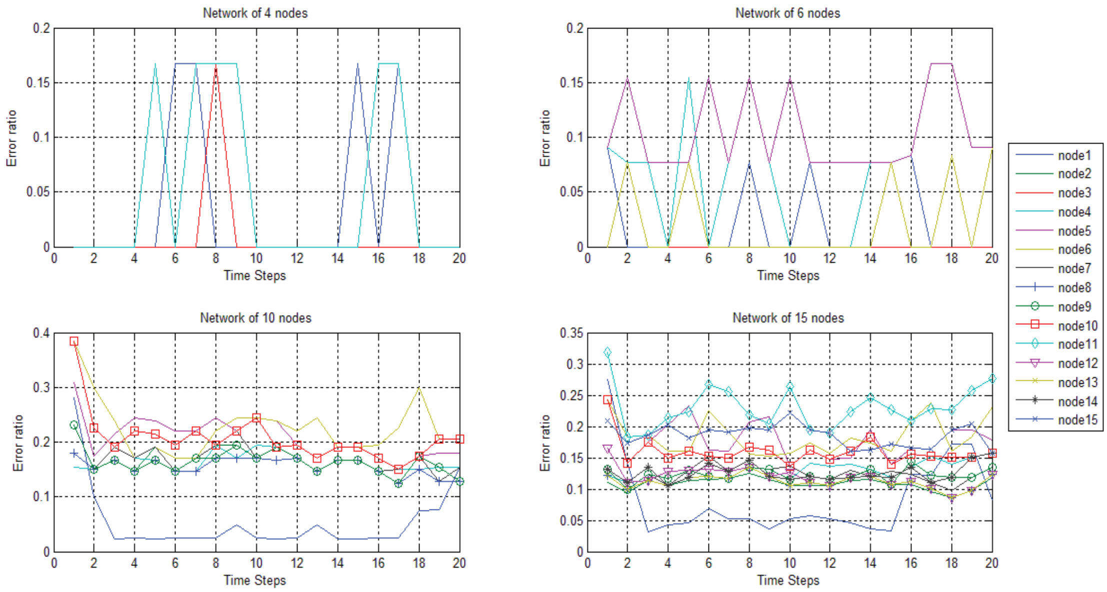

Figure 2: Error rate in nodal view for mobile scenari0

OLSR protocol periodically declares 1-hop, 2-hop and destination node information for each and every node within a network. The total number of active links at an instance of time was calculated by adding up all the links based on this information. We assumed symmetric network links between nodes. We then generated adjacency matrices based on link information for full view topology and each partial view topology. Each element in the adjacency matrix specifies the condition of a link in the network; i.e. if L_{ij}=1, then an active link between nodes N_i and N_j exist and if L_{ij}=0, there is no link between the two nodes (i.e. they are out of each other's range). If all links are bidirectional, the adjacency matrix in this case will be symmetric. The adjacency matrix is updated periodically as OLSR updates its tables.

At each instant once the adjacency matrices were built for each nodal_view (distributed view), and combined to create the full_view (centralized view). we compared each nodal_view adjacency matrix with full_view adjacency matrix to determine the number of undiscovered links and the accuracy of nodal_view topology discovery. In each case we ran the simulation for 100 seconds.

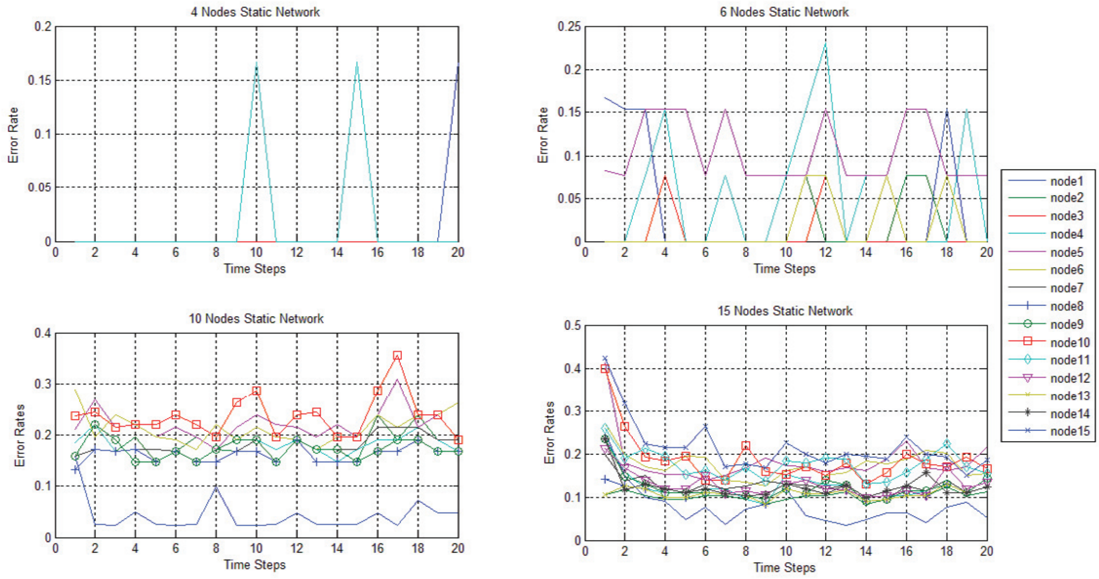

Figure 3: Error rate in nodal view for static scenario

Figure 2 and 3 show the percentage of undiscovered links for different network sizes with and without mobility model respectively. Each time step identifies a visualizer update interval, set to 5 seconds. Each simulation ran for 100 seconds. In both mobile and non-mobile cases the number of undiscovered link in nodal views (in comparison to the full view) is calculated at an instant of time and presented as error rate. We note that in addition to mobility, there is also a possibility of link loss due to lower signal strength resulting from environment noise levels, therefore even in non-mobile scenario the error rate could increase in time. In the mobile scenario, however, mobility is the main factor in the increase in the number of undiscovered links. Being a proactive routing protocol, OLSR uses shortest path algorithms to calculate a route instead of flooding mechanism and it continuously updates route information which result in continuous change of link status among nodes. As a result undiscovered links increase with network size. At the first stage of topology discovery the nodal view is rather incomplete for both static and mobile scenarios. Over time, each node discovers more information about network topology. However full stability is never attained because the topology is constantly changing (either because of mobility or change in signal strength). In our simulation, a random expiration time was generated for each active link and after the expiration of this entry, it sensed the link again to check its status. Error rates in Figures 2 and 3 have been defined as the ratio of total undiscovered links (total active links at full_view topology minus total active links in nodal_view topology) and the total active links at full_view topology. As shown in figure 2 and 3, the error rates in the mobile scenario started around 40% and dropped to the 20%-30% range with occasional spikes. In the static scenario, the error rates quickly stabilized below 20% (relating to the links not on the node's shortest path tree) with occasional increases due to noise and weak signal strength.

5. Conclusion and Future Work

In this research we studied real time MANET topology visualization based on the topology and routing data available from the OLSR routing protocol at each node. We examined different approaches in using OLSR data for topology visualization, and showed the efficiency and accuracy of using OLSR databases and messages to build a topology map of the network. We further used simulations to study the performance of centralized versus nodal visualization and identified factors that contribute to errors in topology discovery.

We plan to continue this research by comparing the overhead and performance of centralized versus distributed OLSR topology visualization, calculating the overhead of using TC, analyzing the impact of mobility parameters and patterns with respect to situational awareness systems, improving the topology discovery algorithm by approximating the node locations based on their connectivity and signal strength, and comparing OLSR with other MANET routing protocols in a situational awareness system.

Acknowledgements

This work was done in collaboration with Solana Networks, Nepean, Canada, and was supported by a grant from the Natural Sciences and Engineering Research Canada (NSERC) through NSERC Engage industrial collaboration program. The authors are thankful to Nabil Seddigh and Biswajit Nandy of Solana Networks, and Mazda Salmanian of Defence R&D Canada, for their helpful comments, guidance and feedback.

References

[1] M. Gregoire and L. Beaudoin, "Visualisation for Network Situational Awareness in Computer Network Defence", In Visualisation and the Common Operational Picture; 2005 (pp. 20-1 – 20-6). Meeting Proceedings RTO-MP-IST-043, Paper 20. Neuilly-sur-Seine, France.

[2] M. Gerla, “From Battlefields to urban grids: New research challenges in ad hoc wireless networks”, Pervasive and Mobile Computing, 1 (2005) 77-93.

[3] B. Q. Nguyen, “A Tool for Visualizing Mobile Ad-Hoc Network Topology Definition Files", Technical report ARL-TR-350, US Army research Laboratory, Adelphi, MD, February 2006

[4] H. Tang and M. Salmanian, “Lightweight Integrated Authentication for Tactical MANETs”, The 9th IEEE International Conference for Young Computer Scientists, 2008, pp. 2266-2271.

[5] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC3626, http://hipercom.inria.fr/olsr/rfc3626.txt

[6] J. Moy, “OSPF version 2”, RFC2328, 1998 http://tools.ietf.org/html/rfc2328

[7] T. H. Clausen, P. Jacquet, and L. Viennot, “Investigating the Impact of Partial Topology in Proactive MANET Routing Protocols”, 5th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2002

[8] Akio Koyama, Kenyu Kamakura, and Leonard Barolli, “MANET-viewer: a visualization system for mobile ad-hoc networks”, Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2009 Workshop), pp. 452-457, 2009.

[9] D. Estrin, M. Handley, J. Heidemann, S. McCanne, Y. Xu, and H. Yu, “Network visualization with Nam, the VINT network animator” Computer 33 (11), pp. 63–68, 2000

[10] Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, A Visualization and Analysis Tool for NS-2 Wireless Simulations: iNSpect, Proceedings of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 503-506, 2005.

[11] J. Mark Belue, Stuart H. Kurkowski, Scott R. Graham, Kenneth M. Hopkinson, Ryan W. Thomas, and Joshua W. Abernathy, “Research and analysis of simulation-based networks through multi-objective visualization” In Proceedings of the 40th Conference on Winter Simulation, pp. 1216-1224, 2008.

[12] Georgia Institute of Technology, “Georgia Tech Network Simulator (GTNets)”, Last Accessed 29 July 2010, http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/

[13] Karygiannis, A. and Antonakakis, E. mLab: A Mobile Ad Hoc Network Test Bed. 1st Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing in conjunction with the IEEE International Conference in Pervasive Services 2005, July 14, 2005

[14] A. Karygiannis, E. Antonakakis, and A. Apostolopoulos, “Host-based network monitoring tools for MANETs”, Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks, pp.153-157, 2006.

[15] http://cutebugs.net/files/wns2-yans.pdf

[16] NS-3 Logging Module. http://www.nsnam.org/docs/tutorial/html/tweaking.html