JMEE: A SCALABLE FRAMEWORK FOR JTRS WAVEFORM MODELING & EVALUATION

Sheetalkumar Doshi,
Scalable Network Technologies Inc, Los Angeles CA

Unghee Lee,
Scalable Network Technologies Inc, Los Angeles CA

Rajive Bagrodia,
Scalable Network Technologies Inc, Los Angeles CA

Chris Burns,
SPAWAR Systems Center, San Diego CA

ABSTRACT

In this paper we present the architecture of a Scalable JTRS Modeling and Evaluation Environment (Scalable JMEE), which uses hardware-in-the-loop with real-time emulation to support a scalable and application-centric evaluation of the JTRS communication networks. Scalable JMEE includes a combination of live components, real-time emulations of communication protocols, together with high fidelity simulation models of the physical environment. Scalable JMEE includes the ability to incorporate accurate wireless environment effects due to node mobility, urban and rural terrain, weather, fading and shadowing. The environment can support the direct integration of various third party real world applications that include streaming video, Voice over IP (VoIP) secure file transfer and web browsers. Scalable JMEE has been used to test and analyze JTRS networks with 500+ radios in diverse operational scenarios. Results from such analysis will help to accurately characterize the application mix that can be reliably supported by given deployments of future and current force communication assets operating under realistic traffic loadings.

I. INTRODUCTION

A number of DoD programs in the Army, Navy, and Air Force are being deployed to provide advanced C^{4}ISR capabilities for network-centric warfare. These include the Joint Tactical Radio System (JTRS) program that is an essential component of the network-centric operations paradigm and the Navy's ForceNet initiative that aims to develop future Navy operational capabilities that will leverage the JTRS capabilities. Given the significant role for the JTRS networks in future military operations, it is imperative that the Services have the capability to analyze current and future JTRS systems, their functions and interoperability issues with legacy networks, on a large scale with an appropriate mix of end user applications.

The next generation of *distributed C^{4}ISR applications* must not only be resilient to fundamental characteristics of a wireless network (e.g., transient packet and connectivity loss due to terrain or environmental factors), but must also be able to interoperate with other applications that share a common underlying communication network. It is important that analysts be able to *incorporate the dynamics of the underlying network when developing a distributed application and to validate design decisions.* The ability to perform such application-centric analysis in advance of widespread availability of the physical devices and networks provides significant cost benefits in testing, training, and related contexts.

Our approach for conducting application-centric analysis is to leverage state-of-the-art modeling and simulation technologies to develop *realistic* environments with hardware- and software-in-the-loop interfaces for real world applications and networks. Such an environment provides a viable approach for end-to-end performance prediction of net-centric operations in an efficient, yet scalable manner. For instance, the combination of *operational* applications with models of the underlying networks offers a promising solution to assess the impact of network dynamics on the performance of *actual applications* that are already being used by the military. The network models mimic the behavior of the underlying networks and provide analysts with a *configurable* environment to conduct repeatable experiments with the traffic profile exactly as it might be generated from the corresponding applications themselves. Similarly, integrating a small *physical* network test bed with a much larger *simulated* network allows an analyst to incorporate realistic physical effects while simultaneously scaling the study to networks much

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

larger than those that are physically accessible to the analyst.

We present such an environment for application-centric, interoperability and scalability analyses of JTRS waveforms, which we term as Scalable JTRS Modeling and Evaluation Environment (Scalable JMEE). The framework employs a combination of *live* components, emulations of communication protocols together with high fidelity simulation models of the physical environment to provide a *configurable* environment for conducting repeatable experiments. JMEE contains high fidelity models for realistic and accurate wireless signal propagation effects due to geographical terrain as well as obstructions due to foliage and buildings in urban environments.

In the Section III we present an overview of JMEE and its architecture details. In Section IV, we illustrate JMEE's benefits by conducting application-centric analysis of radio networks configured in the Scalable JMEE running a 'WNW-like' waveform. Section V provides the conclusion and an outline of future work planned for the tool.

II. RELATED WORK

Network simulators like ns-2[7], OPNET[8] offer an environment for high fidelity model development of JTRS waveforms and the ability to conduct small-scale stochastic traffic based analysis of simulated networks running the high fidelity waveforms. Interfacing live systems and applications via hardware-in-the-loop requires that the execution speed of the simulation matches real time. Furthermore, conducting large-scale network analysis with hardware-in-the-loop operation requires that the real time execution constraint be met while simultaneously allowing scalability with high fidelity waveform models. Commonly used techniques by analysts using these simulators for addressing this challenge are a) abstraction, and b) extrapolation. Abstraction involves reducing the complexity of the waveform model or wireless communication effects by ignoring minute details that are seemingly ancillary, so the simulator can scale with number of nodes as well as maintain real-time execution. However, this technique affects the accuracy of the analysis results, since the abstractions can cause deviations from the intended network operation. An example is abstracting out detailed wireless communication effects in order to reduce complexity, maintain real-time execution and achieve scalability. The accuracy of the results obtained from a simulation that has such an abstraction is questionable, since the effects that are abstracted would have led to a different chain of events at the propagation layer in the simulation, possibly leading to totally different results; the impact of control overhead at the routing layer that affects the performance of data transmissions at the MAC layer is a well-known phenomenon. The technique of extrapolation involves deriving results of large-scale networks from small-scale analysis results. This technique is based on the assumption that the waveform behavior exhibited in a small-scale network will not change as the network size increases. This however may not be true, since many protocols are known to have a different behavior when the network size increases beyond a certain limit. Hence applying extrapolation to derive large-scale analysis results from small-scale network analysis may lead to inaccuracies in the prediction of performance of large-scale networks.

We base JMEE on the QualNet network simulator [3] which has already demonstrated the ability to run high-fidelity models at all layers of the network stack for a large-scale mobile, wireless network consisting of thousands of radios with realistic wireless communication effects, and at the same time maintain real-time execution by leveraging multi-core, parallel processing architectures. We describe the architecture of JMEE in Section III and discuss how it achieves the goal of scalability with high fidelity models while maintaining real-time execution.

III. ARCHITECTURE OF JMEE

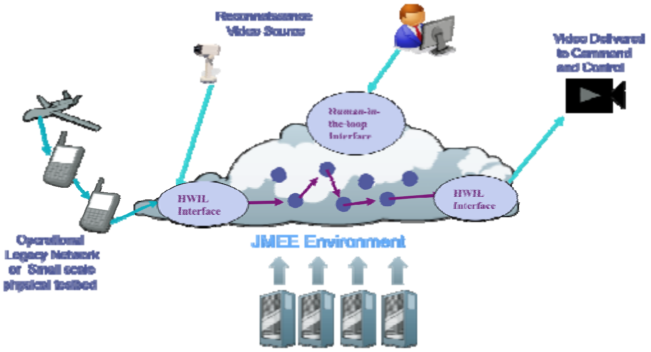

Figure 1: Overview of JMEE

As indicated in Figure 1, JMEE offers a hardware-in-the-loop interface to real world networks and applications through which they can interact with the scenario configured in JMEE. JMEE also offers a human-in-the-loop interface for the analyst to interact with the simulated scenario. The configured scenario in JMEE has the ability to execute on par with real time, so that physical networks and applications can seamlessly interface with it. This real-time execution can be achieved for network sizes ranging from a few hundred to thousands of nodes.

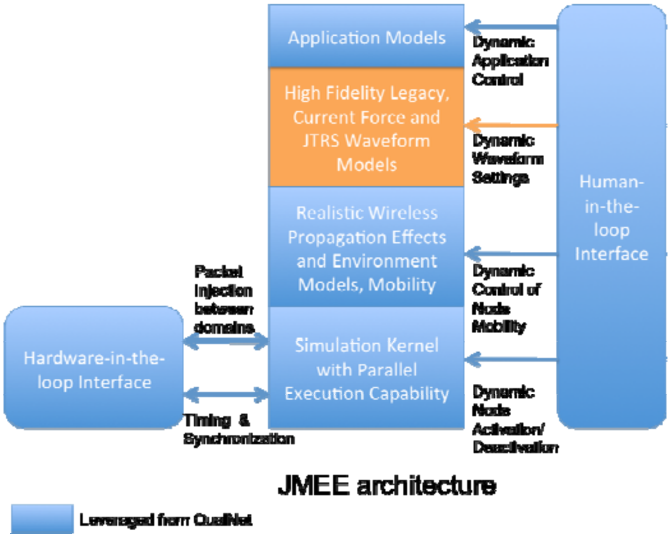

Figure 2: Architecture Overview of JMEE

Figure 2 provides an overview of the architecture of JMEE. As indicated, JMEE leverages many of the QualNet network simulator's capabilities and features. In this section, we briefly describe each of the preceding components.

1. High Fidelity Legacy, Current Force and JTRS waveform models

JMEE contains high fidelity models for legacy, current force and future force JTRS waveforms. High fidelity models for legacy and current force systems include models for Link-11, Link-16, EPLRS and SINCGARS radios. In addition, detailed representations of all protocols that comprise the network services, MI and MDL layers of the Wideband Network Waveform (WNW) were developed and successfully validated against shared code models. High fidelity models for future waveforms are also envisaged for the JMEE.

2. High Fidelity Application Models

JMEE inherits high fidelity stochastic models of various applications like FTP, VoIP, HTTP, streaming video, to name a few. While one of the main objectives of JMEE is to allow analysis with real applications, including models of these applications is useful when careful control of the parameters of these applications is required. For example, it is useful to configure stochastic models of applications to serve as interfering traffic in the simulated network while conducting application centric analysis with real applications.

3. Realistic Wireless propagation and Environment Models

JMEE leverages already available high fidelity wireless communication effects in QualNet. These include effects that occur due to signal propagation, geographical terrain, physical obstructions and weather. QualNet accurately models a range of physical environmental effects as indicated in Table 2. QualNet also has the capability to include detailed terrain models in the simulation. A study done by Hsu et al. [2] compared performance results from a real world ad hoc wireless network deployment to the results obtained from a model of the network in QualNet and concluded that QualNet modeled the deployment scenario with remarkable accuracy, thus validating the ability of QualNet to model realistic wireless environmental effects.

Table 1: Simulation Models for Wireless Environmental Effects in QualNet leveraged in JMEE

Path Loss	Free Space, 2-Ray, Urban Propagation models, Weather effects
Shadowing	Lognormal
Fading	Ricean, Rayleigh
Terrain	ITM, DEM, DTED, TIREM, CTDB7, CTDB8

4. Simulation Kernel with Parallel Execution Capability

JMEE must ensure that the simulation execution time is in synchronization with real-time so that the concept of time is the same for both the simulated network nodes and the physical networks and applications. This requires a simulation architecture that can efficiently process events generated by the high fidelity models to enable the advancement of simulation time clock in synchrony with real-time. Maintaining this synchronization imposes a significant constraint on the event processing capacity of the simulator. In case of scenarios that involve a large number of nodes, perhaps numbering in the hundreds or thousands, the number of events generated with high fidelity protocol models is enormous. A simulator running on a single processor will typically be overwhelmed in such a case will not be able to satisfy the constraints of real-time execution on a continuing basis. Under these conditions, the simulation execution time will not match up to real-time speed and the network operation shall report errors. The simulator hence must support parallel execution on multiple processors or multiple computing nodes to increase its capacity to process events. In this regard, JMEE leverages the capability of the QualNet simulation kernel for parallel execution. This means the large number of events that are generated in scenarios containing hundreds of nodes running high fidelity JTRS waveform models are processed in an efficient manner by partitioning the simulation and running partitions in parallel on dual core, quad-core processor machines or on parallel computing clusters. This capability ensures that these simulation events are processed such that the clock time required to simulate every event is less than some a priori bound (termed as transactional real-time simulation [1]). Thus achieving transactional real-time simulation with JMEE kernel's parallel execution feature satisfies the much desired requirement of scalability with high fidelity JTRS waveforms while still maintaining real-time execution.

5. Hardware-in-the-loop interface

The hardware-in-the-loop interface in JMEE is responsible for capturing packets from physical applications and operational JTRS networks and inserting these packets into the JMEE simulation and vice versa. The function of this interface is to ensure timing synchronization between the physical and simulation domains, and to ensure that the packet semantics are maintained as packets are injected between domains.

JMEE inherits its hardware-in-the-loop interface from QualNet. QualNet has an efficient hardware in the loop interface called IPNE to couple physical network and applications with the simulated network. IPNE controls and advances the simulation clock to keep in step with the real time clock, bringing about timing synchronization. IPNE captures packets from operational networks and applications using the *libpcap* [9] packet capture library functions, processes the captured packets to ensure that packet formats and byte orders in the headers match the semantics followed in the simulation domain and inserts these packets in the appropriate layer of the simulated protocol stack. IPNE also identifies packets in the simulation that need to be injected into the physical domain, does appropriate processing for ensuring that the packet and header contents conform to semantics followed in the physical domain, and inject them into the physical domain using the *libnet* [10] packet injection library functions.

6. Human-in-the-loop interface

A human-in-the-loop interface for JMEE is required so that the network analyst can conduct what-if analysis in a dynamic manner and such an interface can be employed in the advance use case of network operator training. JMEE inherits the human-in-the-loop capabilities of QualNet. In QualNet, this interface is accessible via its graphical user interface, and it allows the user to dynamically activate/deactivate nodes, control node mobility and protocol stack settings, assign application traffic priorities and initiate and terminate background application traffic flows. The interaction between the Human-in-the-loop interface and the simulated scenario occurs via a set of dynamic API's in the QualNet kernel that expose the scenario and waveform properties that can be dynamically set during scenario execution. This framework provides a rich toolset for 'what-if' analysis, and coupled with the hardware-in-the-loop interface for integrating real application traffic with the JMEE simulated scenario, allows the user to directly see the impact of changes made via the human-in-the-loop interface on the real world application performance.

IV. APPLICATION-CENTRIC ANALYSIS USING JMEE

In this section, we describe the experiments conducted for application-centric analysis of several network configurations running a waveform similar to WNW in JMEE. The objective is to highlight the capabilities of JMEE to conduct such analysis, rather than presenting detailed performance results for a particular waveform.

Real world applications were interfaced with the JMEE simulated scenarios using the IPNE hardware-in-the-loop interface. These applications included the following third party open source applications:

1. Network performance measurement applications: iperf [4] and ping
2. VLC[5], a streaming video application
3. Linphone [11], a VoIP application
4. Ftp (File transfer protocol) client and server

Specifically, we focused on the effect of hop count, number of nodes in a subnet, and mobility on the performance of real world applications. Such experiments when conducted on a physical testbeds are extremely resource-intensive (in terms of both human and equipment costs) and require extensive amount of time to gather results. Using JMEE though, these experiments can be easily configured with minimal resource requirements and require a few hours to gather and analyze the results. Figure 3 depicts the configuration of experiments.

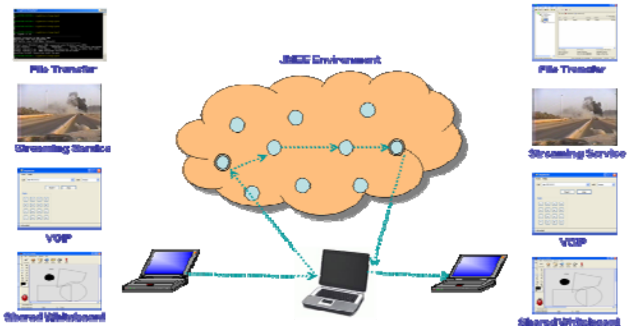

Figure 3: Application-centric analysis setup

For each scenario configured in JMEE, the physical layer data rate for WNW radio interface was set to 2 Mbps and the transmit power was set to 43 dBm. The configured scenarios included terrain data using the DEM format. Application centric analysis was conducted 600S into the simulation, after the networks had converged and results data were gathered. Pairs of nodes were identified in the JMEE scenario as the real-world application source and destination nodes, based on the objectives of the experiment. For example, for conducting the experiments to observe the effects of hop count on applications' performance, the source and destination nodes pairs in JMEE for the real world applications were selected such that their hop distance in the scenario matched the hop count of interest for that experiment .

The following performance metrics were collected for each application:

Iperf: We use the maximum UDP goodput as a performance metric for experiment results. The maximum goodput is the amount of data received in a specific amount of time in the Iperf application between the sender-receiver pair. Unless otherwise specified, the packet size was 1470 bytes and the buffer size was 8 Kbytes for UDP in IPerf.

Ping: The metric that was collected was the round trip time (RTT) between the source and destination nodes.

VLC: The streaming rate of the video at the source was 512 kb/s bitrate for video and 128kb/s bitrate for audio. The video quality was observed at the destination node .

VoIP: The audio quality of the session was observed, and the VoIP packet end-to-end delay and jitter was noted.

FTP : The time taken for a ftp client running on one WNW-like platform in JMEE to download a file of 1.2 MB size from an ftp server running on the another node.

Application-centric Analysis Results

In this section, we present a sample set of analysis results. The results are NOT representative of actual performance for any legacy or current waveform as the protocol parameters are not necessarily set to appropriate values for given scenarios and network deployments. Rather, the intent is to show the ability of the scalable JMEE capability to observe performance of real applications as a function of any network protocol parameter (e.g., slot size, number of slots, frame size, number of hops, etc.) or scenario parameter (e.g., mobility, terrain, number of radios, density of radios, etc.) and demonstrate the interaction among protocols at multiple layers of the stack.

1. Increasing no of radios (or nodes) on applications between 1 hop neighbors

The first simple experiment restricts all application traffic to flow between 2 radios that are 1 hop away from each other while increasing the total number of radios in the scenario. It is expected that as the number of nodes increase in a single subnet, the overhead of control messages such as routing protocol packets is bound to rise. Also, more nodes in the network will lead to higher delay in transmission opportunities for each node. This experiment aims to determine the impact of this (indirect) control overhead on the application; thus demonstrating the significant interactions between different layers of the stack and the need to represent all layers for accurate understanding of end-end application performance.

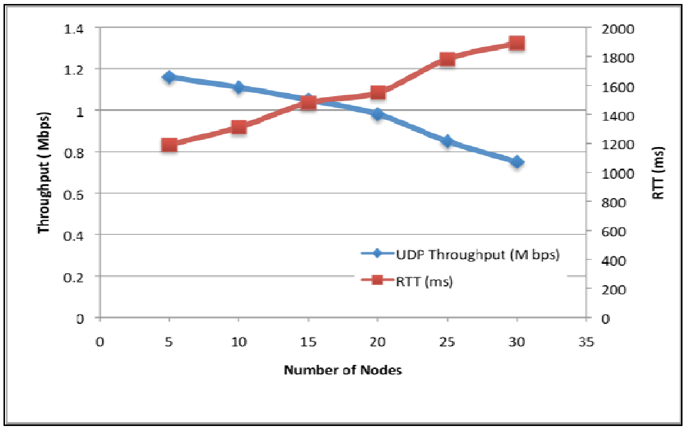

Figure 4: Impact of number of nodes in a network on 1 hop Round Trip Time and Iperf UDP Throughput

Figure 4 shows the UDP goodput reported by Iperf over 1 hop and the ping RTT results as the number of nodes in the network increase. The ping RTT results for this set of experiments indicate that the 1 hop round trip time increases by 60% as the number of nodes increase from 5 to 30, while all protocol parameters are held constant. This is an important observation for delay sensitive applications that are envisioned to run on the WNW network. UDP goodput decreases by 40% as the number of nodes in the subnet increase. Figure 5 shows the impact on the quality of video streaming as the number of nodes in the network increase. We present selected results for the 5, 15 and 30 nodes cases. From our observations, the video quality for a 512 Kbps video stream for 1hop transmissions does not degrade significantly up to 20 nodes in the network. Beyond 25 nodes, as seen in 30 nodes case, we observe a fair amount of degradation. Figure 6 shows the VOIP metric of latency and mean jitter as a function of number of nodes in the network. As the number of nodes increase from 5 to 30, the 1 hop latency rises by more than 100% and mean jitter increases, though by an insignificant amount. This corroborates our RTT observation that delay sensitive applications like VoIP will be affected significantly by the increase in number of nodes in the network. Figure 7 shows the download time of a 1.2 MB file using the FTP application as a function of number of nodes in the network. As can be seen, the file transfer time increases as the number of nodes in the network. The download time increases sharply beyond 20 nodes.

Figure 5: (Top Left) Source Video Stream, (Top Right) Destination quality for 1 hop transmission with 5 nodes in the network, (Bottom Left) Destination quality with 15 nodes in the network, (Bottom Right) Destination quality with 30 nodes in the network

2. Impact of the number of hops on application performance

In this set of experiments, we show the impact of hop count on the performance of the real world applications. A scenario of 15 nodes divided among three subnets was configured in JMEE, with certain nodes configured as gateways between subnets. The source and destination nodes pairs for the real world applications were selected based on their hop distance in the scenario.

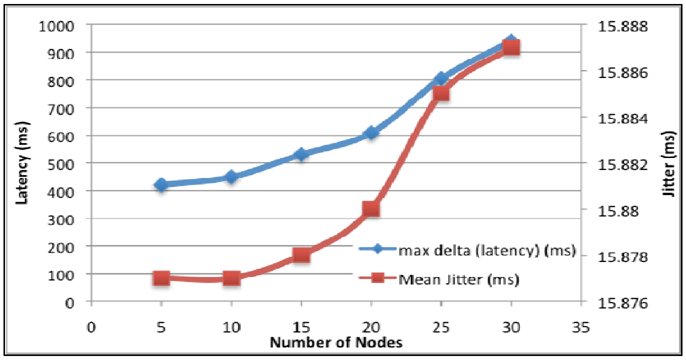

Figure 6: Impact of Number of Nodes in the Network on End-to-end Latency and Jitter of 1 hop VoIP session

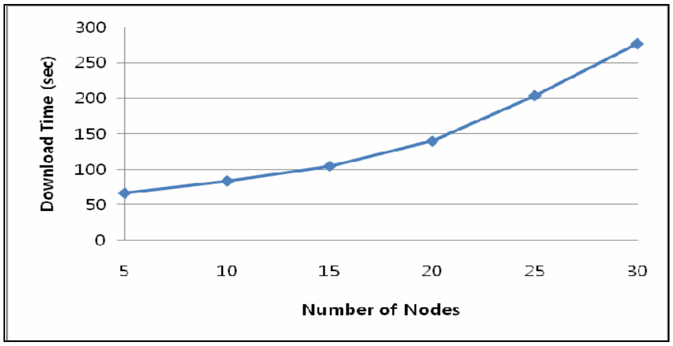

Figure 7: Impact of number of nodes in the network on 1 hop ftp download

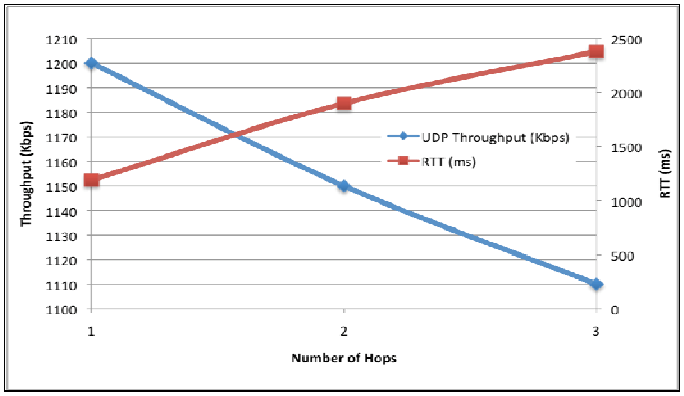

Figure 8: Impact of number of hops on Iperf UDP Throughput and ping RTT

The ping RTT results from Figure 8 show that each additional hop adds 600 ms of RTT. The Iperf results indicate that the increase in hop count does not impact UDP severely. Figure 9 shows the effect of hop count on the video quality at the destination. The increased number of hops gradually degrades the quality of streaming video at the destination. Figure 10 shows the impact of number of hops on the VoIP application performance. We observe that the latency of the session rises in a linear manner as the number of hops. Each additional hop adds around 200ms of latency. VoIP session jitter also exhibits a linear increase as number of hops increase. Figure 11 shows the completion time for ftp download as a function of number of hops. As the number of hops increases, the completion time linearly increases.

Figure 9: (Top Left) Source Video Stream, (Top Right) Quality at destination 1 hop away, (Bottom Left) Quality at destination 2 hops away, (Bottom Right) Quality at destination 3 hops away

3. Impact of mobility on application performance

In this experiment, we show the impact of mobility of nodes on the performance of the real world applications. We use the random way point model for the mobility of nodes in JMEE. We conducted experiments for speeds upto 20 m/s in steps of 5 m/s. We observed the performance of the real world applications over 1 hop transmissions.

In this experiment, RTT is measured at the time that network converges. As a result, RTT remains stable as the mobility (Figure 12). However, in case of goodput measurements using Iperf, node mobility of 10 m/s and beyond significantly affects UDP goodput. The topology changes due to mobility increases the frequency of the route breakages. As a result, UDP throughput drops as mobility increases (Figure 12). Figure 13 shows the impact of increasing mobility on the video quality at a destination 1 hop away from the source node for mobility speeds of 5m/s, 10m/s and 20m/s. We observe that the quality of the video starts deteriorating for mobility 10m/s and beyond. VoIP metrics of latency and mean jitter as the speed of node increases in the mobile network are shown in Figure 14. As the mobility increases, the latency and jitter increase proportionally. Especially, jitter increases sharply as the mobility increases beyond 10m/s. Node mobility of 10m/s and beyond also significantly increases the FTP download time as observed in Figure 15.

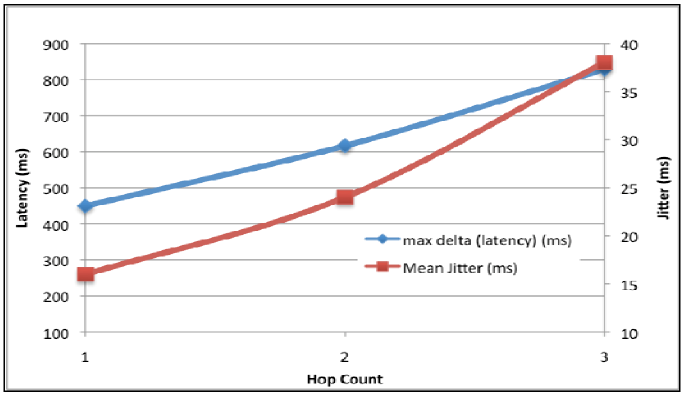

Figure 10: Impact of hop count on VoIP session latency and jitter

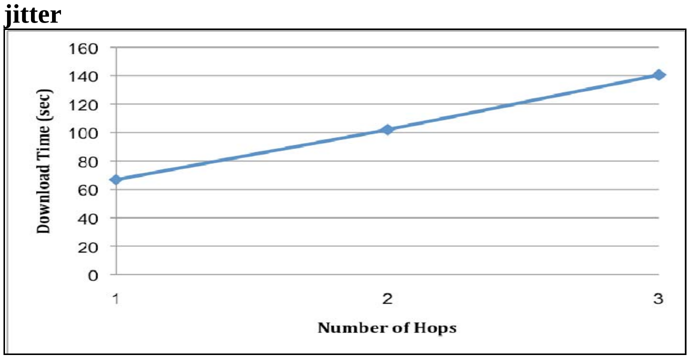

Figure 11: Impact of number of hops on completion time of ftp download

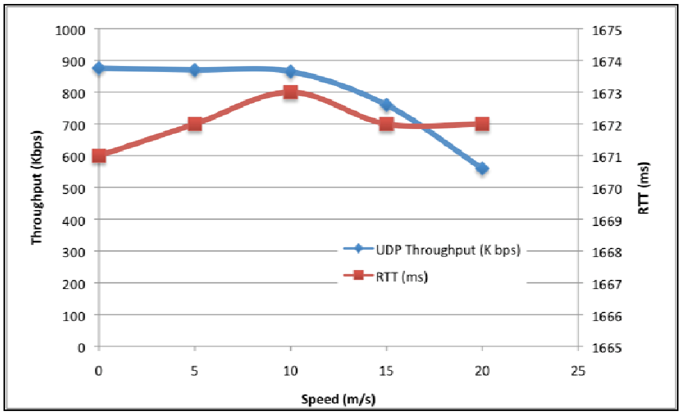

Figure 12: Impact of Mobility on 1 hop Iperf UDP Throughput and ping RTT

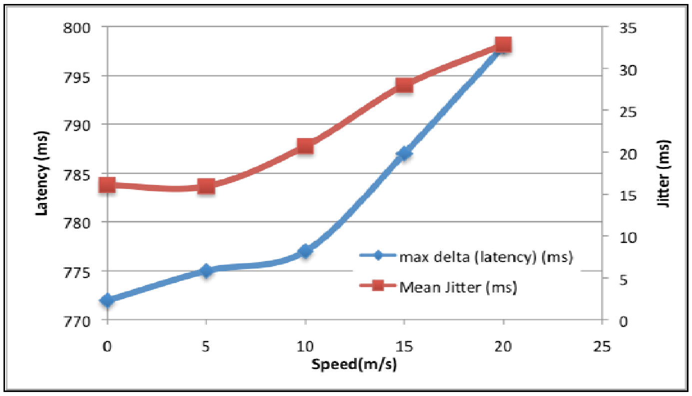

Figure 13: Impact of mobility in the Network on End-to-end Latency and Jitter of VoIP session

Figure 14: (Top Left) Video streaming source, (Top Right) Quality at destination with no mobility, (Bottom Left) Quality at destination with 10m/s mobility, (Bottom Right) Quality at destination with 20m/s mobility

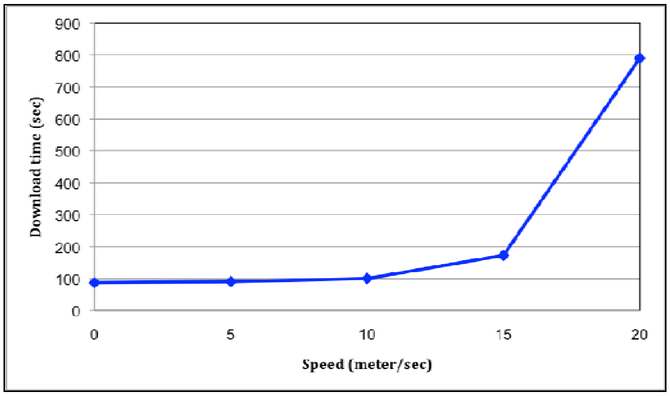

Figure 15: Impact of mobility on FTP download time

V. CONCLUSION AND FUTURE WORK

In this paper, we presented JMEE, a scalable tool for JTRS waveform modeling and evaluation that uses hardware-in-the-loop for interfacing physical networks and live applications to a simulated scenario in JMEE. JMEE provides an efficient, controllable, easily configurable environment to conduct analyses like application-centric analysis, interoperability and scalability testing, typically done on physical testbeds at the design stage of the waveform. The ability of JMEE to support parallel execution is the key enabler for scaling with high fidelity models of waveforms and wireless communication effects while maintaining the real-time execution required for hardware-in-the-loop operation with the physical networks and applications. Through our experiments for application-centric analysis of networks configured in JMEE running a WNW-like representative waveform, we demonstrate how JMEE's capability can be leveraged for analysis of different JTRS waveforms and can be used to draw conclusions about network applications that could be reliably supported on the JTRS networks in its current design state.

Currently work is on-going to augment JMEE with advanced analysis capability, which include features like visual network troubleshooting, support for COTS protocol analyzers, and network events, statistics logging in a database format.

REFERENCES

[1] R. Bagrodia, K. Tang, S. Goldman and D. Kumar, "An Accurate Scalable Communication Effects Server for the FCS System of Systems Simulation Environment," Proceedings of the 2006 Winter Simulation Conference, 2006.

[2] J. Hsu, S. Bhatia, M. Takai, R. Bagrodia, "Performance of Mobile Ad Hoc Networking Routing Protocols in Realistic Scenarios," Proceedings of IEEE MILCOM 2003, 2003.

[3] Scalable Network Technologies Culver City CA, "QualNet User Manual v 4.5." http://www.qualnet.com, 2008

[4] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, K. Gibbs, Iperf Version 2.02, available at http://dast.nlanr.net/Projects/Iperf/, May 2005

[5] VideoLAN Project, VLC version 0.8.5. Available at http://www.videolan.org/vlc , 2006.

[6] S. Doshi, U. Lee and R. Bagrodia, "Wireless Network Testing and Evaluation using Real-time Emulation", ITEA Journal of Test and Evaluation June 2007.

[7] The Network Simulator ns2, http://www.isi.edu/nsnam/ns/, 2008

[8] OPNET Technologies. http://www.opnet.com, 2008

[9] Libpcap, available from http://www.tcpdump.org.

[10] Libnet, available from http://www.packetfactory.net/Projects/Libnet/

[11] S. Morlat. Linphone, an open-source SIP video phone for Linux and Windows. http://www.linphone.org/.