A TECHNIQUE FOR DETERMINING RADIO-SIGNAL PROPAGATION IN AN EMULATED WIRELESS ENVIRONMENT

Brian B. Luu, Rommie L. Hardy, and George T. Tran
U.S. Army Research Laboratory
Adelphi, MD

ABSTRACT

The Mobile Ad hoc Network Emulator (MANE) test bed relies on accurate determination of radio signal propagation to emulate realistic mobile ad hoc network (MANET) connectivity. The determination of radio wave propagation is a very complex process that depends on numerous factors such as the transmitting medium (e.g., vacuum, air, or water), antenna properties of transmitters and receivers, and the geometrical environment (i.e. terrain). The Terrain-Integrated Rough-Earth Model (TIREM) software is an Army standard for computing the path loss of radio wave propagation, but it cannot currently be used directly by the MANE system software to determine network connectivity. This paper describes the implementation of an efficient preprocessing technique for using and integrating TIREM into the MANE software system to improve the precision of the calculated radio propagation path loss. The implemented technique has expedited the determination of connectivity and the decision to forward network packets in the MANE test bed.

INTRODUCTION

Originally developed by Naval Research Laboratory, the Mobile Ad hoc Network Emulator (MANE) [1] is a software and hardware test bed tool used to emulate a mobile ad hoc network (MANET) environment.

The U.S. Army Research Laboratory (ARL) has improved and used a version of MANE test bed to conduct research on wireless mobile ad hoc networks. One of the improvements is the integration of the Terrain-Integrated Rough-Earth Model (TIREM) into MANE to provide more fidelity in calculating the radio propagation path loss. More accuracy in determining the radio path loss in turn provides more accuracy in determining network links of nodes in MANET environment. The integration of TIREM in determining the radio path loss also expedites the decision to forward network packets in the MANET environment.

A new interface was created to provide an option to use the pre-computed path losses in the MANE test bed. This paper describes the implementation of an efficient preprocessing technique for using and integrating TIREM into the MANE software system to improve the precision of the calculated radio propagation path loss.

In the next section, we provide background information on MANE and TIREM. Then, we describe the pre-computed technique and discuss the benefit of using the technique. In the conclusion, we mention an underway plan of using high performance computing systems and MANE with pre-processing technique to increase MANET emulation capability at ARL.

BACKGROUND

MANE

As mentioned above, MANE is a computer software system working along with some hardware to emulate a MANET system. Four main programs (modules) of the MANE software were used in the ARL test bed: Global Positioning System (GPS) Emulator, Forwarding Engine, Range Model, and Test Node (TN) Packet Treatment.

- **GPS Emulator module**

The GPS Emulator (GPSE) program reads in time-stamped GPS information and distributes it to test nodes using network multicast. For ARL implementation, a MANET scenario, which contains GPS information of test nodes at time steps, is generated by the ARL *Topodef* tool [2] (a visual system and method for designing specific mobile network topologies). At every time step, GPS positions (latitude, longitude, and altitude) are provided for each test node by their corresponding *log* files as collective information called a network topology. All topologies make up a scenario of the entire movement of MANET nodes (test nodes).

The GPSE reads in the GPS information of test nodes from the *log* files containing their positions at a particular time in the format of hours:minutes:seconds and position location in degree format for longitude, latitude, and in meter unit for altitude. This longitude and latitude format is converted into a compressed format using positive integers (instead of double precision floating point) and scaled by a factor, 60000, to retain precisions (up to 1 thousandth in minute degree of longitude and latitude), and the altitude is rounded off to a nearest integer. GPSE has a number of existing options along with added improvement from ARL to alter the GPS scenario information, such as to pause all test nodes at one particular topology or to step to a later topology specified by a step number. Usually the GPS Emulator program should be executed by a system separated from test nodes or the MANE server.

- **Range Model module**

The Range Model program is used to determine byte error rate (*ber*) (the probability of error of a byte in a transmitted packet) of emulated wireless network packets which are being transmitted in an emulating mobile ad hoc network environment. The determination of *ber* is based mainly on a given transmitting power, noise power, signal modulation, and free space path loss (FSPL). The model is implemented with only two signal modulations, BPSK (binary phase-shift keying) and QPSK (quadrature phase-shift keying). There is no implementation of signal modulation changes during the communication transmission. The computed *ber* will be provided to the Forwarding Engine program to determine the packet error rate (PER) by an approximation formula (*ber* x packet length) to save computation time. But, for the implementation at ARL, the PER is computed by the formula $(1-[1-BER]^{L})$ where BER is the bit error rate, and L is the packet length in bits. Therefore, for ARL applications, the Range Model program will provide BER to the Forwarding Engine program instead of *ber* to improve calculation accuracy.

- **Forwarding Engine module**

Normally, both the Forwarding Engine program and the Range Model program are executed by a central system called the MANE server. This central system has a connection to all of the network interfaces (which emulate wireless interfaces) of the test nodes. Multiple MANE servers can be connected together (through server-to-server interface) to form a cluster of nodes that participate within the same MANET environment. The program reads all the data packets received on the interfaces of a server connected to that of the test nodes, forwards the received packets to other MANE servers, and either forwards or drops the received data packets to the interface(s) of destination test node(s) depending on the computed information provided by the Range Model program.

- **TN Packet Treatment module**

The TN Packet Treatment program is executed at test nodes to control the amount of network traffic going to and from the test network interface of test node to emulate the effects of the MAC layer. The amount of network traffic coming in and out of test nodes is limited to a constant value specified in the MANE configuration file by a RAW-CHANNEL-CAPACITY parameter. The outbound network traffic (defined as output traffic from test node applications) is regulated by blocking applications, and the inbound network traffic (defined as input traffic to test node applications) is controlled by dropping packets. The channel rate of test network interface is calculated over the interval of time specified by the RATE_LIMIT_WINDOW parameter with the minimum of 0.24 second for 1500-byte packets (corresponding to 50kbps). No delays of packet are used to control inbound traffic.

TIREM

The Terrain-Integrated Rough-Earth Model [3] is a computer software library that calculates basic median propagation loss (path loss) of radio wave over irregular earth terrain. The calculation method was developed in the early 1960s, revived, improved, and evolved into a TIREM software version for use by the Department of Defense (DoD). The DoD TIREM version is now distributed by the Defense Information Systems Agency Joint Spectrum Center for DoD users. The TIREM code was written in the FORTRAN programming language, but had a C++ programming interface. It is available for use in the Microsoft Windows XP and Red Hat Linux 9.0 based environment based on the DoD version.

The TIREM can be used for radio frequencies in the range of 1 through 20,000 MHz over terrain elevations which are specified by a set of discrete points between the great-circle path of the transmitting antenna and receiving antenna. The earth terrain information can be provided by the digital terrain elevation data (DTED). TIREM provides more accuracy in the radio propagation model than the FSPL model by taking into account the transmitting medium (surface refractivity and humidity), antenna properties (height, frequency, and polarization), and geometrical environment (relative permittivity, conductivity, and terrain elevations). The calculation of path loss is also determined by the effects of free space spreading, reflection, surface wave propagation, diffraction, tropospheric scatter propagation, and atmospheric absorption but not ducting phenomena, fading, ionospheric effects, or absorption due to rain or foliage.

PRE-COMPUTED TECHNIQUE

To effectively use TIREM for MANE, a C++ program is used to invoke TIREM to compute the path losses of radio transmission between any pair of test nodes whose GPS positions are provided by *log* files. The result of the computed path loss of each pair of test nodes for every topology in a given scenario is saved in a binary format file (i.e., path-loss-matrix file). This file will serve as an input into the Range Model module to determine the link connection between any pair of communicating test nodes in a topology.

Since the latest version of TIREM is available for interfacing with a C++ program in a Microsoft Windows XP environment, a C++ program named *"plt"* was developed and executed in the Windows XP environment to create a binary file (path-loss) to store the computed path loss of every symmetric pair of nodes. The *plt* program reads in the information of *log* files (each test node has an associated *log* file). It then reads in the GPS information of all the test nodes at every topology from the *log* files and calls functions in the TIREM library to compute the path loss in dB (decibel) for every pair of nodes. In this current version, a pair of nodes is considered symmetric in radio communication (equal path losses in either direction (send or receive)). Based on the reciprocity theorem [4], this assumption is reasonably acceptable even with earth terrain and with no consideration of the effect of the earth ionsphere layer. By using symmetric communication, the size of the path-loss-matrix file is reduced in half. The following is a structure description of the path-loss-matrix file:

- List of node identification numbers (IDs) in binary integer format (typically 4 bytes).
- The network topology number in binary integer format (typically 4 bytes)
- Path losses in binary floating point (single precision) format (typically 4 bytes in IEEE 754 format [5] (IEEE Standard for Binary-Floating Arithmetic) for most system implementation) of strictly upper triangular matrix (not including diagonal) elements (symmetric communication).
- The data structure is repeated until the last topology of the scenario.

Using a typical 4-byte size for integer and floating point representation, the path-loss-matrix file size can be determined by:

$$\left[n+\left(1+\frac{n^2-n}{2}\right)nt\right]4 \quad bytes$$

where n is the number of test nodes, and nt is the number of topologies in a scenario.

The computed path loss results are listed sequentially, based on the rows of the strictly upper triangular matrix, where the elements are specified by the node IDs listed at the beginning of the path-loss-matrix file. Using a list of node IDs to specify the strictly upper triangular matrix enables the MANE server to read in and locate the pre-computed path losses.

The output of the *plt* program is the computed path-loss-matrix file which is used as an input for the modified MANE program which accepts a new option specified by *–p* or *–precomputed-pathloss* and the path-loss-matrix file name. With the binary format representation in the path-loss-matrix file, the Range Model module can quickly read in pre-computed path losses and speedily determine the link connectivity of node pairs by using the overloading function feature of C++ programming.

The current path-loss computation process is illustrated in Figure 1. The numbered arrows show the flow of data in the process. The first step is to use the ARL Topodef tool to design a specific dynamic topology of a MANET under test and to generate a set of *log* files, the arrow labeled (1). The second step is to use the *plt* program to read in the contents of the *log* files (2) and use the TIREM library (3) to compute and generate the path loss matrices (4). The last step is to transfer the path-loss-matrix file (and the *log* files) to the MANE test bed.

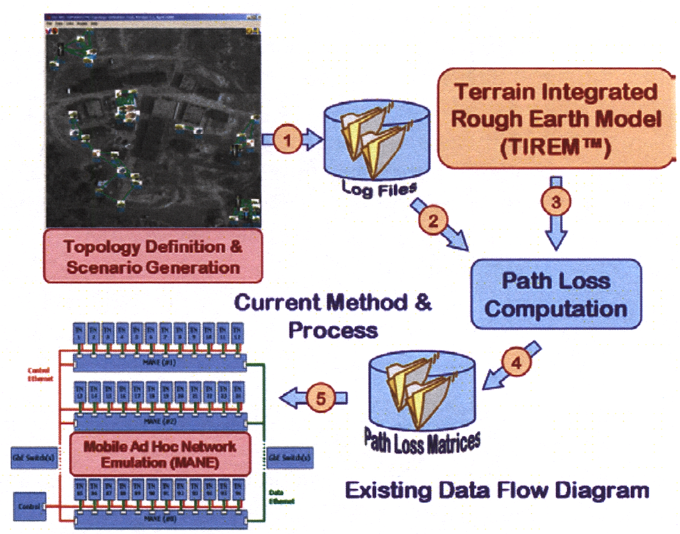

Figure 1. Pre-computation process and the MANE test bed

Executing the MANE software system with the *–p* option specifies that it should use the pre-computed path-loss matrix file. Conversely, without the specified option, the path-loss matrix is disregarded by the MANE system. Whether the option *–p* is specified, the MANE system always requires that a set of *log* files be available since programs in MANE need them.

If the path-loss matrix is not pre-computed, the MANE server will use the mobility traces (contents of the *log* files) and a propagation model, e.g., the FSPL model, to compute the path losses for every network packet it receives. Then it uses the computed result to determine the packet error rate for that particular packet (sending/receiving between those two nodes), which affect its decision to forward or drop the packet.

DISCUSSION

For every packet generated by a test node, it requires n-1 computation of path losses to determine link connectivity in the emulated wireless MANET environment. Hence, the path-loss computation is computationally intensive and heavily depends upon the number of test nodes in a given topology. Therefore, it may not be computationally practical to determine the wireless link connectivity in real time if the number of test nodes drastically increases (e.g., an order of magnitude) for a realistic scenario or the number of packets being generated by test nodes substantially increases (e.g., a fast data rate network) for a practical data rate communication.

Using the *–precomputed-pathloss* option will help accommodate the scalability required in the MANE emulation by alleviating two challenges for the MANE server: (1) minimizing computation and (2) speeding up data packet forwarding in near real time. The time it takes to compute link connectivity is sped up by making the path-loss information readily available and by eliminating the duplication of computation for packets of same sources that are sent during the same topology (in very close time together) where locations of test nodes do not change. Minimizing computation will clearly enhance the ability of the MANE server to quickly decide whether network packets should be forwarded (in near real time) to emulate link connectivity between test nodes.

Evaluating the performance of the MANE system with and without the *–precomputed-pathloss* option presents quite a challenge due to the limited amount of time available to accurately and efficiently test, compare, and analyze results. The basis of evaluation will be determining if there is a level of accuracy gained by using the pre-computed path-loss matrix instead of the free space path loss model and if there is any system degradation (loss of performance in processing link connectivity) due to heavy computation requirement. The performance of the MANE system depends on a number of factors such as processors used by the MANE server (where the computation of link connectivity occurs), capacity of the local area network (LAN) connecting MANE server and test nodes (for emulating wireless interfaces of test nodes), and the number of test nodes emulated in a MANET scenario. A simple way to test system degradation is to use a large number of test nodes as discussed above. Using a large number of test nodes is restricted by the current hardware implementation (which is limited to 48 test nodes). ARL has resorted to use virtualization technology to implement a large number of test nodes (virtual machines as virtual nodes). Therefore, efforts have been concentrated to implement and tune hardware and software to support emulation of large number of virtual test nodes. Future tests will be conducted to determine whether using this method (pre-computed path loss) on a large scale topology (greater than 100 nodes) will sustain system performance and link connectivity for properly emulating a MANET environment.

The pre-computed option is only usable if the GPS information of the test nodes are made available prior to real time execution of the MANE server. One way to resolve the computation of the path-loss matrix in real time is to use a multi-core processor system for the MANE server and run *plt* on a separate processor as a thread or a separate process and compute and update the path-loss matrix table only for nodes having positional changes in a subsequent network topology. The cost of computing the path-loss in the aforementioned approach is sped up for packets with the same source and destination and being transmitted close in time. This minimizes the number of updates being made to the path-loss matrix.

The main obstacle of implementing the computing path loss using TIREM in real time is the unavailability of the TIREM source code or the compiled TIREM library for the same executing environment of MANE software (which is Linux environment at kernel 2.6 level). Therefore, ARL was able to successfully employ only the Windows version of the TIREM library to compute the path loss.

CONCLUSION

The pre-processing technique for using and integrating TIREM into the MANE software system provides more precision in determining the link connectivity of an emulated MANET environment and expedites the decision of forwarding network packets, which is the fundamental process of emulating a MANET environment. With anticipation, using the pre-computed path loss option in the MANE server will enable ARL to conduct experiments with the emulation of a MANET system consisting of 500 to 1000 nodes. The fast decision of forwarding network traffic also provides more fidelity in testing network links of high data rates. Considerations and plans of using high performance computer systems (e.g., multi cluster processor systems) along with MANE and pre-computed-path-loss option to experiment MANET environment with large number of nodes (e.g., 5,000 nodes) are underway at ARL through the DoD Mobile Network Model Institute program.

Acknowledgement

The authors appreciate useful comments from anonymous reviewers and the support from ARL management.

The authors also thank Mr. Binh Nguyen of ARL for editing the paper and for illustrating the path-loss computation (Figure 1).

Disclaimer

The findings in this paper are not to be construed as an official Department of the Army position unless so designated by other authorized documents. Citation of manufacturer's trade names does not constitute an official endorsement or approval of the use thereof.

REFERENCES

[1] Network and Communication Systems Branch, Mobile Ad-hoc Network Emulator (MANE), The U.S. Naval Research Laboratory (NRL), Washington, D.C. http://cs.itd.nrl.navy.mil/work/mane/index.php (accessed June 17, 2008).

[2] Nguyen, Binh, "The ARL Topodef Tool for Designing Mobile Ad-Hoc Network Topologies to Support Emulation," Military Communications Conference, 2007. MILCOM 2007. IEEE, pp. 1-6, 29-31 Oct. 2007.

[3] The Terrain-Integrated Rough-Earth Model TIREM, http://handle.dtic.mil/100.2/ADA296913 (accessed June 17, 2008).

[4] Simon R. Saunders and Alejandro Aragon-Zavala, "Reciprocity," pp. 70-71, in *Antennas and Propagation for Wireless Communication System*, 2nd edition, John Wiley & Son Ltd, 2007.

[5] IEEE 754 Standard for Binary Floating-Point Arithmetic. URL: http://grouper.ieee.org/groups/754 (accessed June 17, 2008).