mLab: A Mobile Ad Hoc Network Test Bed

A. Karygiannis and E. Antonakakis

National Institute of Standards and Technology
{karygiannis, manos}@nist.gov

Abstract

Over the last few years, research in the area of mobile ad hoc networks (MANETs) has focused on routing protocol performance improvements, security enhancements, power consumption optimizations, and quality of service metrics. The availability of sophisticated network simulation tools, such a NS2 and GloMoSim, has allowed researchers to study MANETs without purchasing the mobile nodes themselves or conducting costly and time-consuming field trial tests [PAPAD, HU, ZAPAT]. While there is an abundance of research based on simulations, actual implementations of ad hoc routing protocols and applications are very limited by comparison. Although proposed application areas benefiting from MANETs range from sensor networks, vehicle safety, military reconnaissance, first responder assistance, smart homes, and factory automation, these application scenarios have remained largely the domain of university researchers or government funded laboratories. This paper presents a wireless MANET test bed currently under development whose goal is to help researchers and developers bridge the gap between simulations and actual MANET deployments.
Keywords: ad hoc, test bed, MANET.

1. Introduction

The testing of ad hoc networking protocols in a laboratory environment allows researchers the opportunity to validate theories in practice, to test simulation assumptions, and to discover and prioritize practical problems facing ad hoc network users and developers alike. Testing ad hoc network implementations in a laboratory environment, however, also presents a number of challenges. The most obvious challenge is being able to test the effects of node mobility on the ad hoc routing protocols and ad hoc applications. Moreover, configuring individual nodes, installing patches, monitoring log files, updating software, and debugging beta releases of experimental software distributions on a modest size ad hoc network can be very time-consuming. Recreating realistic environmental conditions and signal transmission characteristics using off-the–shelf computing nodes and wireless cards in a laboratory setting is also very difficult [NRL]. Outdoor field tests of mobile ad hoc networks reveal realistic use case scenarios and can validate expected results under real-world conditions, but these test can be prohibitively expensive and out of reach for the typical researcher or developer [MALTZ]. The mLab test bed strikes a balance between desktop simulations and outdoor field tests by allowing users to develop and test ad hoc protocols and applications in a laboratory environment and to take simulated systems one step closer to actual deployment.

2. mLab Overview

The mLab wireless MANET test bed is currently comprised of three main tools: mNet, mDog, and mSignal. The mNet tool allows users to create arbitrary ad hoc network topologies, mDog allows users to monitor and capture packets transmitted by its neighbors, and mSignal allows the user to vary the signal strength of the transmission signal. The mLab tools have been written using C and shell scripts. A typical mLab platform is shown in Figure 1. The mLab platform consists of a central Linux desktop or laptop machine that communicates with the member nodes of the ad hoc network through a wired interface (eth0), while the ad hoc nodes communicate with each other through the wireless interface (wlan0) as shown in Figure 1. The mDog tool requires wireless cards built on the Prism2 chipset. The nodes in the mLab ad hoc network are Intrinsyc CerfCubes 255 running the

Familiar distribution of Linux.[1][2] The source code for this and other tools is available for download at http://csrc.nist.gov/manet/. The following sections provide a brief description of each tool's current capabilities, limitations, and implementation details.

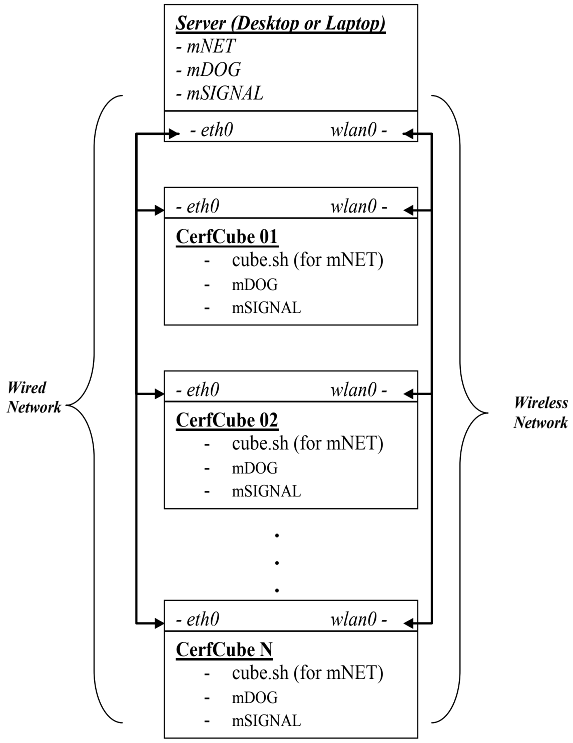
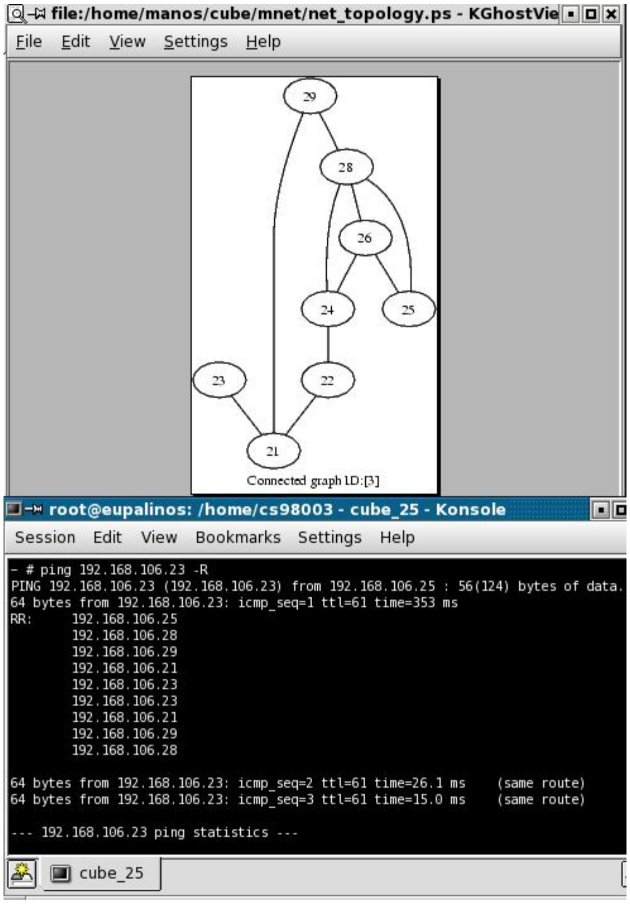


Figure 1. The mLab network configuration.

3. mNet

NIST's prototype mNET tool allows users to automatically generate arbitrary logical network topologies in order to perform real-time performance measurements of routing protocol implementations. By changing the logical topology of the network, M-NET users can conduct tests on an ad hoc network without having to physically move the nodes in the ad hoc network. Given the number of nodes in the ad hoc network test bed, each node's IP and MAC address, mNet creates arbitrarily connected graphs and updates each node's IP_TABLES accordingly through socket servers running on each network node in order to reflect the new logical topology. The arbitrary graph is represented in an adjacency matrix that is then translated into the corresponding IP_TABLES. mNet uses the open source graph visualization tool Graphviz to display the logical topology of the ad hoc network.[3] Figure 2 below shows a sample logical topology generated by mNet. The ad hoc network consists of nodes 21 through 29. Node 27 was arbitrarily generated with a node degree of 0 and therefore is considered to be beyond the transmission range of the other nodes. Figure 2 also shows a simple ping test from node 25 to node 23. In the first instance, the response time for the ping test from node 25 to node 23 was 353ms. The following two ping test response times were 26.1ms and 15.0ms. The results of this simple test reveal the difference in the route discovery time between the first ping test and the subsequent two ping tests.



```
- # ping 192.168.106.23 -R
PING 192.168.106.23 (192.168.106.23) from 192.168.106.25 : 56(124) bytes of data.
64 bytes from 192.168.106.23: icmp_seq=1 ttl=61 time=353 ms
RR:     192.168.106.25
        192.168.106.28
        192.168.106.29
        192.168.106.21
        192.168.106.23
        192.168.106.23
        192.168.106.21
        192.168.106.29
        192.168.106.28

64 bytes from 192.168.106.23: icmp_seq=2 ttl=61 time=26.1 ms    (same route)
64 bytes from 192.168.106.23: icmp_seq=3 ttl=61 time=15.0 ms    (same route)

--- 192.168.106.23 ping statistics ---
```

Figure 2. **A sample mNet output and ping test.**

[1] For more information on Intrinsyc CerfCubes see URL: http://www.intrinsyc.com.

[2] For more information on the Familiar distribution of Linux see URL: http://www.handhelds.org.

[3] For more information on Graphviz see URL: http://www.graphviz.org/.

The mNET tool allows users to save and replay different mobility scenarios, to control the maximum and minimum node degree, produces an output in the form of an adjacency matrix for further analysis, and provides a framework for building additional ad hoc network testing tools. The adjacency matrix can include weighted values to represent and control, for example, the signal strength of the transmission signal, QoS metrics, or other experimental values. Moreover, since the global topology of the ad hoc network is known, researchers can benchmark the actual performance of their ad hoc routing algorithms and applications against the theoretical optimal performance.

The mNet tool can be used, for example, to test ad hoc network Intrusion Detection Systems (IDS). An ad hoc network IDS may have difficulty distinguishing between legitimate routing errors and malicious node activity such as intentional packet dropping. IDS anomaly detection techniques and signatures may benefit from the use of mNet since this tool can be used to generate mobility scenarios and test the IDS' ability to correctly diagnose the cause of routing errors encountered.

The current implementation of mNet generates successive arbitrary graphs at predefined time intervals. When mNet generates a new logical topology, the network will experience an increase in the number of RREQ and RERR messages. An IDS must be able to differentiate between legitimate RERR messages and those that are the result of malicious activity. Generating arbitrary topologies creates scenarios that can help researchers test experimental IDS systems under difficult conditions. Although this creates a challenging environment for testing ad hoc networks, realistic mobility scenarios are more likely to be characterized by fewer and more gradual movements. We hope to include a gradual transition from one arbitrary graph to another in a future version of mNet. Although we are able to control each node's logical perception of the network topology, we cannot control the network congestion resulting from having a large number of ad hoc nodes in close proximity in a relatively small research laboratory. The scalability of mLab requires additional study, but we expect that this issue will not prevent budget-conscious researchers from collecting useful results a modest size network.

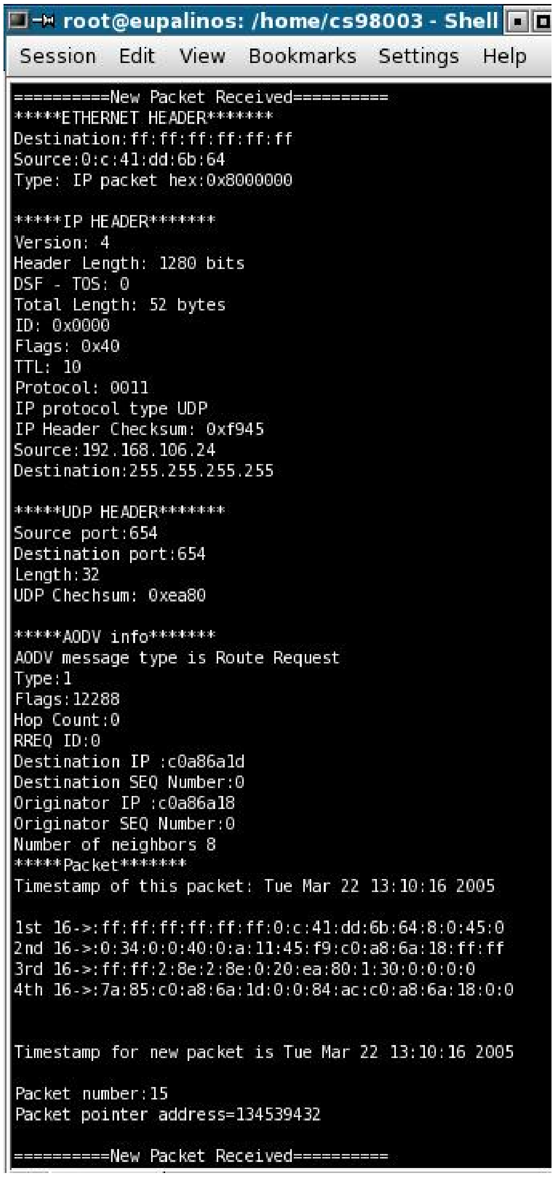
4. mDog

Traffic monitoring in wired networks is usually performed at switches, routers and gateways, but an ad hoc network does not have these types of network elements where audit data for the entire network can be collected. A wired network under a single administrative domain allows for discovery, repair, response, and forensics of suspicious nodes. A MANET is most likely not under a single administrative domain, making it difficult to perform any kind of centralized management, control, fault diagnosis, or intrusion detection. Network traffic can be monitored on a wired network segment, but ad hoc nodes or sensors can only monitor network traffic within their observable radio transmission range [ZHANG]. mDog is an ad hoc network protocol packet monitoring and capture tool, or what is commonly referred to as a packet sniffer.

The mDog tool is capable of providing diagnostic information on packets sent and received by nodes in an ad hoc network. mDog allows a user to compare the expected output of a node to the observed output of the node without logging in or having a user account on the node under observation. The ability to observe the behavior of nodes not under another node's administrative control allows mDog to serve as a useful tool for ad hoc network monitoring, diagnosis, and IDS. mDog requires wireless cards that are built with the Prism2 chip set because it supports the promiscuous mode.

Figure 3 shows a sample mDog output of an AODV RREQ packet sent from a source node with an IP address 192.168.106.24 (hexadecimal number c0:a8:6a:18) to node with an IP address 192.168.106.29 (hexadecimal number c0:a8:6a:1d). The node with IP address 192.168.106.29 is represented by the hexadecimal number c0:a8:6a:1d. This packet is the originating RREQ that is broadcast (IP address 255.255.255.255) and has a Hop Count of 0. In addition, mDog displays the RREQ ID, Destination and Originator Sequence Number, Source and Destination port, a time stamp, and the raw hexadecimal output [PERK1].

mDog can be used for fault diagnosis and as a module of IDS Watchdog-based systems [ZHANG]. For example, simulations often fail to differentiate between malicious network activity and spurious, but typical, problems associated with an ad hoc networking environment. A node that sends out false routing information could be the one that has been compromised, or merely one that has a temporarily stale routing table due to volatile physical conditions. A malicious node can drop packets on purpose or due to network congestion. Malicious nodes may enter and leave the immediate radio transmission range at random intervals or may collude with other malicious nodes to disrupt network activity and avoid detection. Malicious nodes may behave maliciously only intermittently, further complicating their detection. A node that sends out false routing information could be a compromised node, or merely a node that has a temporarily stale routing table due to volatile physical conditions [PATWA].



```
==========New Packet Received==========
*****ETHERNET HEADER*******
Destination:ff:ff:ff:ff:ff:ff
Source:0:c:41:dd:6b:64
Type: IP packet hex:0x8000000

*****IP HEADER*******
Version: 4
Header Length: 1280 bits
DSF - TOS: 0
Total Length: 52 bytes
ID: 0x0000
Flags: 0x40
TTL: 10
Protocol: 0011
IP protocol type UDP
IP Header Checksum: 0xf945
Source:192.168.106.24
Destination:255.255.255.255

*****UDP HEADER*******
Source port:654
Destination port:654
Length:32
UDP Chechsum: 0xea80

*****AODV info*******
AODV message type is Route Request
Type:1
Flags:12288
Hop Count:0
RREQ ID:0
Destination IP :c0a86a1d
Destination SEQ Number:0
Originator IP :c0a86a18
Originator SEQ Number:0
Number of neighbors 8
*****Packet*******
Timestamp of this packet: Tue Mar 22 13:10:16 2005

1st 16->:ff:ff:ff:ff:ff:ff:0:c:41:dd:6b:64:8:0:45:0
2nd 16->:0:34:0:0:40:0:a:11:45:f9:c0:a8:6a:18:ff:ff
3rd 16->:ff:ff:2:8e:2:8e:0:20:ea:80:1:30:0:0:0:0
4th 16->:7a:85:c0:a8:6a:1d:0:0:84:ac:c0:a8:6a:18:0:0


Timestamp for new packet is Tue Mar 22 13:10:16 2005

Packet number:15
Packet pointer address=134539432

==========New Packet Received==========
```

Figure 3. A sample output of the mDog tool showing the capture of an AODV RREQ packet.

5. mSignal

By manipulating each node's IP-TABLEs, the mNet tool is able to create a logical topology different from the actual physical topology. Although mNet is able to externally control the logical topology of the ad hoc network through some network programming techniques, controlling the physical properties of the radio signal itself requires additional work in order to emulate real world conditions where the quality of the wireless signals are affected by node mobility and the immediate environment.

The mSignal tool allows users to vary the signal strength of the wireless card by using hostap-drivers0.2.6 and C code. The hostap-drivers0.2.6 includes an experimental txpower function that can control the signal strength for wireless cards that support multiple transmit powers. The txpower function, however, cannot precisely control the signal strength because the experimental algorithm in the hostap driver does not include any feedback from the measured transmission power and allows transmission power values that affect the signal quality or cause interference. The updated txpower function is expected to be available in future releases of the hostap driver.

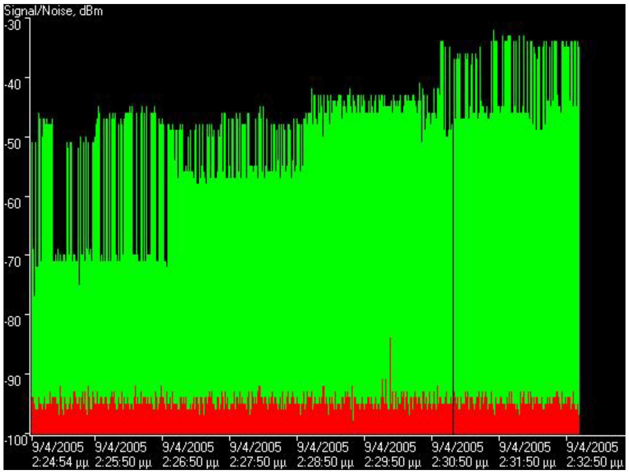


Figure 4. A sample wireless card output showing how the experimental txpower function can vary the signal strength.

Figure 4 shows a sample output of a wireless card whose signal strength is being varied by the txpower function. The txpower function accepts inputs ranging from –43dBm, the lowest, to 20 dBm, the highest. Note that there is not a one-to-one mapping of programmed power to measured power in this experimental version and that users would have to perform calibration measurements for their cards. Even though the experimental txpower function cannot precisely set the power output of the card, the ability to linearly vary the output will allow for useful experimental results.

mSignal also allows the user to set the wireless card to maximum power or to turn the wireless card off. Turning the wireless signal off without updating the corresponding IP_TABLES can be used to simulate a faulty node with stale IP_TABLES. Setting the wireless card to the maximum power can be used to diagnose transmission problems. Varying the signal strength can also be useful for conserving power on resource-constrained devices. mlab can be also be use to conduct research in Quality of Service (QoS) QoS routing refers to the discovery and maintenance of routes that can satisfy QOS objectives under given resource constraints. For example, if a node was aware of the approximate distance of its nearest neighbors based on the received signal strength and the density of the network, the node could reduce the signal strength as appropriate or estimate the probability of successful communication over a particular route [PUNNO]. We are currently investigating ways to better emulate the physical characteristics of the transmission medium and to further improve this tool.

mSignal also allows individual nodes to monitor the signal strength of packets received from neighboring nodes. By monitoring the changes in the signal strength of neighboring nodes, mSignal along with mDog, can try to infer certain properties in the immediate transmission range of the node, such as, the quality of the transmission medium, distance to other nodes, velocity, likelihood of packet loss, and QoS metrics.

6. Conclusions

This paper presented a prototype MANET test bed currently under development whose goal is to help researchers and developers bridge the gap between simulations and actual MANET deployments. mLab includes several tools that can help IDS developers test and verify that their systems can detect malicious activity under realistic conditions. IDS mLab can also be used as a learning tool for student laboratory exercises in advanced networking courses. Many ad hoc routing protocols have been proposed by researchers, but due to the relatively high cost of building and testing an ad hoc network, much of the work is limited to simulations. Although simulations are well suited for studying scalability and performance issues, practical problems are often overlooked and experimental insights lost. The goal of our project is to create an open source mLab test bed that will allow developers to test, troubleshoot, and monitor the performance of their ad hoc networking protocols and applications. The current version of mLab allows researchers and developers to create logical topologies, control the signal strength of the radio signal, and capture packets transmitted over the wireless interface. Researchers interested in building on the existing work are encouraged to contact the authors.

10. References

[CORSO] S. Corson, and J. Macker, 1999. Mobile ad hoc networking (MANET), IETF RFC 2501, January, 1999.

[HU] Yih-Chun Hu, D. B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), pp. 3-13, IEEE, Calicoon, NY, June 2002.

[MALTZ] J.B. Maltz and D. Johnson. Lessons from a full-scale multi-hop wireless ad hoc network test bed. IEEE Personal Communications Magazine, 2001.

[NRL] NRL Mobile Network Emulator, Naval Research Laboratory, Washington, DC, January 24, 20003.

[PAPAD] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks, January 2002.

[PATWA] A. Patwardhan, J. Parker, A. Joshi, A. Karygiannis and M. Iorga. Secure Routing and Intrusion Detection in Ad Hoc Networks, Third IEEE International Conference on Pervasive Computing and Communications, Kauaii Island, Hawaii, March 8-12, 2005.

[PERK1] C. Perkins and E. Belding-Royer and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, July 2003.

[PUNNO] R. J. Punnoose et al. Optimizing Wireless Network Protocols Using Real-Time Predictive Propagation Modeling. RAWCON, Denver, CO, August 1999.

[ZHANG] Y. Zhang and W. Lee. Intrusion detection in wireless ad hoc networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, pages 275–283. ACM Press, 2000.

[ZAPAT] M. G. Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. In Internet Draft, 2002.