

# MMAN - A Monitor for Mobile Ad hoc Networks: Design, Implementation, and Experimental Evaluation

Hanif Kazemi  
Deloitte Consulting LLP  
Washington, DC  
[hkazemi@deloitte.com](mailto:hkazemi@deloitte.com)

George Hadjichristofi  
University of Cyprus  
Nicosia, Cyprus  
[ghadjich@ucy.ac.cy](mailto:ghadjich@ucy.ac.cy)

Luiz A. DaSilva  
Virginia Tech  
Arlington, VA  
[ldasilva@vt.edu](mailto:ldasilva@vt.edu)

## Abstract

Mobile Ad hoc NETWORKS (MANETs) are networks in which mobile routers are connected via wireless links forming dynamic topologies. An important function of network management in a MANET is to observe network conditions: at the node level, this may mean keeping track of the traffic load; at the network level, the system must monitor active routes and changes in the network topology. In this research, we introduce a Monitor for Mobile Ad hoc Networks, (MMAN) to address the challenges of monitoring MANETs. We formulate an overall design structure and present an implementation of our framework for a MANET running the Optimized Link State Routing (OLSR) protocol. The unobtrusive and distributed nature of MMAN allows the system to adapt to the constantly changing nature of MANETs and to provide valuable network management, security assessment, and traffic analysis information. Our system produces a dynamic picture of the network level and node level information on a graphical user interface. The system is non-intrusive, generates no additional traffic on the MANET it monitors, and requires only modest processing and storage resources.

## 1 Introduction

Wireless mobile ad hoc networks (MANETs) are self-organizing and autonomous networks, consisting of mobile nodes that act as routers and collaborate with one another to form a network. A MANET may operate in standalone fashion or as a stub network. Use of MANETs is expected to increase substantially as personal computing becomes more ubiquitous. For example, the “One Laptop per Child” program [1] has developed inexpensive laptops and plans for mass distribution to developing countries for education. These laptops will use ad hoc wireless mesh networking to form their own communication networks out of the box [2]. Another example is Ozone Inc. [3], an Internet service provider in France that uses ad hoc technologies to provide widespread commercial Internet access in Paris. Since

MANETs do not require the costly communication infrastructure of legacy wireless networks and can be easily deployed, they are a viable solution for military battle sites, disaster sites, vehicular wireless communication, etc.

Some of the characteristics of MANETs include dynamic network topologies, high likelihood of network partitioning, undefined geographic coverage area, unlimited number of participating nodes in the network, and, for certain end systems, limited bandwidth, storage capacity, battery life, and processing power. These characteristics pose a significant challenge to the management of MANETs. In particular, designing and implementing a monitoring solution that is capable of providing up-to-date information regarding the network’s overall health remains a challenge.

In this paper we propose a solution, MMAN (Monitor for Mobile Ad-hoc Networks), in which multiple monitoring nodes collaborate to produce a snapshot of network conditions. These monitoring nodes passively sniff network traffic and gather information from the network to construct partial network views. They then aggregate these local views and produce a comprehensive picture of network conditions. The communication between all management nodes takes place out of band. Our monitoring solution does not depend on the MANET to operate, hence it is robust to network partitioning, link breaks, dead nodes, and node misbehavior in the monitored MANET.

The information provided by our monitoring system can be used for network management, as well as for security assessment, including anomaly detection. Information regarding individual node behavior can be used for identifying selfishness behavior in the network. Also, an approximation of arriving and departing traffic load at each node is important in the context of quality of service, load balancing, and congestion control. Furthermore, the network topology picture can provide valuable information to network management in detecting preferred routes and bottlenecks, discovering network partitioning, and in detecting faults.

We developed a proof-of-concept implementation of MMAN, which works with the Optimized Link State Routing (OLSR) protocol. Through experimental studies

---

This work was partially supported by the National Science Foundation under Grant No. CNS-0519825.

we were able to verify the feasibility and workability of the system. The scheme proved to be robust with respect to mobility, rapid changes in the network topology and node connectivity. During our experiments we observed that MMAN was able to reflect the changes in the MANET with less than two seconds of delay. Also, when deployed in a high-traffic environment, with multiple TCP and UDP flows throughout the network, MMAN was able to report the traffic load on each node accurately and consistently. The major contributions from this work are:

- A distributed framework for monitoring and performance assessment of MANETs.
- A monitoring solution for MANETs that is capable of producing an up-to-date picture of the network topology without requiring complete coverage of the monitored network by a single monitoring node.
- A traffic assessment scheme that provides statistics of node-level traffic activity that can be used in load balancing, identification of network bottlenecks, etc.
- A solution for dynamic assessment of individual node cooperation in forwarding packets for their neighbors, which helps identify selfish and malicious nodes in a MANET.
- An implementation of a MANET monitoring system based on the methods introduced in this paper. In this implementation, the monitored MANET uses OLSR as its underlying routing protocol.

Section 2 describes previous work in the area of monitoring and fault detection of MANETs. Section 3 presents our proposed monitoring and assessment framework for MANETs. Section 4 describes the implementation of MMAN on a MANET that uses OLSR as its routing protocol. We present the results of our performance evaluation in Section 5. Finally, Section 6 summarizes the findings and contributions of this research and discusses future work.

## 2 Related work

Our proposed solution for monitoring and performance assessment of MANETs can be linked to both MANET monitoring systems [4-7] and MANET behavior-based solutions [8-12].

In [4], Badonnel, Festor and State propose a customized Simple Network Management Protocol (SNMP) and management information base (MIB) for ad hoc networks. The authors suggest a probe-based architecture based on a set of ad hoc monitoring probes spread over the network and on a manager that is responsible for constructing the whole network view. A probe is a node that captures and

analyzes network messages and builds up information in a customized MIB. Information from each node/probe is then propagated via SNMP to a manager node, which integrates all the information.

In [5] the authors propose GUERILLA, a hierarchical monitoring and management system. Similar to [4], management agents throughout the network collect information regarding their neighbors and relay their findings to a network manager. Unlike [4], the network manager processes incoming information from the agents and uses it to set network policies, which are then transmitted back to agents. GUERILLA requires development and deployment of a new SNMP MIB on all nodes. Also, since the detailed design and implementation of the system is yet to emerge, its effects on network bandwidth utilization is unclear. Both [4] and [5] provide incomplete coverage of the network, meaning that detailed traffic and cooperation information is only available for nodes that are *directly* under the coverage of a probe node or agent. Similar to [4] and [5], the authors in [6] propose the design of an Ad hoc SNMP (ANMP) with a new MIB that is installed on all of the nodes. A three-level hierarchy is used for reporting information that provides a wider coverage of network monitoring.

The authors of [7] proposed DAMON, a hierarchical agent-sink architecture for monitoring MANETs. In this scheme, nodes are equipped with a collector module, which collects statistics by analyzing the traffic control messages of the AODV routing protocol. These management agents collect network information and periodically aggregate and relay their findings to the sink(s). Transmission of monitoring information from nodes to agents and agents to sink(s) competes with existing network traffic.

Although monitoring solutions [4-7] that rely on a middle management layer to collect information from MANET nodes are scalable and can provide node and network-level information, they introduce management overhead to the network. Also, reporting status or responding to polls requires processing by MANET nodes and reduces battery life. Furthermore, these approaches fall short of addressing the problem of network partitioning, where a group of nodes are out of reach of the network monitor/probe. In our proposed solution, we do not inject any traffic into the monitored MANET. We employ promiscuous packet capture tools on independent Monitoring Units (MUs) for gathering information and out-of-band communications for transmitting the findings to one or multiple network manager nodes. MUs collect information passively and are not part of the monitored MANET, and thus network partitioning does not affect the performance of our monitoring solution. Furthermore,

we do not require MANET nodes to install new software or run any daemon, hence preserving nodes' limited resources. Our scheme's distributed nature provides scalability; the number of MUs is not predetermined before a MANET's deployment, but can increase as the size of the MANET increases. Table 1 shows a comparison of the various monitoring systems.

Features:	[6]	[4]	[5]	[7]	MMAN
New software must be installed on all nodes	•		•	•	
Injects management traffic into the MANET	•	•	•	•	
Places constraints on MANET resources (nodes' battery life and CPU consumption)	•		•	•	
Requires a new MIB	•	•	•		
Can provide network topology picture	•	•	•	•	•
Can provide cooperation assessment for nodes under coverage	•	•	•	•	•
Is robust to MANET partitioning					•
Is cheat-proof against a group of malicious or selfish nodes		•	•	•	•
Is tolerant to the loss of some management units		•		•	•

**Table 1: Comparison of MMAN to the monitoring solutions proposed in [4-7].**

Behavior-based solutions [8-12] have been proposed to detect selfish or malicious behavior in the network. (Selfishness, where a node or a group of nodes in a MANET refuses to forward their peers' packets, could significantly affect the performance of a MANET.) These approaches tend to be suited to the self-organized nature of a MANET as nodes evaluate the behavior of their peers, share their evaluations, and dynamically adjust their own behavior. For example, once there is a

consensus that a particular node is a non-cooperative one, all other members of the network avoid requesting or providing services from and to the misbehaving or selfish node.

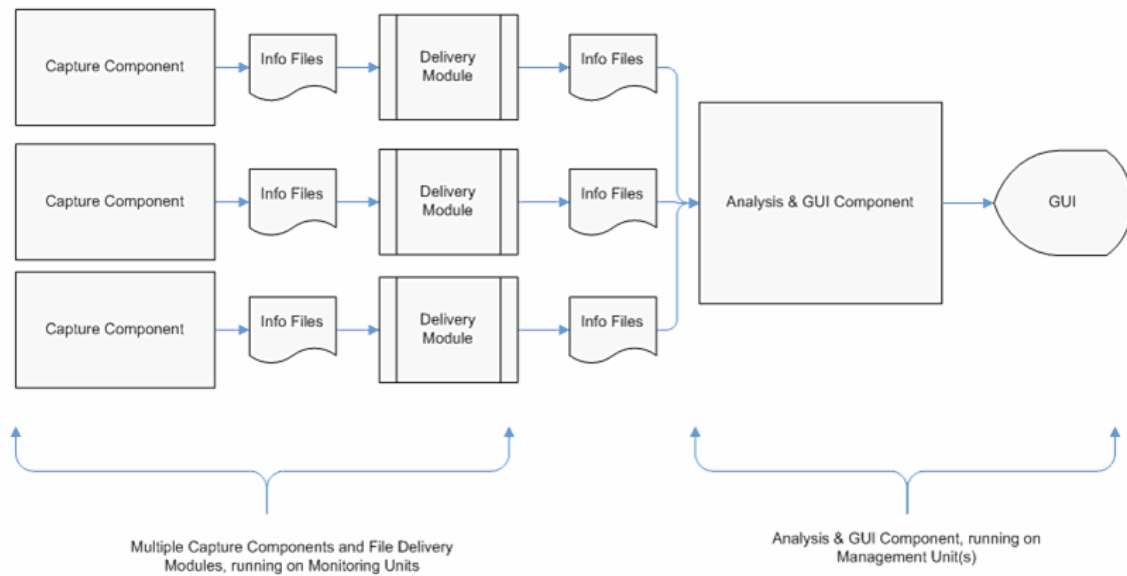
Behavior-based evaluation systems [8-12] require either secure hardware or a tamper-proof software suite to be installed on all network nodes. Also, to monitor and report misbehavior, nodes need to consume their battery, CPU power, and bandwidth, which are all scarce resources in a MANET. Unlike behavior-based assessment systems, MMAN is unobtrusive; it does not inject management traffic into the network, does not need nodes to respond to queries or evaluate their neighbors and can also detect colluding misbehavior.

### 3 Design and Implementation

One of the design objectives for our monitoring system was robustness to network partitioning, dynamic topology changes, low available bandwidth, and geographically unlimited and dispersed deployment area. The solution also had to dynamically produce an accurate picture of network topology that would provide routing and traffic information. Furthermore, by monitoring incoming and outgoing traffic from each monitored node, the system had to be able to assess nodes' cooperation in forwarding packets for their neighbors.

MMAN relies on multiple monitoring stations that collaborate and combine information to maintain an accurate and up-to-date view of the current topology. A number of Monitoring Units (MUs) are deployed throughout the MANET. These MUs are responsible for collecting information regarding network behavior (network topology picture, link changes, etc.). The information collected by the MUs is delivered to management nodes, where they are consolidated, analyzed and presented on a graphical user interface (GUI). In practice, one of the MUs can be designated as the management node.

Our solution requires that each MU be equipped with two network interfaces. One wireless interface card collects packets of the MANET in promiscuous mode while the other interface (which can be wireless or wired) is used to communicate information out-of-band between the MUs and management units. Thus, MMAN does not inject any additional traffic and results in no overhead to the monitored network.



**Figure 1: The overall flow of the system.**

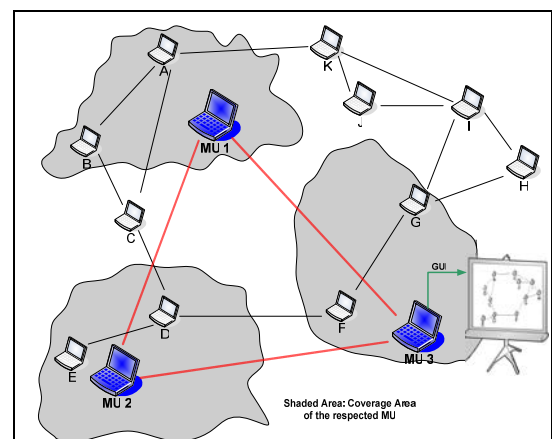
MMAN has three major independent components (Figure 1): a *Capture* component, a *File Delivery* component, and an *Analysis & GUI* component. The *Capture* component, deployed on MUs across the network, has the responsibility of sniffing and analyzing packets, and summarizing the findings in output files, called *Info Files*. These files are then communicated to the *Analysis & GUI* component via the *File Delivery* module running on all MUs.

The *Analysis & GUI* component receives the *Info Files* produced by MUs for further analysis, consolidation and production of the final outcome, which is presented on a dynamic GUI. Figure 2 represents a sample scenario for MMAN deployed to monitor a MANET.

In Figure 2, MU3 plays the role of an MU as well as the management unit. It produces a GUI output by consolidating and analyzing the incoming information from MU1 and MU2 and its own *Capture* component, providing a comprehensive picture of the network-level and node-level information. This view can be obtained over partitioned networks if MUs exist in the disjoint partitions.

The *Capture* component collects desired information from the network under surveillance. For its design, we considered two options: using polling/trap to collect information directly from each node; or promiscuously capturing packets off the air and analyzing them to obtain the specific desired information. It was our design goal to

avoid injecting any extra traffic into the monitored network, and therefore the *Capture* component is designed to capture packets in a passive and promiscuous mode, and to analyze them to extract the desired information regarding the topology, traffic, and cooperation.



**Figure 2: Multiple MUs collect information and share their findings.**

Moreover, because of possibly high packet rate, the *Capture* component is designed to simplify the

aggregation and analysis procedure and require as little processing power as possible. After pre-determined capture periods, the *Capture* component produces an *Info File*, containing a summary of its findings. The format of these output files consists of compressed information to facilitate quick and efficient data sharing between MUs and management nodes.

The GUI presents both network-level and node-level information. In terms of network level information, the GUI displays the dynamic network topology and node connectivity. The node-level information displayed includes traffic information, as well as an assessment of each node's level of cooperation in forwarding packets for its neighbors. Cooperation indicates how mobile nodes balance the conflicting requirements of conserving their own energy and bandwidth and providing a service for their peers. To report the traffic activity and represent the cooperation level of each node the *Capture* component analyzes all UDP and TCP traffic. More specifically, each *Info File* produced contains four sets of data for each node under the coverage of the MU:

1. Number of data packets originated by the node;
2. Number of data packets destined (sunked) to the node;
3. Number of data packets *expected* to be forwarded by the node; and
4. The number of data packets that the node *actually* forwarded for other nodes.

Cooperation level at each node is calculated as the ratio of the number of *actually* forwarded to the number of packets *expected to be* forwarded by that node.

The traffic load information of each node is shown as the rate of incoming traffic to a node (including packets that are expected to be forwarded and packets that are actually destined to the node) and outgoing traffic (including packets that the node has generated and packets it has forwarded for others).

The two main components of the system, the *Capture* component and the *Analysis & GUI* component, were implemented using the Java programming language in Linux environments (Fedora Core 4 and 5, Slackware Linux 10.2). Using Java helped us to take advantage of the JPCAP packet capture suite [13], a SourceForge open source project that provides toolsets for capturing network packets. The object-oriented nature of Java helped us to efficiently manage the complex task of information analysis in the *Analysis & GUI* component.

The *Capture* component in this proof-of-concept implementation was adjusted to match the specification of

the Navy Research Laboratory's (NRL) implementation of the Optimized Link State Routing (OLSR) protocol, in particular capturing and analyzing topology control (TC) and HELLO messages generated by the protocol [14].

## 4 Experiment Methodology

To validate the performance of MMAN and observe its performance in a real wireless environment, we conducted our experiments in the following two settings, both using IEEE 802.11b/g:

1. A 10-node MANET deployed across a large residential house, covering an approximate area of 1 acre (~40,000 sq. ft.), with some nodes inside and some outside the building (see Figure 3).
2. A 10-node MANET deployed in an office building (see Figure 4).

MANET nodes in both experiments were running the NRL OLSR routing protocol with different versions of the Linux operating systems: one node had *Fedora Core 5*, four nodes had *Fedora Core 4* and five nodes had *Slackware Linux 10.2* as their operating systems.

In both experiments, we evaluated the performance of our system for the following scenarios:

1. The MANET was monitored using only one Monitoring Unit, positioned to have partial coverage of the network; and
2. The same MANET was monitored by two Monitoring Units, with 80%-90% total coverage.

Our test environment was subject to shadowing, interference and signal degradation causes such as walls, electrical devices, other wireless networks (802.11 b and g) etc. Experiments were run for 6 periods, each 30 minutes long.

The approximate coverage areas of *MU1* and *MU2* are shown in Figure 3 and Figure 4. Throughout MMAN evaluation experiments, one monitoring node also performed the role of the management unit, running both *Capture* and *Analysis & GUI* components. When using two MUs, another laptop was used to run only the *Capture* component and transferred its *Info Files* to the management unit using an out-of-band wireless link.

MMAN's performance was evaluated from the following perspectives:

- Ability to produce a dynamic and up-to-date picture of the MANET topology.

- Ability to show traffic information (incoming and outgoing traffic load) for covered nodes. This capability was tested for different levels of network traffic load (high and low volume) and different types of traffic flows (TCP and UDP).
- Ability to present an assessment of cooperation level for nodes under the coverage area of the MUs, under different traffic loads.
- Assessment of storage capacity and CPU processing requirements for management nodes.

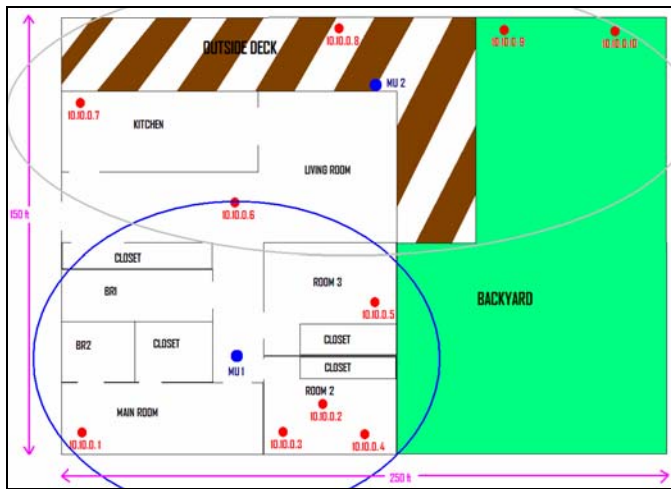


Figure 3: A 10-node MANET deployed across a 38,000 sq. ft. house.

## 5 Experimental Results

The performance of MMAN was evaluated from different perspectives as explained in Section 4. In some cases, we used *iptables* to force desired topologies and create a partitioned MANET, and in other situations natural interference and distance caused incomplete coverage and network partitioning.

### 5.1 Performance of MMAN with Incomplete Coverage and Network Partitions

To better understand the results of partial versus complete coverage of the MANET by the MUs, we examined the performance of MMAN in the network shown in Figure 3. A screenshot of the topology produced by the *Analysis & GUI* component when one MU (MU1) covers half of the MANET nodes (Figure 5) showed incomplete routing and traffic details. However, with addition of another well-positioned MU, (MU 2) we observed that MMAN was able to produce a complete topology picture of the MANET with details of traffic activity for nodes under coverage (Figure 6). Also, MMAN was able to monitor

network partitions, assuming of course that there was a MU in each partition and the MUs could still communicate with the management unit using their out-of-band communication link.

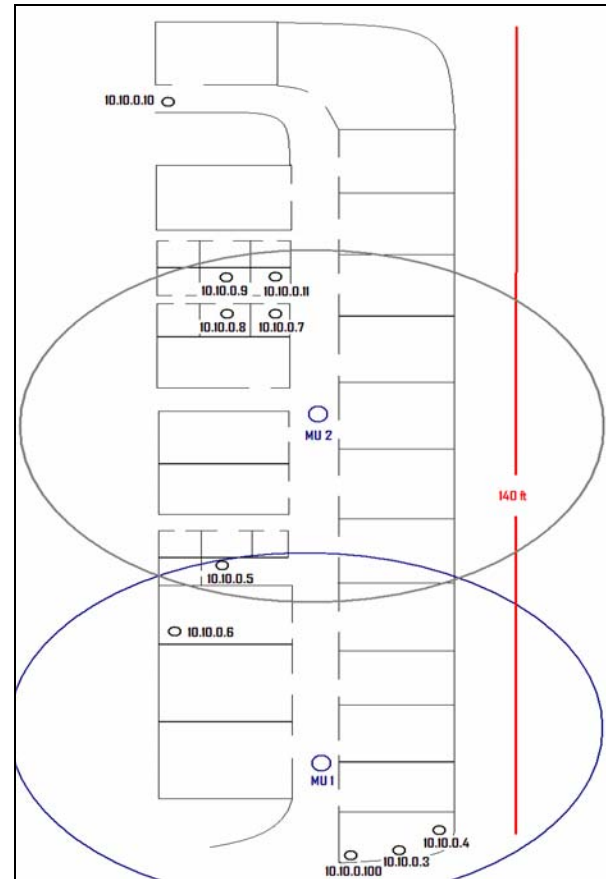


Figure 4: A 10-node MANET deployed in a typical office environment.

### 5.2. Assessing Traffic Load and Cooperation

Only nodes within the MUs' coverage area will have node level information (i.e. traffic rate and cooperation level). Since this information is produced by counting the packets that are transmitted to or from each node, it was important for us to observe the performance of the system in both low-volume and high-volume traffic environments, and when there was complete and partial coverage of the MANET.

Regardless of traffic rate (10 packets/sec to 210 packets/sec) MMAN was able to produce an accurate and dynamic picture of the traffic rate for each node under coverage (shown as an "Incoming || Outgoing packets/sec" label below each node in Figure 6). Also, when a node was configured to not fully cooperate (and would not forward some packets), we observed that

MMAN was able to correctly update the node's cooperation level (shown as a blue/red bar next to each node in Figure 6) with a ~2 seconds delay.

### 5.3 System Storage and CPU Requirements

To estimate the CPU requirements on MUs and management nodes, we took advantage of the Linux *top* utility. We observed that, on average, a *Capture* component consumes about 3-4% CPU power of our laptop (Fedora Core 5, Pentium M 1.6 GHz) and the *Analysis & GUI* component needs about 3% CPU power of the same computer. When both components were deployed on this laptop, CPU dedication to MMAN was below 8%. Table 2 shows a summary of storage capacity and out-of-band communication link bandwidth requirements for a MMAN with two MUs.

Parameter	Value
Number of <i>Info Files</i> communicated by local <i>Capture</i> component	800 <i>Info Files</i> in 40 minutes
Number of <i>Info Files</i> communicated by a remote <i>Capture</i> component	600 <i>Info Files</i> in 30 minutes
Average size of <i>Info Files</i>	630 bytes
<i>Info File</i> size range	330-830 bytes
Bandwidth requirement for transmission of <i>Info Files</i> from the remote <i>Capture</i> component to the <i>Analysis &amp; GUI</i> component	~1.6 Kbps
Total storage capacity required on the <i>Analysis &amp; GUI</i> component	< 1 MB

**Table 2: Storage capacity and BW requirements for a 30 minute run.**

## 6 Conclusions

In this paper, we presented the design, implementation and experimental performance evaluation of MMAN, our distributed MANET monitoring system. We tested the performance of MMAN under varied networking conditions – with different network densities, complete and partial coverage of the MANET, altered node cooperation levels and various traffic rates - in a real MANET environment. In general, MMAN was capable of producing the expected results under all circumstances.

We illustrated the effectiveness of our solution in detecting selfishness, showing network partitions, and its robust performance under high-volume traffic. Finally,

we presented a summary of the system's modest CPU processing and storage capacity requirements.

One area of future work is the development and testing of the *Capture* component for other ad hoc routing protocols, in particular those employing reactive routing.

MMAN was successfully deployed in the 2007 Mobile Ad-hoc Network Interoperability And Cooperation (MANIAC) Challenge ([www.maniacchallenge.org](http://www.maniacchallenge.org)), an academic competition where teams equipped with two laptops each form a MANET and compete to ensure delivery of their traffic while minimizing the resources spent forwarding other teams' traffic.

## 7 References

- [1] N. Negroponte, "One Laptop Per Child: Non-Profit Movement for Spreading Education Across the World," Aug. 2006; <http://www.laptop.org/>.
- [2] H. Slay et al., "BingBee, An Information Kiosk for Social Enablement in Marginalized Communities," *Proc. of 2006 S. African Inst. of Comp. Scientists in Developing Countries Conf.*, Mar. 2006. pp. 107-116.
- [3] Ozone Communications Inc., Wireless Internet Provider, Paris, France, Mar. 2006; <http://www.ozone.net/en/reve.html>.
- [4] R. Badonnel, R. State and O. Festor, "Management of Mobile Ad hoc Networks: Information Model and Probe-based Architecture," *Intl. Journal of Network Management*, Oct. 2005. pp. 335-347.
- [5] S. Chien-Chung et al., "The GUERILLA Management Architecture for Ad hoc Networks," *Proc. Military Comm. Conf. (MILCOM)*, Oct. 2002. pp. 467-472.
- [6] C. Wenli, J. Nitin and S. Singh, "ANMP: Ad hoc Network Management Protocol," *IEEE Journal on Selected Areas in Comm.*, Dec. 1999. pp. 1506-1531.
- [7] E. M. Belding-Royer, K. N. Ramachandran and K. C. Aimeroth, "DAMON: A Distributed Architecture for Monitoring Multi-hop Mobile Networks," *Proc. IEEE SECON*, Oct. 2004. pp. 601-609.
- [8] S. Buchegger and J.Y. Boudec, "Performance Analysis of the CONFIDANT Protocol," *Proc. of the 3<sup>rd</sup> ACM Intl. Symp. on Mobile Ad hoc Networking & Computing*, Apr. 2002, pp. 226-236.
- [9] P. Michiardi and R. Molva, "CORE: Collaborative REputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks," *Proc. of 6th Joint Working Conf. on Comm. and Multimedia Sec. (IFIP TC6/TC11)*, Apr. 2002. pp. 107-121.
- [10] K. Balakrishnan, D. Jing and V. K. Varshney, "TWOACK: Preventing Selfishness in Mobile Ad hoc Networks," *Proc. WCNC*, Mar. 2005. pp. 2137-2142.

[11] M. Baker, K. Giuli and S. Marti, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," *Proc. MOBICOM*, Aug. 2000. pp. 255-265.

[12] J. Chen, S. Zhong and Y. R. Yang, "SPRITE: A Simple, Cheat-proof, Creditbased System for Mobile Ad-hoc Networks," *Proc. 22nd Annual Joint Conf. of the*

*IEEE Computer and Communications Societies (INFOCOM 2003)*, Mar. 2003. pp. 1987-1997.

[13] JPCAP: A Java API to Libpcap Packet Capture Utility, Nov. 2006; <http://jpcap.sourceforge.net/>.

[14] Naval Research Lab (NRL) implementation of Optimized Link State Routing protocol. Oct. 2006; <http://pf.itd.nrl.navy.mil/projects.php?name=olsr>.

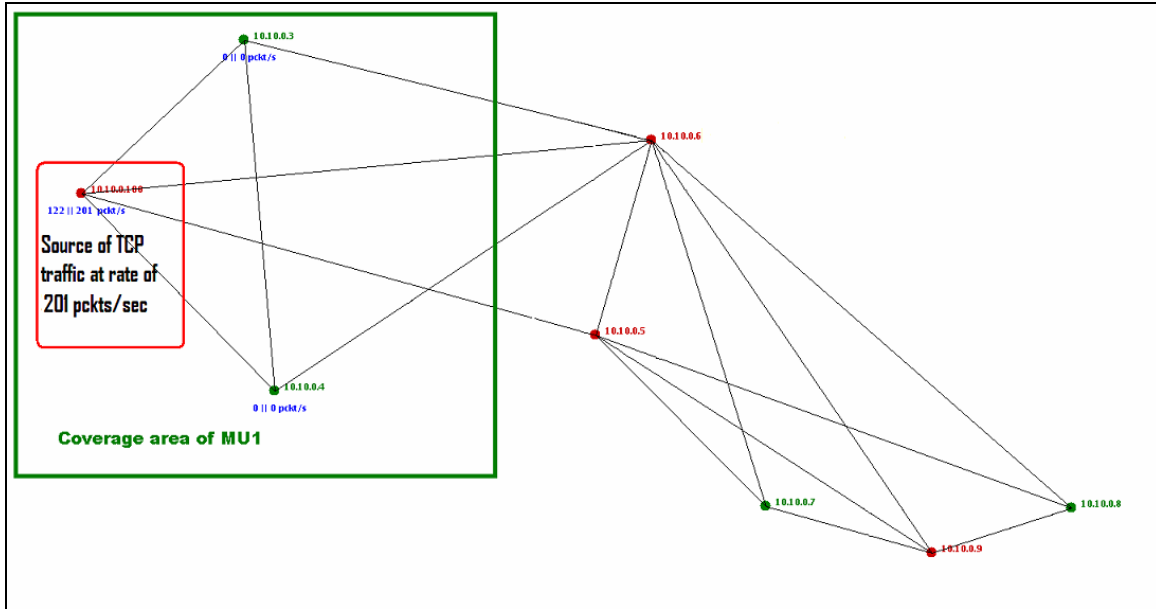


Figure 5: Incomplete coverage of the network where traffic and cooperation information is available only for covered nodes using one MU.

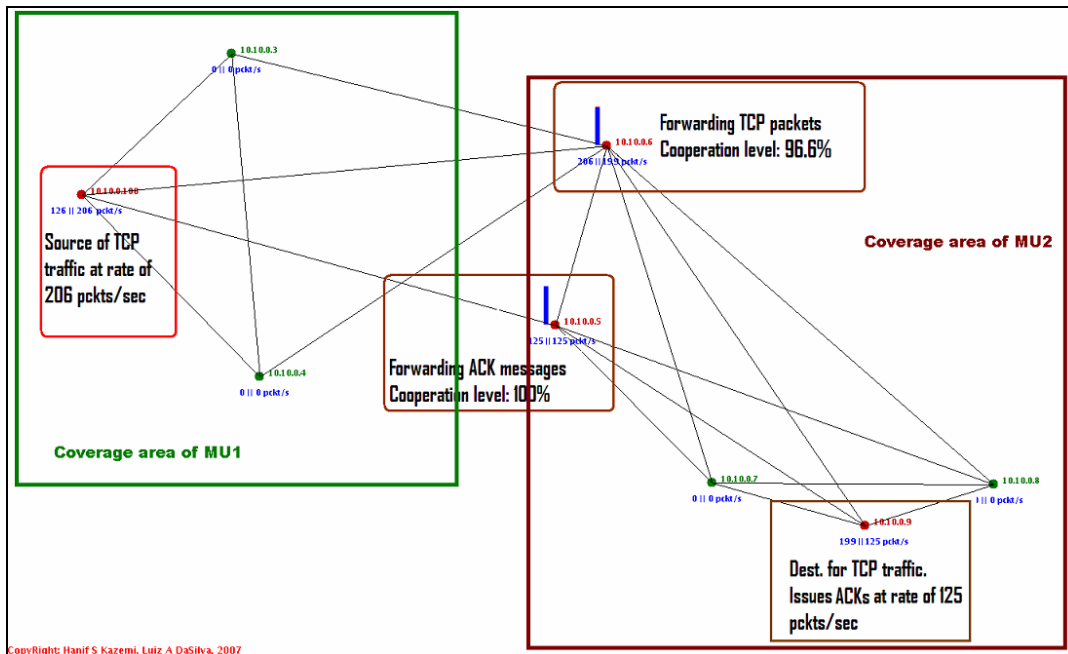


Figure 6: Use of two MUs (total coverage of ~90%) helps MMAN to produce a complete picture of the network topology and node-level information for the entire MANET.