

Hardware and Software Integration to Support Real-Time Space-Link Emulation

Robert Murawski¹, Kul Bhasin², David Bittner³,
Aaron Sweet⁴, Rachel Coulter⁵, and Devin Schwab
NASA Glenn Research Center
Cleveland, OH 44107

Abstract—Prior to operational use, communications hardware and software must be thoroughly tested and verified. In space-link communications, field testing equipment can be prohibitively expensive and cannot test to non-ideal situations. In this paper, we show how software and hardware emulation tools can be used to accurately model the characteristics of a satellite communication channel in a lab environment. We describe some of the challenges associated with developing an emulation lab and present results to demonstrate the channel modeling. We then show how network emulation software can be used to extend a hardware emulation model without requiring additional network and channel simulation hardware.

I. INTRODUCTION

Software and protocols to be used in space-link communication networks must be thoroughly verified and validated prior to operational use. Testing products using traditional wireless networks is generally not sufficient as the problems associated with space-link communications are significantly different than that of traditional terrestrial-based wireless communications. Space link communications are typically characterized by very long delays over asymmetrical and unidirectional communications paths. Thus, careful testing of such protocols and software is necessitated.

Testing such equipment in field equipment is generally not a viable solution due to the associated costs. Therefore, many developers rely on either analytical studies or network simulations to validate their products. Analytical studies generally rely on simplifications and assumptions or and often do not consider the randomness that characterize wireless communication channels. Network simulation software can be used to more accurately model these characteristics, however, the specialized models developed for such simulations may not be high enough fidelity to accurately test a real system.

These issues have led researchers to develop a third alternative: network emulation. There are two chief differences between simulation and emulation. First, network emulation models are of high enough fidelity that they are capable of interacting with real hardware components. Second, due to the interaction with hardware components, network emulation must be performed in real-time. The goal of emulation is that these hardware components do not realize that they are communicating with network emulation software. Thus, the performance of the software and protocols can be observed as if they were utilizing a space-link without the associated cost and complexity of utilizing real systems.

The use of network emulation in wireless ad-hoc networking can be found in related research [1], [2], [3]. Again, many of the complexities of network emulation are amplified when emulating space-link communications. Therefore, to accurately emulate, for

example, a wireless satellite communication link, we must develop a specialized network emulation environment specially tailored to the requirements of space-link communications.

The specific case of space-link emulation has been researched as well [4]. In this case, the authors have developed a real-time space-link emulation environment. The basis of this emulation is the Elaboration Unit, a software emulation tool which models the link characteristics of a satellite communication link. One of the goals of our network emulation testbed is the capability to test not only software components and networking protocols, but to test hardware components such as satellite modems as well. Therefore, utilizing a purely software based network emulation solution to model the space-link communication characteristics is not sufficient in our case.

In this research, we have developed a space-link emulation testbed which utilizes a combination of hardware and software tools to accurately model the characteristics of satellite communications. Our goal is to have a space-link emulation testbed flexible enough that either hardware or software products can be tested over an emulated wireless link. For hardware components, such as satellite modems, we utilize a hardware channel simulator to accurately model the link characteristics of the wireless channel. Software components can be tested as well as their data is routed through satellite modems with the associated delays and error rates of the wireless channel.

II. SYSTEM MODEL

In this paper, we utilize multiple hardware and software simulation and emulation tools to accurately model a space communication link to a low earth orbit (LEO) satellite. Communication to LEO satellites can typically take two forms: direct communication and bent-pipe relay communications. Bent-pipe communication links, such as communicating through a Tracking and Data Relay Satellite (TDRS) are used to increase the duration of a pass as line of sight (LoS) to low orbit satellites can be limited from ground based systems due to terrain and the curvature of the Earth. In this paper, we first emulate a direct communication channel between a ground site and a LEO satellite using a hardware channel simulator. We then show how, by utilizing software emulation tools, these emulation scenarios can be expanded without the added cost of acquiring more channel simulation hardware.

A. Direct Channel Emulation

The direct communication link is modeled using the RT-Logic [6] T400CS hardware channel simulator. This device accepts intermediate frequency (IF) analog signals at 70 MHz and modifies the signal based on the desired wireless channel characteristics. To model the bidirectional direct communication channel, two channel simulation cards are required to emulate both the forward and return wireless channels as shown in Figure 1.

The RT-Logic channel simulator does not perform the link budget estimations required to determine what attenuation, Doppler Shift,

¹Corresponding Author, Robert Murawski currently works for Vantage Partners, LLC. Email: robert.w.murawski@nasa.gov

²Email: kul.b.bhasin@nasa.gov

³Email: david.m.bittner@nasa.gov

⁴Aaron Sweet currently works for Qualcomm, Inc

⁵Rachel Coulter currently works for Texas Instruments

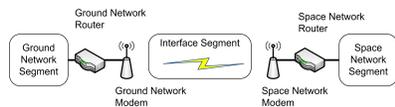


Fig. 1. Network Emulation Testbed (Direct Communication)

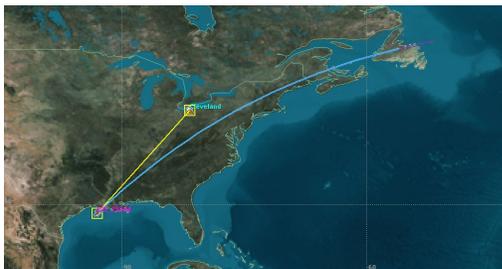


Fig. 4. STK Scenario for Direct Channel Emulation



Fig. 5. STK Scenario for Bent-Pipe Channel Emulation (3d View)

etc should be applied to the emulated wireless channel. To generate this information, we develop simulation models within the Satellite Tool Kit (STK) [7] developed by AGI. Based on the simulation parameters, STK provides orbital dynamics calculates to determine the location of the satellites during the simulation, as well as link budget analysis based on the simulated hardware components (i.e. antennas, amplifiers, modems, etc). A view of the STK simulation used to model the direct channel communication path is shown in Figures 4 and 5. In addition to the RT-Logic channel simulator, network hardware (i.e. routers and switches) are required to direct data traffic through the emulated wireless channel.

B. Bent-pipe Communication Emulation

To model a bent-pipe communication link, two additional wireless channel would be required for forward and return communications to the relay satellite as shown in Figure 2. In our emulation lab, the T400CS channel simulator only has two channel cards available, and thus, the full bi-directional bent-pipe communication link cannot be emulated with this hardware alone. To model the additional communication links between the ground site to the relay satellite, we utilize EXata [5], which is a software network emulation toolset. As EXata cannot perform the high-fidelity link budget analysis or orbital dynamics required for this model, we again utilize STK to perform

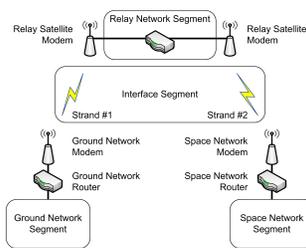


Fig. 2. Network Emulation Testbed (Bent-pipe Communication)

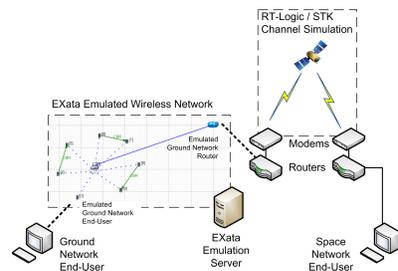


Fig. 3. EXata Enhanced Network Emulation Testbed

the required calculations. Our results show that this joint emulation is possible, however, with the current hardware and interface tools, the co-emulation leads to excessive delays through the EXata emulation model. In future work, the performance of the co-simulation software could potentially be improved to allow real-time co-emulation of these scenarios.

C. Network Emulation Testbed

In this section, we define the different components of the two communication link models: direct communication and bent-pipe communication. In each model, there is a ground segment and a space segment. However, to support the additional EXata emulation, the implementation of the ground segment is slightly different for the bent-pipe communication model, as described in the following sections. The bent-pipe communication model has an additional interface segment which represents the TDRS relay communication. The hardware, software, and network configurations used for each segment are defined in the following sections:

1) *Ground Network Segment (Direct and Bent-Pipe Models)*: The ground network segment consists of a gigabit Ethernet switch and a router. All components on the ground network segment are within their own, unique IP subnetwork.

For the direct communication model (Figure 1), the router is a physical device which connects to a serial satellite modem to facilitate communication over the simulated wireless channel. The router in the bent-pipe communication model is a virtual device within the EXata scenario model. Equipment communicating through the bent-pipe network must communicate through the EXata emulation scenario, which models the propagation delay and bit error rate (BER) as estimated by STK. A more detailed view of how EXata is integrated into the end-to-end emulation model is shown in Figure 3.

2) *Space Network Segment (Direct and Bent-Pipe Models)*: The space network segment consists of a gigabit Ethernet switch and a router, similar to the ground network segment. For both the direct and bent-pipe communication models, the space network segment the router is a physical device which connects to the serial satellite modem for communication over the simulated wireless channel. All components on the space network segment are within their own, unique IP subnetwork.

3) *Relay Network Segment (Bent-Pipe Model)*: The relay network segment is only utilized in the bent-pipe communication model (see Figure 2). Typically, the relay between the ground and space segments is handled by a Tracking and Data Relay Satellite (TDRS). Due to the much higher altitude of the relay satellite, the duration of a line-of-sight pass is extended at the cost of higher delays and more propagation losses.

In the bent-pipe communication model, the relay segment consists only of a router and a satellite modem. For communications with the ground segment, the relay network segment router routes traffic to the EXata emulation server which contains a virtual instantiation

of the router. Packets are then transmitted over the EXata simulation model using the STK link budget calculations. Packets destined for the space network segment are sent to a serial satellite modem for communication over the simulated communication channel.

4) *Interface Segment (Direct and Bent-Pipe Models)*: The interface segment is where the estimated channel characteristics are emulated. This includes the propagation delay, propagation loss, bit error rate (BER), Doppler shift, and noise that would be expected on a real space-link communication channel. In Figure 2 two individual strands of the interface segment are identified. The link budget estimations for each strand are computed through the Satellite Tool Kit (STK) [7] developed by AGI. Within STK, we have modeled three platforms: the ground site, TDRS relay satellite, and the International Space Station (ISS). The transmitters and receivers on each of these platforms are modeled including the signal gains (amplifiers, antennas, etc) as well as signal losses (cabling, insertion, etc). STK also models the orbital dynamics of the satellites to determine the transmission distances as well as the propagation delay and signal attenuation between the various network segments.

The first wireless strand is emulated using the EXata network emulation toolset. When packets arrive at the EXata emulation server, they are converted from Ethernet packets to data structures formatted for use within the EXata protocol models. Within the EXata emulation scenario, the packets are routed over a generic wireless communication link. The channel characteristics of the emulated communication link are provided from the STK simulation scenario through a custom EXata physical layer model developed at NASA Glenn Research Center. After packets are sent through the emulation space link, they are converted back into Ethernet packet and sent to the Relay Network Segment's router.

The second wireless strand is emulated using the RT-Logic T400CS channel simulator. In this case, Ethernet data packets arrive at either the relay or space segment routers and are converted to a serial data stream for transmission. The serial data is then sent to a satellite modem which modulates and encodes the digital data into an analog signal suitable for wireless transmission. The 70MHz IF signal is then transmitted through the RT-Logic channel simulator which modifies the analog signal based on the desired channel characteristics as determined by STK. For this wireless strand, it is important to note that the signal attenuation as determined by the link budget analysis cannot be directly imported into the channel simulator. This is because much of the field hardware (i.e. antennas, amplifiers, etc) are not present in our emulation lab. Therefore, the gains and losses associated with this equipment must be quantified carefully and compensated for using gain offsets within the RT-Logic / STK integration tool.

III. DIRECT COMMUNICATION SCENARIO RESULTS

Let us assume that we are modeling a direct communications link between a ground station located at NASA Glenn Research Center in Cleveland, Ohio and an orbiting LEO satellite. In this scenario, a generic LEO satellite at approximately 1000 meter altitude was selected such that a full pass over the Cleveland based ground station takes approximately 18 minutes. In our STK scenario, the following parameters are used for the transmitter and receiver for both the forward and return link:

- **Transmitter**

- Antenna Gain: 41.038dBi
- Cable Loss: 2dB
- HPA Gain: 40dB
- Modem Output Pwr: 20dBm

- Center Freq.: 14.5GHz

- **Receiver**

- Antenna Gain: 41.038dBi
- Cable Loss: 2dB
- LNA Gain: 20dB

A. Signal Attenuation Emulation

Initially, the propagation distance between the LEO satellite and the ground station is approximately 3574.5 kilometers. At this distance, STK estimates that the free space attenuation loss of a signal at 14.5 GHz would be approximately -186.74 dB. In addition to free space loss, STK estimates atmospheric losses, rain fading, and scintillation loss. All of these losses are combined into an estimated propagation loss as shown in Figure 6:

Note that in Figure 6, free-space loss due to the distance between the satellite and ground site contributes most to the overall propagation loss. Also, at the start and end of the satellite pass, when the satellite is on the horizon with respect to the ground site, the scintillation and rain losses are excessive, which could result in lost communications as these propagation losses are directly reflected in the carrier power at the receiver as seen in Figure 7.

The wireless communication channel is modeled using the hardware channel simulator developed by RT-Logic. However, we are not simply emulating the propagation losses of the channel, but the entire end-to-end communication system. Thus, additional gains and losses which are not present in our emulation lab must be accounted for when the channel gain parameter is sent from STK to the channel simulator. The RT-Logic/STK plugin uses the carrier power at receiver (see Figure 7) as an input to the RT-Logic channel simulator. This value includes the EIRP of the transmission source, all modeled propagation losses, and the gain of the receiving antenna. This value, however, does *not* include the additional gain of the receiver low noise amplifier (LNA) and associated cabling and insertion losses between the LNA and the receiver modem. To account for this additional gain, these values are added the 'Gain Offset' parameter within the RT-Logic STK plugin.

After accounting for the additional LNA gain, the RT-Logic STK plugin will set the gain of the channel simulator to the receive signal strength as estimated by the STK simulation model. This would be sufficient to model the channel attenuation if the input to the RT-Logic channel simulator was exactly 0dB. In our emulation lab, there are several other parameters which contribute to a gain offset value which maps the STK estimated gain to the emulation lab equipment. These parameters are shown in Table I. To show how applying this gain offset results in an accurate modeling estimated link budget, we compare the observed channel power in the emulation lab to the estimated values as determined by STK.

In Figure 9 we see the observed receive signal power which is estimated in Table I. The observed channel power in this figure is the average of 50 sweeps by a spectrum analyzer. The observed gain of -53.47dBm is approximately 0.45dB less than what is estimated through STK.

B. Signal Delay Emulation

To test the channel delay through the emulated wireless channel, we send ICMP ping messages through the wireless channels using routers which are physically connected to each of the satellite modems. The ICMP ping message travels through the forward channel from the ground site to the LEO satellite and a ICMP response message is sent through the return channel. Thus, the round-trip-time (RTT) of the ping message will include the propagation delay from

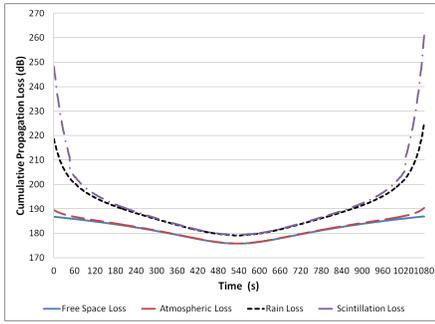


Fig. 6. Cumulative Propagation Loss for Direct Channel Model

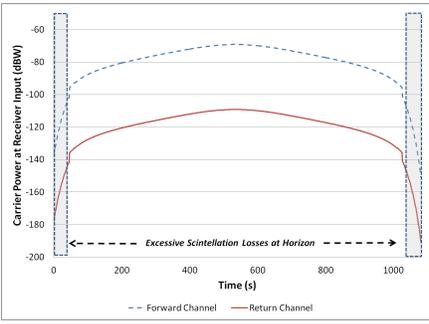


Fig. 7. Estimated Carrier Receive Power for Direct Channel Model

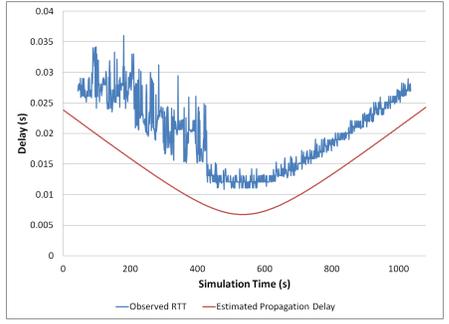


Fig. 8. Observed RTT compared with estimated propagation delay

TABLE I
LINK BUDGET CALCULATIONS IN STK MAPPED TO EMULATION LAB

STK Link Budget Calculations	
EIRP	69.04 dBW
Free-Space Loss	-185.82 dB
Atmospheric Loss	-0.91 dB
Rain Loss	-13.57 dB
Scintillation Loss	-2.80 dB
Receiver Gain	41.04 dB
Carrier Power at Receiver	-93.02 dBW
Carrier Power at Receiver	-63.02 dBm
Receive Signal Strength (w/ 20dB LNA)	-43.02 dBm
RT-Logic STK Plugin Offset Calculation	
Modem Output Power	25.0 dB
Pre-processor Gain	5.0 dB
dBW to dBm Conversion	30 dB
Cable / Insertion Losses	1.376 dB
Power Divider Losses	3.75 dB
LNA Gain	10.0 dB
Total Offset	75.126 dB
RT-Logic Calculations	
STK Estimation	-93.02 dB
Gain Offset	75.12 dB
Input to Channel Simulator	-17.9 dB
Hardware Calculations	
Modem Output Power	-25.0 dBm
Cable Losses	-1.376 dB
Pre-processor Gain	-5 dB
Power Divider Losses	-3.75 dB
Channel Simulator Gain	-17.9 dB
Expected Observed Channel Power	-53.02 dBm

both the forward and return channels. Furthermore, there are delays in addition to the propagation delay, such as queuing and processing delays in the routers, as well as processing and transmission delays in the modem. Therefore, we expect the observed RTT to be higher than the STK estimated propagation delay. However, the observed RTT should follow the same trend as is estimated by STK.

In Figure 8 we see the observed RTT through the emulated wireless channel over the entire 18-minute pass. As you can see, during the first 48 second, and for the final 44 seconds of the pass, there is no observed RTT reported. This is because, at those times, the scintillation loss and rain loss are so extreme, that the modems are not able to pass the ICMP ping messages. This excessive pathloss can be seen in Figure 6.

The observed RTT, as expected, is higher and more random than the propagation delay as estimated through STK. The additional delay is mainly due to the processing time in the modems for encoding and modulation of the digital data.

C. Additional Emulated Parameters

Two additional parameters can be emulated using the channel simulator and STK simulation software: Doppler Shift and Noise Generation.

Doppler Shift: Doppler Shift is the apparent shift in frequency due to the relative motion of the transmitter and receiver. In this direct communication example, the LEO satellite is orbiting around the earth and therefore is moving relative to the ground station. The shift in frequency due to the Doppler Effect is estimated within STK and can be modeled in real-time using the RT-Logic channel simulator.

In Figure 10 we show that the channel simulator can model the Doppler Effect. In this figure, we use the test-mode on the transmitting modem such that it is only transmitting a pure 70MHz carrier wave. Thus, we can accurately determine the center frequency of the initial carrier wave and compare it to that of the output carrier wave. In Figure 10, we have input a Doppler Shift of 46.4KHz to the channel simulator.

Noise Generation: Based on simulation parameters, STK estimates the noise at the receiver radio in degrees Kelvin. The RT-Logic channel simulator, however, reads the input noise in terms of power spectral density (dBm/Hz). To convert the STK estimated noise to power spectral density, we use the Boltzmann Constant:

$$N \left(\frac{dBm}{Hz} \right) = 10 \log_{10} \left(\frac{kT}{0.001} \right), \quad (1)$$

where k is the Boltzmann constant ($k = 1.3806503 \times 10^{-23} J/K$), T is the equivalent noise temperature in Kelvin as estimated by STK, and N is the power spectral density of the system noise.

The RT-Logic STK Plugin only allows a static value of system noise. In our current models, the system noise remains constant throughout the simulation lifetime. Thus, currently using a static system noise value is acceptable. In future work, if a dynamic system noise is required, further improvements to the STK plugin must be investigated.

In Figure 11, we show the output of the channel simulator for a generated noise floor of -120dBm/Hz. This channel power measurement is the average of 50 sweeps by our spectrum analyzer. Here, you can see that the power spectral density of the signal noise is approximately -124.87 dBm/Hz. The additional loss of approximately 4dB is due to the added losses of adding a power splitter to route the output signal to the spectrum analyzer.

IV. BENT-PIPE COMMUNICATION SCENARIO

To enhance our emulation lab capabilities, we have developed an integration tool which allows an EXata emulation scenario to utilize channel estimations from the STK software package. The bent-pipe

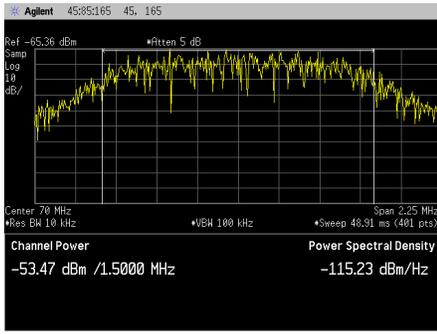


Fig. 9. Observed Channel Gain through Spectrum Analyzer

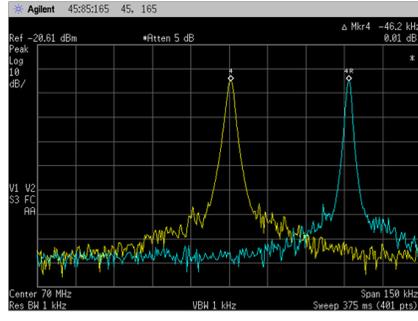


Fig. 10. Observed Frequency Shift due to Doppler Effect

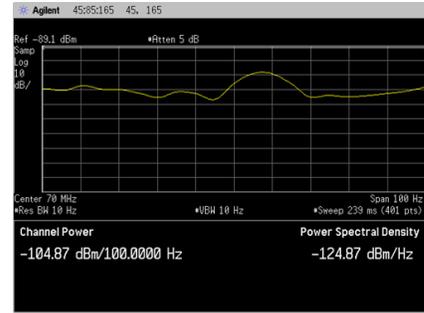


Fig. 11. Observed System Noise

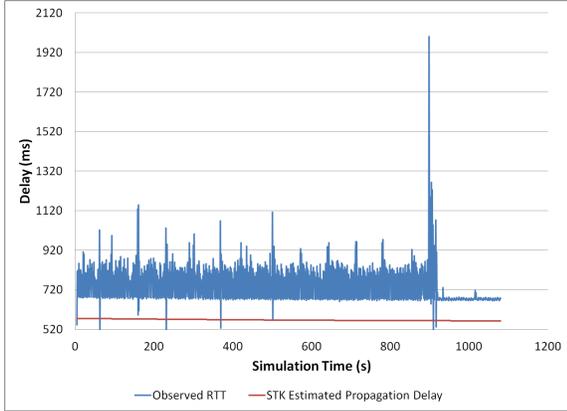


Fig. 12. Observed RTT versus Propagation Delay for Bent-pipe Communication Model

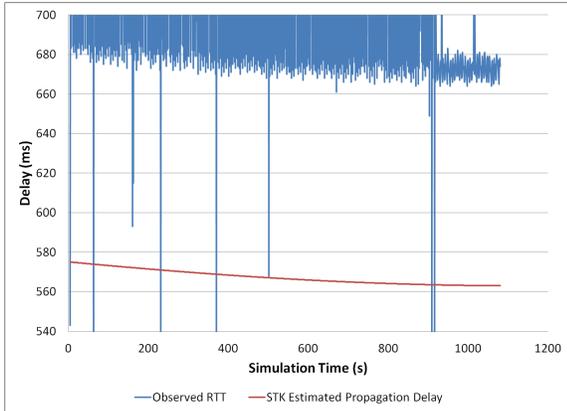


Fig. 13. Observed RTT for Bent-pipe Communication Model (detailed)

communication model defined in Section II-B is emulated using EXata to model the ground-to-relay satellite communications while the hardware channel simulator models the bidirectional link between the relay and LEO satellites.

Traffic flowing from the test ground hardware is first virtualized and then flows through the EXata emulation scenario. The BER and propagation delays of this link are updated through the EXata/STK integration software. With our current hardware and EXata/STK integration tool, the delays associated with this traffic virtualization result in excessive delays through the emulated network. The observed RTT through the full bent-pipe communication model can be seen in

Figure 12 and a detailed view of the delay can be seen in Figure 13. As you can see in these figures, the RTT through the emulated bent-pipe model follows the same trend as the delay associated in STK. However, there is a significant offset in the observed RTT.

The bent-pipe communication model results show that the full software and hardware emulation model is feasible. In future work, we intend to improve the performance of the EXata/STK integration software to reduce the communication overhead between these two software packages and utilize separate. This improvement, as well as running the software on more powerful workstations, should allow us to emulate more complex communication scenarios without the additional overhead of more expensive hardware channel simulators.

V. CONCLUSION

In this paper, we have demonstrated how hardware and software simulation tools can be combined to emulate a wireless satellite communication channel. Wireless channel emulation allows hardware and software to be tested at much lower cost and in non-ideal scenarios which are not possible through field testing. Furthermore, we have shown that through software emulation tools, hardware channel emulation can be extended virtually without the added cost of addition hardware simulators. The combination of hardware and software emulation tools is currently at a proof of concept stage, however, in future studies we will improve the performance of the custom developed software and demonstrate that this combination of hardware and software is realizable.

ACKNOWLEDGMENT

This work was funded by NASA Glen Research Center under the Space Communication and Navigation (SCaN) project. The authors would like to also thank Michael Fuentes and Zach Sims for their contributions to this research.

REFERENCES

- [1] Kubinsky, F.; Lazar, Z.; Varga, A.; Eged, B.; Telegdy, A.; , "Emulation of ad-hoc networks on IEEE 802.11," MIKON-2000, vol.2, no., pp.447-450 vol.2, 2000
- [2] Ivanic, N.; Rivera, B.; Adamson, B.; , "Mobile Ad Hoc Network emulation environment," MILCOM 2009. IEEE , vol., no., pp.1-6, 18-21 Oct. 2009
- [3] Giovanardi, A.; Mazzini, G.; , "Ad hoc routing protocols: emulation vs simulation," Wireless Communication Systems, 2005. 2nd International Symposium on , vol., no., pp. 140- 144, 5-7 Sept. 2005
- [4] Marchese, M.; Perrando, M.; , "A packet-switching satellite emulator: a proposal about architecture and implementation," ICC 2002, vol.5, no., pp. 3033- 3037 vol.5, 2002
- [5] Scalable Network Technologies (SNT), *EXata*, <http://www.scalable-networks.com>
- [6] RT-Logic, *T400 Channel Simulator*, <http://www.rlogic.com>
- [7] Analytical Graphics, Inc., *Satellite Tool Kit (STK)*, <http://www.agi.com>
- [8] SNT, *QualNet*, <http://www.scalable-networks.com>