Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

VALIDATING A NETWORK SIMULATION TESTBED FOR ARMY UAVS

Stephen Hamilton
Colonel Timothy Schmoyer

Electrical Engineering & Computer Science
United States Military Academy
West Point, NY 10996, USA

J. A. "Drew" Hamilton, Jr.

Computer Science & Software Engineering
Auburn University
Auburn, AL 36849, USA

ABSTRACT

Auburn University, through the Army's Aviation and Missile Research, Development and Engineering Center (AMRDEC) has been supporting the Unmanned Systems Initiative (USI) program in three research areas related to unmanned aerial vehicle (UAV). A major element in this work is the development of a high fidelity modeling and simulation testbed to support the USI program. This paper describes the testbed and the verification and validation of the testbed.

1 INTRODUCTION

An Unmanned Aerial Vehicle, or UAV, is a remotely controlled vehicle that is used to perform dangerous tasks without putting soldiers at risk. When the UAV is flown over an area, it sends surveillance video back to a ground station where soldiers analyze the intelligence gathered. Some UAVs are also equipped with weapons to mount attacks without putting American lives at risk.

Soldiers in the UAV platoon communicate with the UAV from a control station, which is connected to a base station antenna on the ground utilizing over 400 feet of various cables. The antenna at the base station relays control information to the UAV in the air and receives video signals from the UAV. One major problem with this setup is the time required to set the system up and take it down. Time is wasted running cables between the control station and base station. Since the UAV system creates such a large radio footprint, the platoon is always at risk of the enemy determining the base station location through RF Triangulation. Therefore, the platoon must be able to quickly tear down the system and move to another location.

The purpose of the simulation testbed is to evaluate alternatives to the 400 feet of cables connecting the control station to the base station with a secure wireless connection. Work is currently being done on the Shadow 200 UAV model, but this technology may later be applied to other UAV models as well.

After much testing, we decided to use the wireless implementation in Figure 1 as the basis for our information assurance research and modeling & simulation testbed. In this wireless implementation, messages are received at the UAV antenna and sent to a computer that makes the necessary video conversions. The information is then encrypted by the Cisco ASA 5510 encryption device. The encrypted signals fed into a Cisco 1240 802.11a and 802.11g (AG) access point (AP), which is used as a wireless bridge with another Cisco 1240 on the other side of the wireless connection. The messages are decrypted by another ASA 5510 and sent to the destination computer, which feeds the video messages through a wired connection to be viewed.

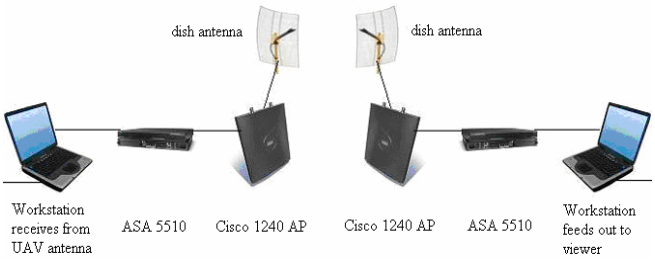

Figure 1. UAV transmission network diagram

2 VERIFICATION AND VALIDATION

A key element of this simulation test bed is the need to do appropriate verification and validation to ensure that the test bed has appropriate predictive power. Network simulation can eliminate much of the uncertainty involved in LAN planning and management. However, non-validated simulations may produce subtly erroneous data.

An advantage of a properly verified and validated simulation is that it can be used to examine various network configurations that may include difficult to acquire hardware like National Security Agency (NSA) approved network encryption devices. In addition, simulation can be used to test various network configurations like modification of frame sizes to improve efficiency.

2.1 Definitions

Validation is the process which establishes the extent to which a model does (or does not) acceptably represent the phenomenon of interest (Hamilton, Nash and Pooch 1997). Broadly speaking, there are two classes of strategies which may be used to validate a model:

Axiomatic: The existence of a set of assumptions which describe the fundamental truths of the problem domain provides the basis for this approach. The validity of the model follows as a consequence of the application of rules of logical inference to the axioms to prove theorems. Ultimately, the model itself will be proven as a theorem. (Sargent 1987) referred to this method as *rationalism*. One advantage of this approach is that it establishes a model which describes causality.

Empirical: The operation of the model is considered to be a filter or a function that maps its inputs to outputs. The performance of the model is compared to our expectation (if the system to be modeled does not exist) or to historical data to determine the model's predictive power. Where historical data exist, we generally infer the adequacy of the model whenever the observed residual values are uncorrelated. If a correlation is observed, this suggests the existence of some additional variable(s) which must be included in the model to make it complete (Hamilton, Nash and Pooch 1997).

These two categorizations of strategies correspond roughly to what Spriet and Vansteenkiste refer to as *theory-driven* and *data-driven* validation, respectively (Spriet and Vansteenkiste 1982). For our testbed we are taking the empirical or data-driven approach.

The term *verification* describes the activity of insuring that a particular implementation faithfully satisfies the requirements of a specification over a given range of inputs. If it is known a priori that a model is valid, then historical data may be used with the understanding that demonstrating a correspondence between program outputs and the physical phenomena being modeled implies the verification of the simulation program in question (Hamilton, Nash and Pooch 1997).

2.2 Historical Validation

As noted Sargent, "If historical data exist (or if data are collected on a system for building or testing the model), part of the data is used to build the model and the remaining data are used to determine (test) whether the model behaves as the system does." This testing is done by driving the model with either distributions or traces." (Sargent 1996).

We plan to drive the simulation with traces and have been working with multiple simulators including OPNET, NS-2 and QUALNET.

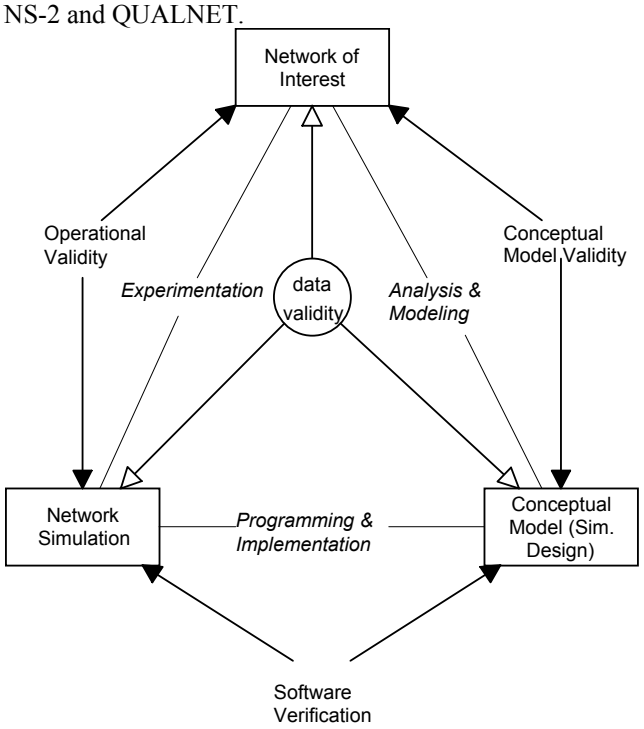

Figure 2. Simplified version of the modeling process (Sargent 1996), (Knepell and Arangno 1991).

Figure 2 is an excellent graphic depiction of the simulation verification and validation process detailing both the modeling and software engineering components of the process. When a model fails validation testing, problems may lie in the conceptual model, in the software implementation (computer model) or in both. Systematic analysis of both the software engineering process and the simulation modeling process is required to ensure an accurate representation.

In our case, we are confident of our data validity – we collect the data from actual field testing. We are concerned that our conceptual model correctly abstracts the unimportant details while capturing the attributes that drive the simulation. We verify the correctness of our implementation by running inputs from field tests into the simulation and determining if the outputs of the simulation are statistically indistinguishable from the outputs of the field test. Our procedure follows closely that illustrated in figure 2. Our objective is to have the predicted and actual results to differ only as a result of stochastic elements in the simulation. Once we have calibrated our simulation model we can then use it to evaluate the operational validity of proposed wireless redesigns of Army UAV ground elements.

3 CREATING SYNTHETIC WORKLOADS

When we drive network simulations with probability density functions (PDFs), we note that the simulations quickly go to steady state as shown in figure 3.

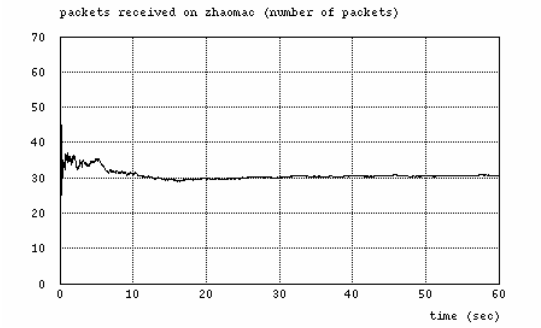

Figure 3. Network simulation reaching steady state.

By steady state we mean that the numbers of observations above and below the mean are approximately equal. Unfortunately, it is well known that network traffic is bursty. How do we bridge this gap? Two methods, scripting and synthetic workloads.

Scripting is well known and straightforward. We take actual field test data and have the simulator read the inputs. Thus, each station on the network follows a script. As has been demonstrated many times, scripting inputs to a correct simulation model will produce accurate "predictions." Of course the predictive power of this technique is somewhat lost since scripting in this manner requires a priori knowledge of the outputs. We use this technique to calibrate our model.

Once we begin using the simulation testbed to evaluate potential designs, we can use synthetic workloads to evaluate candidate designs. Work at Texas A&M University demonstrated that the injection of 25% (or less) actual traffic into a simulation was sufficient to produce statistically indistinguishable results from predicted outcomes measured against actual outcomes (Hamilton 96).

Because our actual test environment involves a small number of nodes and because we completely control the traffic on this test network, it is feasible to inject actual bursty traffic into the simulation testbed from known systems, i.e. the ground control station, the launcher and the headquarters element.

As Pooch and Wall observe, simulation is based on the scientific method (Pooch and Wall 1993). The four commonly accepted steps of the scientific method are:

1. Observation of the system.
2. Accounting for observed behavior
3. Prediction of future behavior based on the assumption that the modeling and understanding of the model are correct.
4. Comparison of the predicted behavior with the actual behavior.

Our methodology is consistent with the scientific method.

4 THE SIMULATION TESTBED

4.1 System Monitoring

A preliminary simulation has been run with our data in OPNET using the network setup in Figure 1.

Figure 4 is a screenshot of the simulation in OPNET. The UAV Client and Server have two traffic flows: one representing the control, and one representing the video stream. The traffic flow is a logical depiction, however in the simulation, the traffic follows the network path across the wireless link depicted by the basic service set (BSS) identifier 1. The ASA in this model is a Cisco Router, configured with similar IP addresses as the actual ASAs used to collect the data.

In order to create a valid simulation, we need a representation of the traffic coming into the network. We used Wireshark (formerly Ethereal) to capture the traffic coming off of the sender workstation, and saved the summary statistics for this capture. Once the traffic flow is created in OPNET based on these statistics, it is replayed when a simulation is run, and statistics can be gathered to verify that messages are being sent or dropped in a realistic way. The traffic captured from Wireshark is after encryption and before it is sent across the wireless bridge.

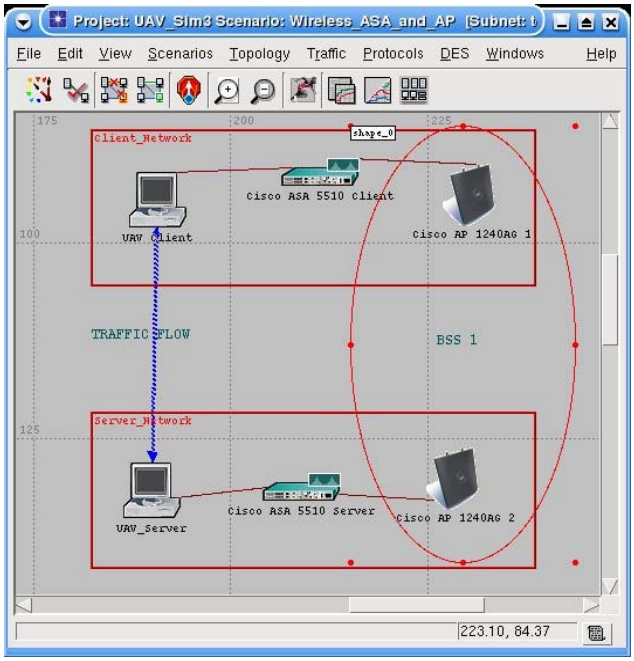

Figure 4. OPNET Simulation Mode

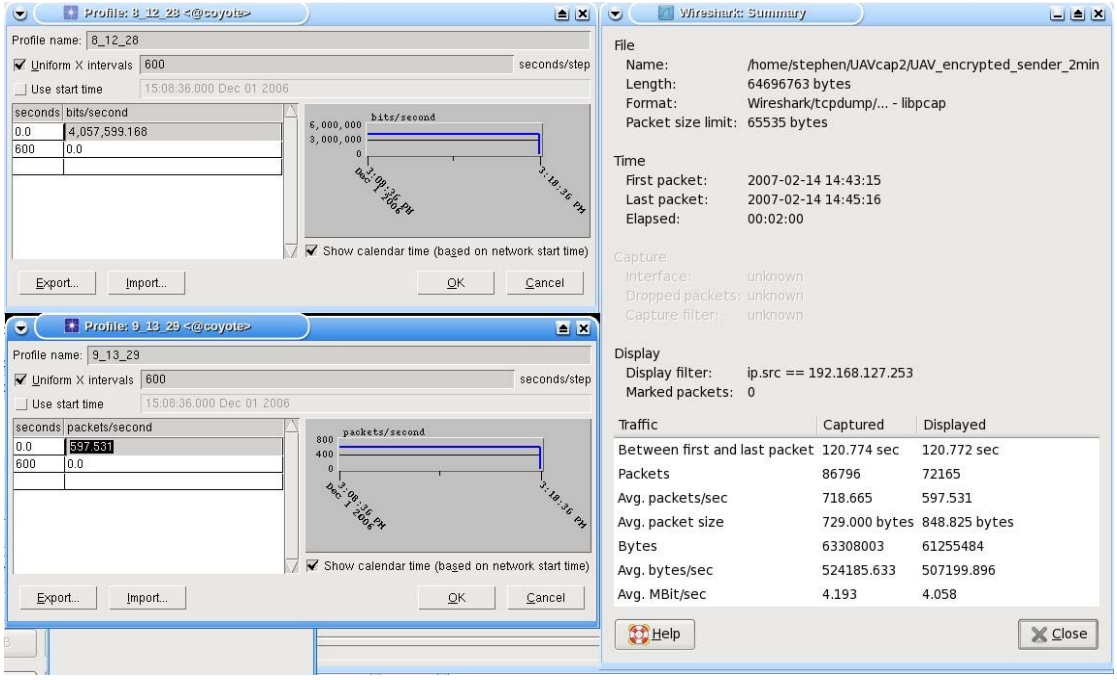

Figure 5. Wireshark traffic capture.

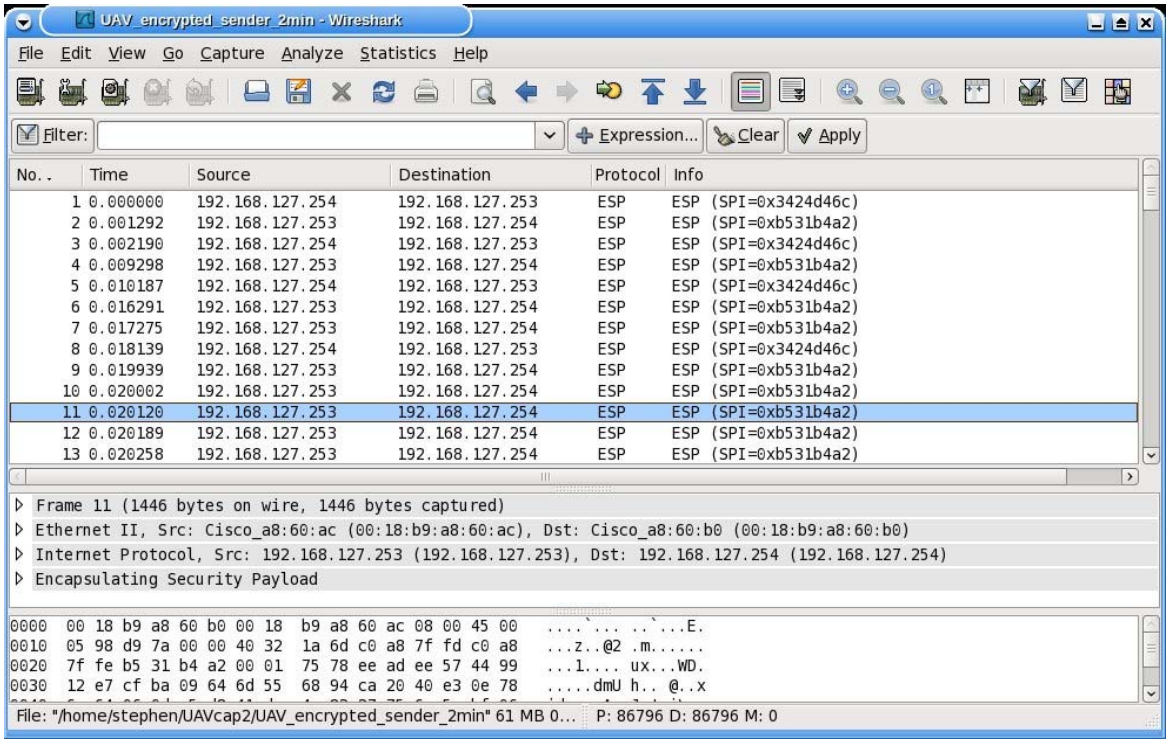

Figure 6. Encrypted packet displayed in Wireshark.

There are two flows of traffic utilized by the UAV: the traffic that controls the UAV and the video captured from the UAV. Since there are two flows, the packet capture is filtered to represent each flow.

One concern with making the UAV ground control station wireless is the security, since wireless transmissions cannot be physically controlled. Currently, the only devices approved to transmit secret classified data, like the UAV video stream, are classified by the NSA as Type I encryption devices. The availability and control of these devices make it difficult to acquire and test them with the wireless UAV setup. Therefore, given the correct specifications of latency and bandwidth overhead that these devices will incur, they can be simulated before going through the acquisition process. In the experiment we used the Cisco ASA, because it meets the cryptographic encryption requirements of Type I, and is available off the shelf. Using the Opnet Netwars model of the KG-175 Taclane or the Motorola NES, the simulation can easily be modified to measure how these devices will affect latency and bandwidth. These two devices encrypt at the IP layer, and are currently approved by the NSA to transmit classified data on an unclassified medium. The choice of which encryption device to use will depend on the availability of these devices for the UAV unit, and could potentially be influenced by the performance of each one in the simulation.

Figure 6 shows the encrypted traffic filtered to display the summary information on the video stream. The captured column in Wireshark displays the summary for the entire capture, while the Displayed column shows the summary on the packets sent, which we filtered out by source IP address. Since the stream was constant, we created the traffic flow in the simulator by taking the average bytes per second (converting it to bits), and average packets per second. These two metrics gathered from our captured data allows the simulator to create traffic similar in packet size and speed to the actual traffic.

5 CONCLUSIONS AND FUTURE WORK

Using special additions to the 802.11g wireless model in OPNET, we will be able to simulate transmission of UAV data over a wireless network. We are now implementing this model in both Qualnet and NS-2. Once these simulations are complete and verified, they can be used to test setups that would be hard to test in the real world. Also, equipment can be tested in the simulation before it is actually bought to help determine if the purchase is worthwhile. The UAV simulations will help to further our goal of making UAV command stations more mobile.

At Auburn University we believe that high fidelity verified and validated modeling and simulation has a critical role to play in all aspects of the system lifecycle. We are using modeling and simulation to support system design, analysis of alternatives and initial testing. We do not claim that modeling and simulation can ever replace operational testing. However, when a simulation is calibrated with and validated against actual test results, the consequent development of a high fidelity test bed can greatly increase the number of test runs that can be evaluated.

The whole point of setting up this simulation testbed is to empirically validate the simulations against historical data. The operation of the model is considered to be a filter or a function that maps its inputs to outputs. The performance of the model will be compared to our expectation (if the system to be modeled does not exist) or to historical data to determine the model's predictive power.

ACKNOWLEDGMENTS

This work has been partially funded by the Unmanned Systems Initiative through the U.S. Army's Aviation and Missile Research, Development and Engineering Center (AMRDEC) at Redstone Aresenal.

REFERENCES

Hamilton, J. A., Jr., 1996. *Multilevel Simulation of Discrete Network Models*, Texas A&M University, College Station, Tex.

Hamilton, J. A., Jr., D. A. Nash, and U.W. Pooch. 1997. *Distributed Simulation*. Boca Raton, Fla.: CRC Press.

Kneppel, P. L and D. C. Arangno. 1993 *Simulation Validation, A Confidence Assessment Methodology*, Los Alamitos, Calif: IEEE Computer Society Press.

Pooch, U.W. and J. A. Wall, *Discrete Event Simulation*, Boca Raton, Fla.: CRC Press.

Sargent, R. G. 1987. An overview of verification and validation of simulation models. In *Proceedings of the 1987 Winter Simulation Conference,* 33-39. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sargent, R. G. 1991. Simulation Model Verification and Validation. In *Proceedings of the 1991 Winter Simulation Conference,* 37-47. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sargent, R .G. 1996. Verifying and validating of simulation models. In *Proceedings of the 1996 Winter Simulation Conference,* 55-64. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Spriet, J. A. and G. C. Vansteenkiste. 1982. *Computer-aided Modeling and Simulation*, London: Academic Press.

AUTHOR BIOGRAPHIES

Major Stephen S. Hamilton received a Bachelor of Science degree in Computer Science from the United States Military Academy in West Point, NY in 1998. He is currently attending Auburn University in Auburn, AL, in pur-

suit of his Masters degree in Software Engineering as a fully-funded Army student. Major Hamilton's previous assignments include: G6 automation officer, 13th Sustainment Command (Expeditionary); Company Commander, Alpha Company, 57th Signal Battalion; S3 Automation Officer, 57th Signal Battalion; G6 Automation Officer, 3d Infantry Division; Platoon Leader, 123d Signal Battalion. He previously published "Poor Man's Digitization of the Battlefield" Army Communicator, Summer 2004. His current research includes information assurance and web enabled database applications. Major Stephen Hamilton is a member of the IEEE and Association of Computing Machinery (ACM).

Lieutenant Colonel Timothy Schmoyer is an Assistant Professor of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY. Colonel Schmoyer has a B.S. in Electrical Engineering from the Worcester Polytechnic Institute, and an M.S. in Electrical Engineering from the Air Force Institute of Technology, 1997. LTC Schmoyer was commissioned in the Signal Corps and later single-tracked in the Information Systems Engineer career field. Assignments include platoon leader and XO in the 1st Signal Battalion, company commander in the 112th Signal Battalion (Special Operations) (Airborne), communications officer for the 3rd Special Forces Battalion, 7th Special Forces Group, and architecture engineer for the Combined Forces Command/United States Forces Korea C/J6. He was deployed with the 1st Signal Battalion during Operations Desert Shield/Storm, 112th Signal Battalion during Operation Uphold Democracy and 3/7th Special Forces Battalion during Operation Safe Border/United Nations Military Observer Mission-Ecuador and Peru.

John A. "Drew" Hamilton Jr., Ph.D., is an associate professor of computer science and software engineering at Auburn University and director of Auburn University's Information Assurance Laboratory. Prior to his retirement from the U.S. Army, he served as the first director of the Joint Forces Program Office, on the electrical engineering and computer science faculty of the U.S. Military Academy, as well as chief of the Ada Joint Program Office. He has a B.A. in journalism from Texas Tech University, an M.S. in systems management from the University of Southern California, an M.S. in computer science from Vanderbilt University, and a Ph.D. in computer science from Texas A&M University. Dr. Hamilton is currently the President of the Society for Modeling and Simulation (SCS), International and the Secretary-Treasurer of the ACM Special Interest Group in Simulation (SIGSIM).