Complex Network Modeling with an Emulab HPC

Virginia Watson Ross
High Performance Systems Branch
AFRL/RITB
Rome, NY, USA
Virginia.Ross@rl.af.mil

David L. Hench
Information Transmission Branch
AFRL/RITE
Rome, NY, USA
David.Hench@rl.af.mil

***Abstract*– To support DoD networks in the field, next generation complex network product designs need to be evaluated for optimum performance. Network emulation plays an important role in evaluating these next generation complex network product designs. From the component level to the system-of-systems level, emulation enables evaluation in a real system context, greatly reducing the cost and time of testing and validation throughout the design cycle. For accurate network synthesis, emulation must support real-time speed, full packet fidelity, and provide transparency.**

For example, the Joint Tactical Radio System (JTRS) has critical needs for network evaluation, including researching the JTRS networking waveforms. With JTRS currently undergoing massive revision, this emulation can help save time and resources in modeling the network for system development and testing.

The Network Modeling and Simulation Environment (NEMSE) capability was developed and installed on the Air Force Research Laboratory/Information Directorate (AFRL/RI) EMULAB high performance computer (HPC), a network emulation testbed, to demonstrate this capability for future network modeling.

The NEMSE environment has demonstrated the capability to incorporate hardware and software elements to provide hardware-in-the-loop network emulation testing and support true network emulation. NEMSE provides parallel execution and highest fidelity models and the scalability and interactivity required to test and evaluate advanced network communication devices and architectures.

This capability benefits the DoD by enabling rapid technology transition of complex network architectures from research laboratories to the field. Actual Joint Tactical Radio System (JTRS) radios, Operations Network (OPNET) emulations, and GNU (recursive definition for GNU is Not Unix) open-source software-defined-radio software/ firmware/ hardware emulations can be accommodated.

***Index Terms*—network emulation, Emulab, OPNET**

I. Introduction

To address next generation complex network product designs, the Network Modeling and Simulation Environment (NEMSE) program was dedicated to facilitating the evaluation and maturation of fundamental research being conducted within the Air Force Office of Scientific Research (AFOSR) Complex Networks program, by creating an emulation test bed environment that addresses transition from simulation to hardware.

NEMSE provides a modeling and simulation environment where the best of current emulation techniques are available to users in an easy-to-use, integrated form on the Air Force Research Laboratory/Information Directorate's (AFRL/RI's) EMULAB. This integration includes: (1) easy installation of software and emulation tools on individual processors, (2) the ability to interact with other tools and the tool's own Application Programmer Interface (API), and (3) license management. NEMSE was designed to provide model development, protocol development, packet statistics, hardware evaluation, prototype hardware development and transition paths for all levels of the Open Systems Interconnection (OSI) protocol stack of a network architecture. NEMSE is currently available for use by DoD scientists and engineers.

The AFRL/RI EMULAB, shown in Figure 1, consists of 86 experimental nodes, each with a 3.0 GHz Quad Core Intel Xeon E5450 processor, 12 MB cache, 16 GB of memory, and a 500 GB hard drive for storage. There are six Network Interface Cards (NICs): (four experimental, one control, one

FIGURE 1: HPC EMULAB

reserved) per node. The system employs network emulation software from the University of Utah (UU) master Emulab facility. Three dedicated servers are used for EMULAB setup and control. Networks for testing are set up using Cisco 6509E programmable routers. Note that "EMULAB" (all capital letters) refers to the AFRL system, while "Emulab" is a more generic term for a system using the UU Emulab software.

The EMULAB system is accessed through an HPC and is remotely accessible, allowing outside researchers to take advantage of these unique system capabilities from geographically separate research sites.

II. Background

The design, scaling and portioning of network emulation test beds is an area of currently active research. Alvarez et al. [1] advanced the idea that for a system under test (SUT), one can consider simulation to model the behavior entirely, emulation to utilize actual entities in real time, and a test bed to be the hardware and software where a scaled version of the targeted system is evaluated. Alvarez et al. also surveyed partitioning techniques and scalability.

A. Emulation Environments

Two of the currently popular test beds are the original Emulab test bed at the University of Utah and the DETER test bed, hosted at both USC ICI and at the University of California, Berkeley (Emulab: Total network testbed [2]). There are many other Emulabs, such as the AFRL/RI EMULAB, also employing the Emulab software from the University of Utah. In most Emulab experiments, one processor of the available processor pool is assigned to each node of the simulation.

The basic Emulab system uses the university standard Network Simulator 2 (NS2) network emulation package and uses Tool Command Language (TCL) scripts. This emulation technique is effective for studying protocols that are above the physical layer, including the critical link layer where routing protocols are developed. While a subset of NS2 is available in the basic EMULAB package, the full NS2 can be easily installed on individual nodes.

This one-node to one-machine type of assignment leads quickly to a trade-off between network fidelity and resource utilization. If the investigator is not prepared to be selective in fidelity and chooses to maintain perfect fidelity for each of the multiple hops between networks, all of the applications on all of the machines feeding each node would have to be scripted. However, this action would quickly exhaust an 86 node EMULAB cluster and would require many programmers to create scripts and run applications. Even expanding each of the processors to fully utilize their eight cores would be insufficient to maintain perfect fidelity. Therefore, selective fidelity is necessary.

Alvarez et al. [1] reviewed efforts to partition scenarios into test beds and proposed a method of partitioning larger experiments into sub-experiments using flow graph techniques. On the other hand, simulation tools such as MATLAB, Optimized Network Engineering Tools (OPNET), NS2, and CORE (a modeling environment from Vitech) generally use reduced fidelity with traffic either modeled by discrete event simulation, with simulated packets, or by statistical analytic flows and demands to generate results. Many of these tools can be used for emulation with real time interfaces or in co-simulation, as described by Harding et al. [3]

Carneiro et al. [4] discussed the need for software reuse from the early simulation phases through final hardware implementation. This reuse is especially important in a research lab where there is a very short time window for software evaluation between receipt of a contractor's or university's software and hardware to integrate the product and evaluate it. Also, the variety of software tools used by contractors makes the task doubly daunting, unless the contractor's development environment can be exactly emulated. NEMSE enables emulation and reuse of software, addressing this need.

Thuente [5] summarizes twenty years of modeling and simulation (M&S) experience, identifying OPNET as the de facto standard simulation tool for modeling and simulation of military airborne communication systems. He points out that OPNET mixes Discrete Event Simulation (DES) and statistical flows. With its Wireless for Defense package, OPNET has excellent mobility and antenna modeling capability. OPNET is generally considered a modeling package with simulated packets in its DES simulation. However, with its System-in-the-Loop (SITL) models, it becomes a real-time emulation package with real packets in the computer's Network Interface Card (NIC) and real packets out of another NIC. OPNET has excellent packet data collection and statistical analysis. OPNET, however, is not a source of real packets.

B. Motivation for NEMSE

Recent failures of networking implementation in the field demonstrate the need for a pragma change in DoD network system development. Corrin [6] called networking in Afghanistan "the perfect storm," with hurdles in transition, network operations, and infrastructure issues that could be addressed by emulation in the proper test beds. The failures in the networking waveforms for the Joint Tactical Radio System (JTRS) program have been highlighted by Axe [7]

Hench [8] has described the Close Air Support Connectivity (CASCON) modeling terminal which was used in the early phases of NEMSE before the EMULAB was available. This terminal used a patch panel to interconnect individual processors instead of the programmed method of the EMULAB.

NEMSE is designed to avoid such problems in network emulations by employing not just a test bed, but a complete modeling and simulation environment. In this environment, field data from hardware, or actual hardware-in-the-loop from the test bed, can be used to verify and validate models. These models can be developed using a variety of software development tools by different contractors and universities, and integrated into the over-all scalable networking emulation.

C. A New Paradigm: Converging Emulation with System Development and Testing

Future military network emulation environments can do well by following the NEMSE model. The emulation environment should be able to support war gaming with operator-in-the-loop to allow for evaluation by military personnel of system performance. This environment should then aid in rapidly prototyping new hardware that can be taken to the field and tested, generating new data to validate and verify new models, restarting a new iteration of the cycle. This concept is shown in the work flow diagram of Figure 2.

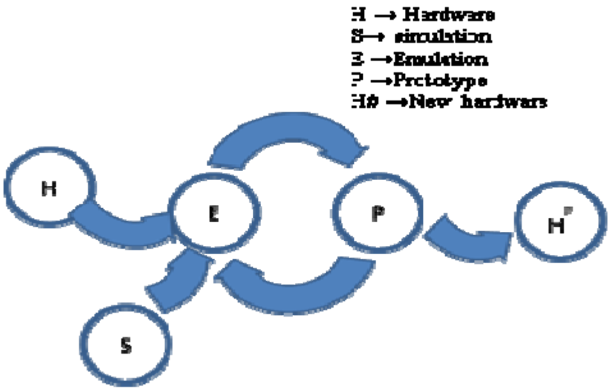


FIGURE 2: IDEALIZED WORK FLOW DIAGRAM FOR EMULATION

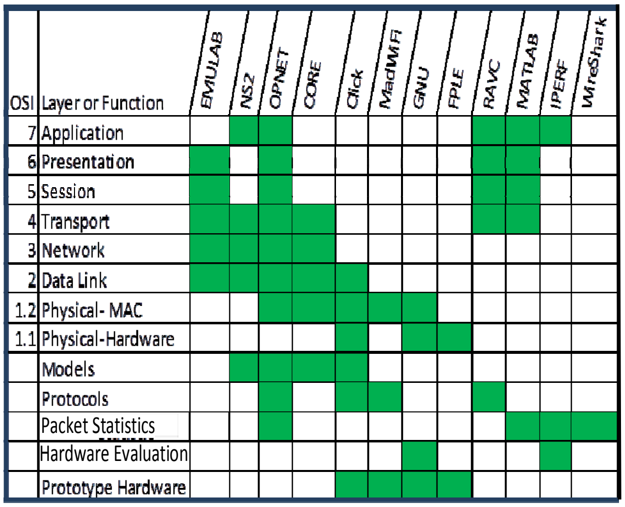
Chruscicki and Hall [9] described an initial environment utilizing a smaller test bed that mixed network emulation with military simulation tools. NEMSE has considered these military simulation tools to be important but out-of-scope of current activities.

III. IMPLEMENTED COMPONENTS

The AFRL/RI EMULAB has been enhanced by the Network Modeling and Simulation Environment developed by the NEMSE team. This environment allows incorporation of carefully selected university, commercial, and DoD standard emulation software. NEMSE provides effective support for both operator-in-the-loop and hardware-in-the-loop testing. During development, there has been an effort to allow reuse of contractor simulations, prototypes and code in emulating a scalable solution in the test bed and in designing updated hardware. The simulation and emulation tools that have been integrated into the environment of NEMSE are shown in Table 1. These tools are described in more detail below.

The left side of this table lists the seven layers of the open systems interconnection (OSI) stack of a network architecture, including the two sub layers of the physical layer, and, below them, the capabilities emphasized by NEMSE: Models, Protocols, Packet Statistics, Hardware Evaluation, and Prototype Hardware.

Table 1: Simulation/ Emulation tools for NEMSE



OSI	Layer or Function	EMULAB	NS2	OPNET	CORE	Click	MadWiFi	GNU	FPLE	RAVC	MATLAB	IPERF	WireShark
7	Application		🟩	🟩						🟩	🟩	🟩	
6	Presentation	🟩		🟩						🟩	🟩		
5	Session	🟩		🟩						🟩	🟩		
4	Transport	🟩	🟩	🟩	🟩					🟩	🟩		
3	Network	🟩	🟩	🟩	🟩								
2	Data Link	🟩	🟩	🟩	🟩	🟩							
1.2	Physical- MAC			🟩	🟩	🟩	🟩	🟩					
1.1	Physical-Hardware					🟩		🟩	🟩				
	Models		🟩	🟩	🟩	🟩							
	Protocols			🟩		🟩	🟩			🟩			
	Packet Statistics			🟩							🟩	🟩	🟩
	Hardware Evaluation							🟩				🟩	
	Prototype Hardware					🟩	🟩	🟩	🟩				

A. EMULAB

NEMSE research has demonstrated that arbitrary networks of individual processors are easy to set up in the EMULAB. Therefore, a process of engineers designing networks using a standard symbol set, intelligent scripting of components, and a system administrator implementing experiments for the engineers, has been implemented using intelligent scripts and standard symbols. If the system administrators get overloaded with experiments, a GUI based design could then be generated. This process is more efficient than a more traditional method of starting with the GUI, as described by Benchaib and Hecker [10], and avoids the steep learning curves for scientists and engineers involved in using multiple emulation tools.

A typical network drawing using this technique is shown in Figure 3. This network uses a standard processor (FreeBSD operation system) accessible from secure shell over VPN, two Windows XP processors accessible from remote desktop over

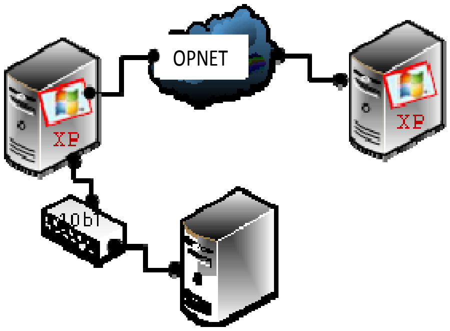


FIGURE 3: A TYPICAL NETWORK DRAWING

VPN, a network switch, and a two terminal cloud using OPNET run time with SITL.

NS2, an emulation tool, has a variety of sources and sinks that can emulate real traffic. A sub set of NS2 is utilized in the EMULAB Tool Command Language (TCL) scripts but the full ns-allinone code for network simulation can be included in a processor.

B. OPNET

An OPNET emulation setup is shown in Figure 4. The SITL module makes this emulation real time, bringing real digital video packets in, which have been captured from a ROVER in field experiments, and sending real packets out.

C. CORE

The Common Open Research Emulator is a DoD available network emulation tool that allows emulation on one or more PCs that is designed to connect in real-time to physical networks and routers. It is a commonly selected tool by DoD network development contractors and needs to be included.

D. Click

Click (Morris et al. [11]) is a software architecture for building flexible and configurable routers that was developed at Massachusetts Institute of Technology. A Click router is assembled from packet processing modules called elements. Individual elements implement simple router functions like packet classification, queuing, scheduling, and interfacing with network devices. A typical Click router is meant to be used in the Linux kernel module. This means that a hardware implementation of the Click developed router is a simple embedded Linux module with processing power matched to measured requirements in the emulation.

E. MadWiFi Driver

MadWiFi driver is an open source 802.11 Linux driver with well documented source code. A proprietary Hardware Abstraction Layer (HAL) in binary code is available for popular Atheros WiFi modules. Since MadWiFi is an actual Linux driver, emulation can be ported to a Linux embedded system with an Atheros card. The Atheros cards are used by the University of Utah Emulab, and the RF physical-layer hardware emulator designed for NEMSE allows prototype hardware development in the physical layer.

F. GNU Radio

GNU Radio, another simulation tool, is open source software for building and testing software-defined radio systems. Used in connection with a suitable hardware device, such as the Universal Software Radio Peripheral (USRP), GNU Radio provides a simple, flexible means for emulating a variety of radio devices.

G. FPLE

An FPGA Physical Layer Emulator (FPLE) was added that allows (1) prototyping a Field Programmable Gate Array (FPGA) interconnect between two computers that can be used to emulate physical layer and channel effects, and (2) integrating an FPGA development and testing environment into the EMULAB cluster to allow simple design by an engineer with limited training.

Graphical development using standard electrical engineering techniques avoids use of layout languages such as Virtual Hypertext Markup Language (VHTML). These techniques make FPGA design accessible to engineers with little additional training.

Operational hardware or a prototype can be incorporated into the EMULAB cluster by using a specially designed NEMSE box as shown in Figure 4.

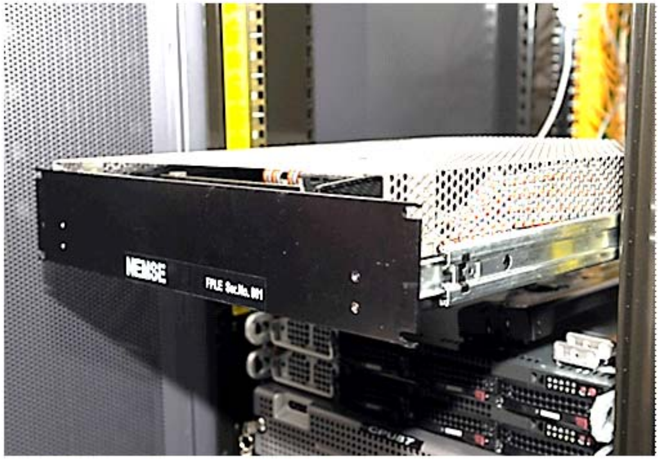


FIGURE 4: NEMSE BOX FOR FPGA EMULATION

H. RAVC

Hench [12] described the Rate Adaptive Video Coding (RAVC) technique, which has been integrated for video emulation. This technique addresses transmitting the best tactical video over bandwidths of 32 kbps to 4 Mbps.

I. MATLAB

MATLAB and OPNET have been used in co-simulations by Harding [3]; other methods of interoperating with MATLAB are also possible. Co-simulation allows MATLAB simulations to be reused in larger emulations.

J. Other Tools

The other tools listed are WireShark and Internet Performance Working Group (IPERF). WireShark is an open source packet sniffer with an excellent API for packet statistics. IPERF is a standard network engineer's tool for generating network traffic and measuring network capacity.

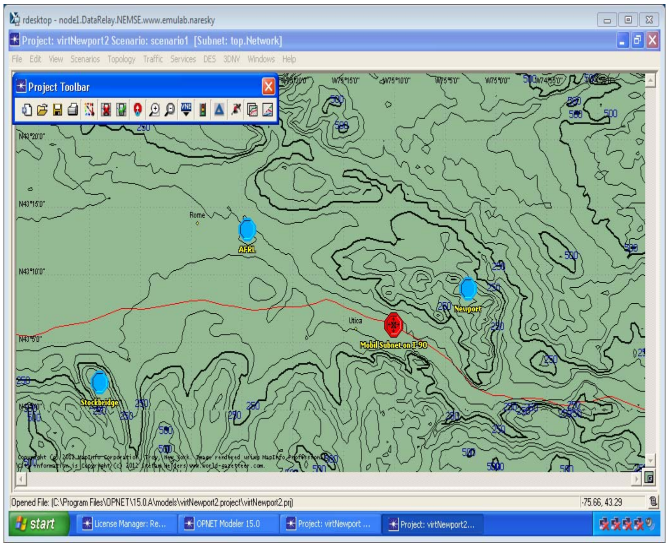
IV. EXPERIMENTAL SETUP

NEMSE addressed the development of an environment that implements the paradigm for converging emulation with system development and testing. During the first two years emulation, techniques were mainly studied in isolation. During the third year, the techniques are being integrated and made interoperable through a series of use cases and exercises that can be run by new users. A promising use case for further development outside of NEMSE is shown below.

A. Typical Use Case Results

NEMSE has demonstrated the ability to emulate all of the layers of the OSI stack via several use cases. In this demonstration, the team has completed the complex process of integrating all of the emulation packages for interoperability into a single emulation. This integration enables most network simulations and emulations to be performed.

An example use case emulation using OPNET terrain models has been demonstrated over the range from some radio towers across the Mohawk Valley in Upstate New York, demonstrating the ability to model terrain in field locations. Facilities and location make the area around AFRL/RI an ideal site for verifying and validating models of terrain as they interact with models of tactical radio waveforms, with a long valley surrounded by mountains. In addition to terrain, the Mohawk Valley is an area with four seasons and a variety of urban, farmland and forests, all of which are important to terrain modeling.

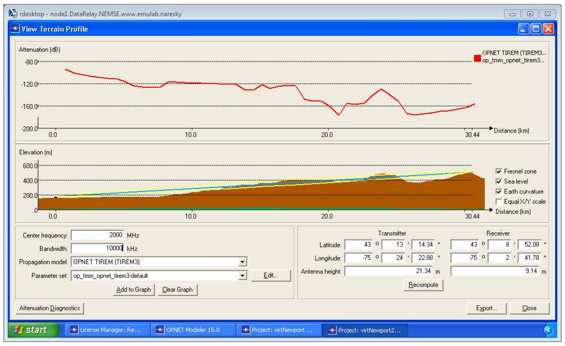

rdesktop - node1.DataRelay.NEMSE.www.emulab.naresky
Project: virtNewport2 Scenario: scenario1 [Subnet: top.Network]
File Edit View Scenarios Topology Traffic Services DES 3DNV Windows Help
Project Toolbar
Rome
AFRL
Newport
Utica
Mobil Subnet on I-90
Stockbridge
start
License Manager: Re...
OPNET Modeler 15.0
Project: virtNewport ...
Project: virtNewport2...

Opened File: (C:\Program Files\OPNET\15.0.A\models\virtNewport2.project\virtNewport2.prj)
-75.66, 43.29

FIGURE 5: MOHAWK VALLEY OPNET SIMULATION

A demonstration of this modeling capability, generated by OPNET using NEMSE, is shown in Figures 5 to 7. Figure 5 shows a map of the Mohawk Valley in Upstate New York as generated by OPNET Wireless for Defense Terrain models.

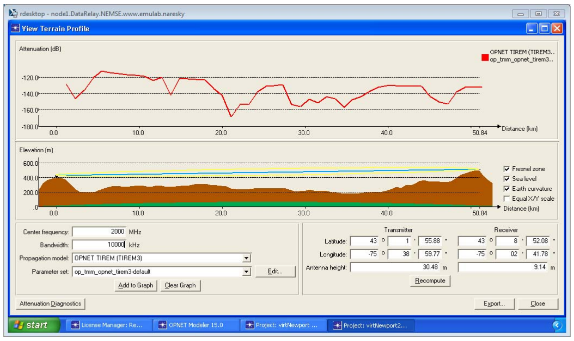
Wireless fixed subnets are placed at AFRL's Rome, Newport, and Stockbridge facilities (blue symbols). A mobile node is placed on highway I-90. Figure 6 shows the attenuation over terrain from Rome to Newport, as generated by the OPNET Wireless for Defense Terrain models. Since there is a mountain in the pathway between Rome and Newport, runs from Rome to Stockbridge and Stockbridge to Newport were selected. Figure 7 shows the predicted Stockbridge to Newport attenuation. A generic 2 GHz, 10 MHz bandwidth transmitter was emulated with an OPNET TIREM model with default parameters.



Center frequency: 2000 MHz
Bandwidth: 10000 kHz
Propagation model: OPNET TIREM (TIREM3)
Parameter set: op_tmm_opnet_tirem3-default

	Transmitter	Receiver
Latitude:	43 ° 13 ' 14.34 "	43 ° 8 ' 52.08 "
Longitude:	-75 ° 24 ' 22.88 "	-75 ° 2 ' 41.78 "
Antenna height:	21.34 m	9.14 m

Figure 6: Attenuation from Rome to Newport



Center frequency: 2000 MHz
Bandwidth: 10000 kHz
Propagation model: OPNET TIREM (TIREM3)
Parameter set: op_tmm_opnet_tirem3-default

	Transmitter	Receiver
Latitude:	43 ° 1 ' 55.88 "	43 ° 8 ' 52.08 "
Longitude:	-75 ° 38 ' 59.77 "	-75 ° 02 ' 41.78 "
Antenna height:	30.48 m	9.14 m

FIGURE 7: ATTENUATION FROM STOCKBRIDGE TO NEWPORT

V. CONCLUSION

The Network Simulation and Modeling Environment has demonstrated the ability to transition laboratory research experience into field capabilities by enabling simulations, code and prototypes to be used in iterative development of new network equipment. In addition, data from the field can be sanitized and used in emulation by academic or other researchers. This capability is expected to lead to quicker development of networked communication systems.

A variable fidelity approach to emulation is proving effective. This approach has been taken where high fidelity approaches based on actual applications and one emulation processor per one actual processor are used for a small area close to the point of study; while emulation tools allowing reduced fidelity, with either discrete event simulation or statistical flows and demands, are used for an enlarged area.

This is a new pragma for DoD networked communications starting with contractor simulation and an older generation of hardware, proceeding through multiple iterations of emulation on a variable fidelity emulation test bed with hardware-in-the-loop, and finishing with field testing of new hardware. This approach is expected to reduce the long lead time for communication equipment development and testing.

Future research can include verification and validation of models over a variety of terrain, weather, and crop covers. This approach also saves time and money by enabling reuse of simulations, prototypes, and code or multiple applications and simulations.

ACKNOWLEDGEMENTS

The authors wish to gratefully acknowledge the support of the AFOSR Complex Networks Program, with Laboratory Research Independent Research (LRIR) and also support from the DoD High Performance Computing Modernization Program for computer time on the EMULAB HPC at the AFRL/RI Affiliated Resource Center.

REFERENCES

[1] Alvarez, A., Garcia, R., Cabrero, S., Paneda, X.G., Melendi, D., and Orea, R. "In pursuit of massive service emulation: a methodology for test bed building," *IEEE Communications Magazine*, vol. 49, no. 9, pp.162-168, 2011.

[2] Emulab: Total network testbed: Other Emulab testbeds, 2012. Available at http://users.emulab.net/trac/emulab/wiki/OtherEmulabs.

[3] Harding, C., Griffiths, A., and Yu, H., "An Interface between MATLAB and OPNET to Allow Simulation of WNCS with MANETs," *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control,* pp.711-716, 2007.

[4] Carneiro, G., Fontes, H., and Ricardo, M. "Fast prototyping of network protocols through ns-3 simulation model reuse, *Simulation Modeling Practice and Theory*," vol. 19, no. 9, pp. 2063-2075, 2011.

[5] Thuente, D.J. "Methodology and ground rules for simulating airborne military communication systems," Proceedings of the 2004 IEEE Aerospace Conference, vol.2, pp. 1061- 1080, 2004.

[6] Corrin, A. "Army faces communications challenges in Afghanistan: 'Perfect Storm' of complex network issues requires attention." *Defense Systems Knowledge*, 09 Dec 2010.

[7] Axe, D., "Failure to communicate: inside the army's doomed quest for the 'perfect radio," www.iwatchnewx.org, Jan. 2012.

[8] Hench, D. L. "Simulation for Close Air Support," *Proceedings of SPIE Modeling and Simulation for Military Operations IV*, Vol. 7348, 734809, 2009b.

[9] Chruscicki, M. C. and Hall, F. "Airborne Networking Component Architecture and Simulation Environment (AN-CASE)," *Proceedings of IEEE Military Communications Conference, 2007*. MILCOM 2007, pp.1-7, 2007.

[10] Morris, R., Kohler, E., Jannotti, J., and Kaashoek, M. F. "The Click modular router," *Proceedings of the 17th ACM Symposium on Operating Systems Principles* (SOSP '99), 1999.

[11] Benchaib, Y., and Hecker, A., "VIRCONEL: A Network Virtualizer," Proceedings of 2011 IEEE 19th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp.429-432, 2011.

[12] Hench, D. L.".Rate-Adaptive Video Compression (RAVC) Universal Video Stick (UVS)," *Proceedings of Mobile Multimedia/ Image Processing, Security, and Applications 2009*," vol. 7351, 73510 S, 2009a.