ABSTRACT

To support research in wireless mobile networks and mobile ad-hoc network security, the U.S. Army Research Laboratory (ARL) has developed a “Wireless Emulation Laboratory” (WEL). A key component of the WEL is a Mobile Ad-hoc Network (MANET) emulation testbed on which algorithms and applications can be subjected to emulated wireless network conditions. The testbed is based on the MANE (Mobile Ad-hoc Network Emulator) software originally developed by the Naval Research Laboratory (NRL). It has since been improved through the incorporation of advanced modeling methods and computing technologies. Important additional features include (1) the integration of the terrain integrated rough earth model (TIREM) propagation model, (2) the use of virtual machine technologies to scale the size of the network, and (3) the inclusion of custom-designed mobility patterns to create a specific dynamic topology of a MANET under test. Currently the WEL testbed can emulate a 101-node MANET and, through the use of virtualization technologies, will scale well beyond that number. This paper discusses the current capabilities of ARL’s WEL for conducting empirical evaluation and demonstration of MANET technologies and concludes with planned future enhancements.

I. INTRODUCTION

Mobile Ad-hoc Networks (MANETs) are the future military tactical communications environment. Since the army is relying heavily on the communications network to maintain information dominance and success, we must ensure a good understanding of how applications are affected in such environments. The emulation environment described in this paper provides us with the capability to evaluate applications and network protocols under a wide number of scenarios and at sufficient scale without having to conduct large numbers of expensive field tests. We recognize the need for realistic test environments which capture the complexities of the tactical operating environment. We are addressing technical risks for future military networks and focusing on the performance and security of MANETs which are a key element in the Army vision. Our approach to emulating mobile ad-hoc environments allows for the analysis of applications under realistic scenarios so that greater understanding can be gained which will lead to network designs with predictable performance. This paper describes our improvements for the WEL and provides details of our initial Mobile Ad-hoc Network Emulator (MANE) environment. This environment supports various mobile ad-hoc protocol algorithms and information assurance issues related to MANETs. It also allows for the establishment of a realistic MANET environment and provides the capability to test applications subjected to real effects. This environment has proven useful in the study and understanding of network algorithms, protocols and information assurance issues within tactical military communications environments.

In section II we discuss the details of the Mobile Ad-hoc Network Emulator. In section III we go into the details of the improvements made to the MANE environment. In section IV we explain some of the future enhancements we plan to implement. The final section we make some concluding remarks.

II. MOBILE AD-HOC NETWORK EMULATOR

A MANET is a wireless network that can be deployed anywhere without relying on dedicated infrastructure. The performance of an ad-hoc network is dependent on the nature of radio propagation, transceiver issues, and channel contention [1]. To support research in wireless mobile networks and mobile ad-hoc network security, ARL created a “Wireless Emulation Laboratory” (WEL). The WEL provides a controlled repeatable emulation environment for the research, development, and evaluation of communication and security algorithms for tactical wireless mobile ad-hoc networks. A key component of the WEL is the mobile ad-hoc network emulation (MANE) environment on which algorithms and applications can be subjected to emulated wireless network conditions.

Our initial MANE consisted of 4 MANE servers, 48 test nodes, and a controller node. The servers are used for controlling the connectivity on the MANET. Each of the 48 test nodes emulates a mobile entity and contains a Pentium 4, 3GHz processors, 1 Gigabit of RAM, and 2 Gigabit
Ethernet interfaces. One interface is connected to the controller node for control, monitoring, and data collection, while the second interface directly connects the test node to one of the MANE servers as shown in Figure 1. The MANE servers, using position data periodically advertised by the controller node, as well as global parameters for transmission power, noise power, and modulation type, compute the MANET connectivity and forward or corrupt packets between the test nodes. The MANE servers themselves share a separate server-to-server data channel over which packets are forwarded between test nodes residing on different servers.

![Figure 1: WEL visual representation](image)

Our MANE software suite, originally developed by NRL, runs within a Linux based distribution environment with IPTABLES network filtering capabilities providing the support of firewall functions and filtering [3]. Each test node runs Fedora Core 3 Operating System. The test nodes are all configured using the OLSR (Optimized Link State Routing) protocol daemon developed at the University of Oslo’s UniK organization, and a TNPacketTreatment, responsible for emulating the effects of the MAC layer [4]. MANE incorporates data from a software tool called Topodef. Topodef allows the user to define network scenarios by describing location information (i.e. emulated GPS longitude and latitude data) for each test node. Topodef will be discussed in detail under section III. The GPS Emulator (GPSE) reads from the log files and multicast the positions of the nodes involved within a given scenario to the MANE servers and test nodes. The servers use the GPS information to determine which nodes have communication links with each other. The Range Model uses a probabilistic (free-space loss) model to determine the likelihood of the arrival of a packet based on the location of the source and destination nodes. It works in conjunction with the Forwarding Engine to determine which packets get dropped [3]. The Forwarding Engine is responsible for forwarding the information between the test nodes. This current capability provided us with the baseline architecture to provide an emulation environment leading to improvements.

### III. IMPROVEMENTS

The baseline MANE environment has proven useful but lacks the required fidelity and scalability to fully support our research needs. To improve MANE’s utility, we have integrated higher fidelity propagation models, provided improved MANE capability as described in the sections below.

#### III.I. INTEGRATION OF TIREM

The baseline version of MANE determines network connectivity using the free-space loss propagation model. The free space loss propagation model does not take into account the effects of terrain or antenna properties. Because of this, it is not a good predictor of network connectivity in complex terrain. The Terrain-Integrated Rough-Earth Model (TIREM), originally developed for Department of Defense (DoD) usage, provides a more accurate radio propagation prediction model by taking into the account transmitting medium, antenna properties, and the terrain profile between the transmitter and the receiver. TIREM is distributed by the Defense Information Systems Agency Joint Spectrum Center for DoD users and is available for use in the Windows XP and RedHat Linux 9.0 based environment [4]. TIREM has become an Army standard for computing the path loss of wave propagation. However, it cannot be used directly by MANE due to the real-time running constraint imposed by emulation. We have integrated TIREM into MANE by implementing a pre-computed path loss option in MANE.

The new option in MANE allows TIREM to preprocess path losses based on the GPS location and terrain information and store the resulting data in binary format (single precision floating point) file. By taking advantage of symmetrical wireless link property, the pre-computed path losses of wireless links are the same for a pair of nodes in both directions to save disk space, memory space, and processing time.

With pre-computation of path losses, MANE can expedite the determination of connectivity and the decision to forward network packets in the MANE testbed and meet the real-time running constraint. The computation times are not only saved by pre-processing but also by eliminating duplication of computation for packets from the same source sent in very close time together where locations of nodes do not change. The new option is not limited to TIREM. Any propagation model can be used to prepro-
cess path losses between nodes and be provided as inputs for this pre-computed-path-loss option of MANE.

### III.II. VIRTUAL MACHINE TECHNOLOGIES

Virtual machine technologies are used in the WEL to facilitate scaling the MANET size in the order of 100’s which will be difficult to emulate due to limited physical space, power requirements, physical devices and network devices. Currently, virtual machine software (VMware) is used to create virtual machines corresponding to test nodes. Taking advantage of multi core processors and large memory capacity, each host server running VMware can concurrently run 62 virtual machines (test nodes) in the current implementation. Each test node is configured with two network interfaces (NIC). One NIC is configured to connect to the control channel local area network (LAN) by bridging to the network interface of the host server and connecting to the control LAN. The other NIC is configured as a test interface which connects separately to the MANE server through a separate virtual LAN connecting host server and MANE server. Virtual local area networks (VLANs) are created by using a 1 Gigabit Ethernet interface between a host server and a MANE server. The software and functionality of the MANE servers and test nodes are identical to the regular MANE configuration.

![Figure 2: Architecture and connection diagram of a MANE server and two host servers.](image)

The current implementation of host servers is based computer systems with two quad core 2.66GHz-Xeon processors and 32 Gigabit of memory. With 256 Megabit memory, configuration for each virtual machine, a host server can sustain 50 virtual machines without using pre-computed-path-loss option. Using the same hardware configuration of host servers, an implemented MANE server can maintain a network emulation of 101 test nodes which we have implemented now into the laboratory. A plan for extending virtualization by using faster processors (3.16GHz instead), more MANE servers, more host servers per MANE server, and more test nodes per host server to increase the emulation capability to 1000 nodes is underway.

### III.III. CUSTOM DESIGNED MOBILITY PATTERNS

Mobility patterns play an important role in the study of MANETs. Network performance can vary greatly depending on the mobility models employed or even if the parameters of the same mobility models are varied [4]. The custom-design of mobility patterns refers to the process of creating a dynamic scenario composed of a series of changing network topologies that meet a particular requirement of a dynamic MANET simulation or emulation.

A specific network topology depicts the explicit placement of the participating nodes at the particular locations where they can form the exact number of internodal links at a certain time and for a certain duration. The establishment of a link between any two nodes is determined by applying a range model; e.g., a fixed range or free-space loss model. Once a scenario has been completed, the time-dependent geographical position information of all the participating nodes are extracted and saved into log files, which will be used to drive the emulation testbed as depicted in Figure 3. This custom-design process is easily accomplished using the ARL Topodef tool, a computer-aided design and animation system and method for designing ad-hoc network topologies [5]. The tool enables a series of topologies to be created from scratch or derived from an existing topology. The former method is often utilized to satisfy a specific need; e.g., a research requirement or training and demonstration purposes. The latter is often used to conduct what-if experimentation to test and evaluate a newly developed MANET technology.

### III.IV. GPS EMULATOR

The GPS Emulator is responsible for distributing emulated node location information to MANE’s forwarding engine, range model and each emulated node within the testbed. The initial version of GPSE would obtain the GPS infor-
mation for each node from script files and multicast the entire scenario to the testbed without giving researchers an option to pause the scenario at specific points for network analysis. We have enhanced GPSE to allow this functionality by including a “pause”, “resume” and “step” feature. These new features make it possible for researchers to run a mobile test repeatedly on MANE to a specific point of interest, pause the scenario and collect data at this point with a greater sense of consistency.

In addition to providing researchers with this flexibility, we saw a need to provide the capability to visualize mobile scenarios as tests were being conducted. GPSE was further modified to work in conjunction with network topology visualization tools like, JavaMap (JMAP) and Watcher. JMAP is a Java-based mobile network visualization tool developed at the Naval Research Lab that supports a wide variety of testing and experimental applications. Watcher is a visualization tool developed by Sparta that visualizes the network’s hierarchy and has the ability to display specific network attacks on MANET’s. GPSE was modified to multicast the GPS position information to both JMAP and Watcher enabling the visualization of all the emulated nodes, the network connectivity between them and their movement as the mobile scenarios progressed.

IV. FUTURE CAPABILITY/ENHANCEMENTS

Although the WEL testbed based on MANE has proven its value for wireless network research, there are several areas that we intend to significantly enhance in order to achieve even greater fidelity. These improvements are discussed below.

IV.I. MULTI-USER INTERFERENCE MODEL

MANE currently has only a limited ability to model the effects of multi-user interference. Each packet forwarding decision is made solely on the basis of bit error rate, which is determined by combining transmitted power and destination noise level with a range model (originally a free-space loss model, now incorporating terrain effects precomputed by TIREM), for either BPSK or QPSK modulation. A packet that is corrupted is simply not forwarded. Since the WEL uses gigabit Ethernet links, packets are transmitted nearly instantaneously with little interaction, unlike the relatively slow channels of the emulated communications.

When used without Test Node Packet Treatment (TNPT), MANE ignores the possibility of packet collision, so forwarded packets are received without error. TNPT adds a simple model which discards any packets received during the emulated time interval spanned by the first received packet, based on the specified channel rate. TNPT also queues packets for transmission at the emulated channel rate. There is currently no provision for MAC-level forward error correction.

Real-world interference effects are much more complex. Because they can significantly affect traffic flow, we are working on improvements in this area. To support the emulation the interference model must operate in real time. Interference modeling could be centralized on the MANE servers by adapting the packet forwarding module. This would require that actual packet transmission be delayed until the end of the time span for its emulated transmission, so that interference from overlapping packets can be applied. These effects could be modeled simply as added noise, or using a more detailed model based on signal characteristics.

An alternative is to adapt the TNPT module to model interference at each receiver node. For better fidelity all packets should be forwarded to the receiver. We plan to first alter the TNPT to drop all overlapping packets, then to gradually incorporate further refinements.

IV.II. AUTOMATED TRAFFIC GENERATION

Within the current MANE environment, traffic is generated using the Real-time Application Representation (RAPR) tool. RAPR is open source software that was developed as part of the software suite incorporated within the MANE package developed by NRL. Its purpose is to generate and respond to real-time network traffic that resembles various application behavioral patterns in a controlled and repeatable fashion. RAPR uses the Message Generation (MGEN) toolkit in order to shape and form network traffic behavior. The behavior of the network is defined according to the characteristics of such applications as http, VOIP, video server/client, and other application generated events. These events are defined by sending TCP, UDP, or a combination of the two in scheduled intervals using user-defined input scripts.

Currently these input scripts are hand-generated to incorporate the various types of traffic created by each individual node within the network. Future improvements include incorporating the creation of the input scripts for RAPR into the ARL Topodef tool. This will alleviate the process of hand-generating input scripts which can become a very cumbersome task as the number of nodes continually increases.

IV.III. HIGH PERFORMANCE COMPUTING

The scalability of the emulation environment is achieved by adding additional commodity computing hardware. This limits the size of the emulation environment to the order of 100’s of nodes due to constraints such as hardware cost, power required, cooling issues and bandwidth.
constraints. In addition, the fidelity of the emulation is limited by need to abstract the physical layer and the effects of propagation, attenuation, and interference, in order to run in real-time.

One way to get around the hardware constraints is to use high performance computing (HPC) hardware. The DoD has a significant investment in HPC hardware that can be leveraged to scale the emulation capability to beyond a 1000 nodes. In addition to the availability of HPC systems with 1000’s of processors, HPC environment has the software tools in place to facilitate the rapid transfer of data between processors, which can increase scalability significantly. In addition, in order to support modeling, the reaction of the C2 applications to communications losses and force-on-force models, real-time RF calculations are needed. Currently the RF calculations have to be pre-computed using scripted mobility. HPC will facilitate the incorporation of detailed RF propagation and physics models running in real-time.

The DoD High Performance Computing Modernization Office has funded ARL to establish a Software Application Institute to develop multi-disciplinary expertise and software tools that transform the ways in which DoD models, simulates, emulates, and experiments with dynamic reconfigurable mobile tactical networks.

In the near term, ARL’s Mobile Network Modeling Institute will reengineer the emulation environment to operate on the HPC resources installed at ARL’s MSRC, and extend the size of the networks that can be emulated past a 1000 nodes. In the mid term, the Institute will extend the HPC MANE environment to support real-time interactions with force-on-force and C2 applications and the implementation a real-time high fidelity RF propagation modeling component into MANE.

In the longer term, the HPC MANE environment will be extended to support virtualization of nodes, allowing multiple mobile nodes to be instantiated on a single HPC processor allowing the emulation network to be extended to support 5,000 nodes.

V. CONCLUSIONS

We have described an emulation environment for analyzing tactical mobile networks. Our realistic emulation environment enables the analysis of large networks in arbitrary scenarios without the need for field tests. This approach provides both transparency and control of ad-hoc behaviors that would otherwise be difficult or impossible to repeat live in the field but which are important in the study of different network applications, behaviors and the performance of such applications. The WEL environment provides the capability to orchestrate, observe, record and analyze the performance of applications and protocols in realistic MANET operating environments. It also provides a platform for transition and live demonstration capability for program managers.

REFERENCES


