NETWORK DESIGN AND IMPLEMENTATION USING EMULATION-BASED ANALYSIS

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

Douglas McKeon, CERDEC, Ft. Monmouth NJ

ABSTRACT

Current network design and implementation processes employ simulation for analysis at the network design phase and directly step into prototyping based on this analysis followed by physical testbed analysis and full scale deployment. Due to abstractions in the simulated models of the network protocols, it is difficult to maintain accurate predictability of the network performance as the process moves into the detailed design and the subsequent prototyping phase. This paper presents an emulation-based approach for network design and implementation and aims to bridge the gap between simulation and the prototyping phase. Emulation-based analysis uses emulations of protocols and applications in real-time, together with high fidelity simulation models of physical hardware, environment effects and mobility.

This paper provides an overview of the approach and presents a case study to highlight its ability to allow for accurate predictability of the network performance as the network matures from design to deployment.

INTRODUCTION

The design and implementation life cycle for a network system is usually made up of two phases. Protocol design, analytical methods of evaluation (algorithm order statistics) and simulation-based analysis make up the first phase of design & analysis. The next phase is the implementation & testing phase which involves prototyping, physical testbed analysis and full scale deployment. In simulation-based analysis, abstracted protocol models of the network components based on the initial design are created in a network simulator and stochastic traffic generation models are used to analyse the performance and behaviour of the network to obtain initial performance measures. Once the designer is satisfied with the measures, the process moves into the implementation & testing phase where the protocols are implemented on hardware to obtain prototypes and testing is done with real-world applications and network performance analyzers on a full scale physical testbed to ensure that the network performance objectives are met. Often, there exists a disconnect between the performance measures obtained during simulation-based analysis and those obtained in the implementation and testing phase; performance obtained during full scale testing on the physical testbed tends to be much worse than anticipated, when compared with the performance that was observed during simulation-based analysis. The reasons why this occurs are as follows: Simulation-based analysis abstracts out important protocol and device details in order to achieve scalability or reduced run times for quick initial results. Protocol and device quirks may be hidden due to these abstractions, only to resurface again in the implementation and testing phase when the network protocol models are actually implemented on prototype hardware and deployed. Also, analytical and simulation-based analyses only evaluate protocols using stochastic traffic or traffic traces but not with the real world applications which are actually going to run on the deployed network. For example, streaming video application performance may be adversely affected at intermittent intervals by a routing protocol that generates heavy control traffic during these intervals. It is desirable to discover possible flaws with the protocols in the network in the design & analysis phase rather than in the implementation & testing phase, as flaws discovered during implementation and testing will lead to a high cost of redesign and reconstruction of the prototypes which are based on a flawed design.

To address these issues, we propose the introduction of a bridging phase between simulation-based analysis in the early stage of network design and physical testbed

1-4244-1513-06/07/$25.00 ©2007 IEEE

analysis just prior to system deployment, so that accurate predictability of the performance of the target network can be maintained across the development lifecycle of network protocols. Emulation-based analysis can serve as this bridging phase, where, as the design matures, the simulated abstract representation of the target network in the simulator is progressively upgraded to a high fidelity emulation of the network where eventually only the physical environment is represented as a simulation module. Emulation-based analysis enables conducting performance analysis of the real world applications on the target network in the design and analysis phase itself. Interoperability testing of the target network with existing real world systems can also be done in the design phase using emulation-based analysis. Few prototype units running the target network protocols can be created in the design phase and interfaced with the several emulated units to conduct scalability tests and discover protocol and device quirks, which can be fixed before the implementation & testing phase.

In the following sections of the paper, we clearly define the process for conducting emulation-based analysis and indicate how it improves the network design and implementation process. We discuss requirements of network simulators to be used for emulation-based analysis and finally, we provide a case study to illustrate the design and implementation of a Mobile Ad Hoc Network (MANET) using emulation-based analysis.

RELATED WORK

Network designers use simulation-based analysis to model and analyze very complex systems with varying degrees of abstraction (or simplification vs. the real system). In simulation, the model is often simplified or abstracted, which can mask the very phenomena (e.g., adaptive behavior) that are captured during testing on physical testbeds. Emulation, on the other hand, is achieved when models in network simulator are detailed enough such that the modeled network performs exactly the same as the physical system. Emulation can thus serve as an acceptable replacement for the system and is a powerful tool that can be employed during various stages of network design and deployment. Previous efforts focusing on emulation [4] [5][10][11] leverage it as an alternative to full scale physical testbed for network testing and evaluation. Most notable is TWINE [10] where the network stack interfaces with simulated physical channel models, propagation models and mobility models, allowing full control over the test scenarios. Our approach extends the use of emulation into the network design stage by formally proposing an emulation-based analysis phase for network design and provides traceability and network performance predictability during the evolution of the network from design to deployment. The stage-by-stage approach for emulation-based analysis which is described in the following section provides this traceability and predictability.

EMULATION-BASED ANALYSIS

Emulation-based analysis involves the use of the network simulator to run high fidelity emulations of protocols and applications in real-time. All aspects of the network are emulated with the highest fidelity except the physical hardware, environment effects and mobility, which are simulated. Analysis is carried out using emulations of the protocols that are to be used in the network instead of the abstract simulation models used in the simulation-based analysis. Figure 1 illustrates how the emulation-based analysis phase fits within the existing approach.

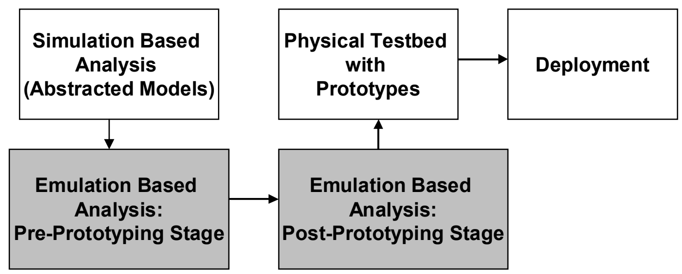

Figure 1: Approach for network design and implementation using emulation-based analysis

To formalize the approach of emulation-based analysis, we propose that this analysis be conducted in two stages:

Pre-Prototyping Stage: This stage follows the simulation based analysis phase and is the first stage of emulation-based analysis. In this stage, real world applications (which are to be run on the target network) are interfaced via hardware-in-the-loop with scenarios in the network simulator running the emulated model of the protocols on nodes. This allows the designer to run initial real world application performance tests with scalability over various scenarios and discover design flaws if any without having to invest in hardware prototypes. Interoperability testing and analysis of the designed network with existing network systems can also be done at this stage by interfacing the existing network system with the emulated network via hardware-in-the-loop.

Post-Prototyping Stage: Once the network designer is satisfied with the real application performance in the pre-prototyping stage, the designer can create few prototypes running the protocol on real hardware. In the post-prototyping stage, these prototypes are interfaced with the emulated models running in the network simulator. Network designers can create large scale networks with few prototype nodes and numerous emulation nodes and analyze protocol behaviour and application performance over various scenarios, with one part of the scenario realized with the prototype nodes and the other part in the simulator.

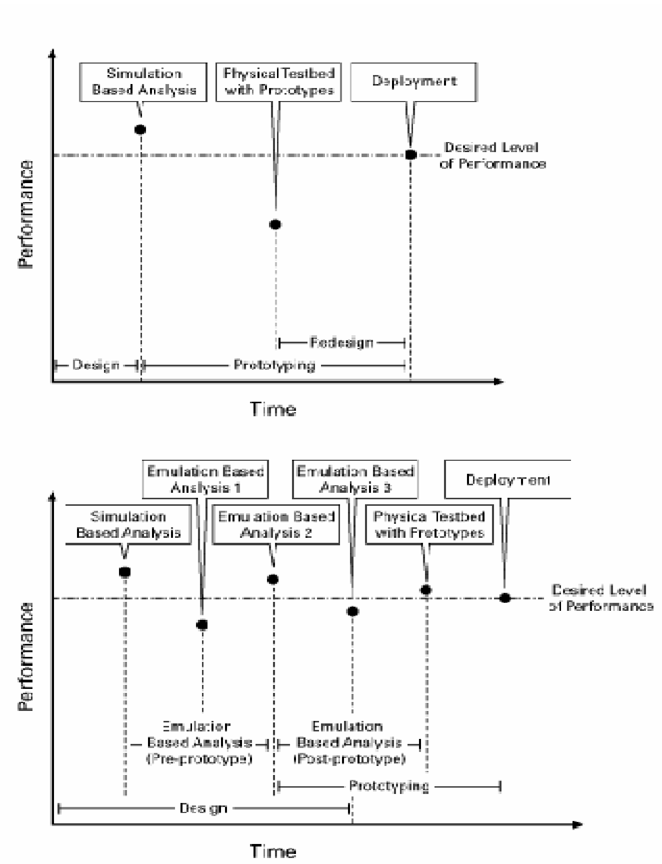

Figure 2: Converging to desired performance level using conventional approach (above) and using emulation-based analysis (below)

The network designer can now gradually introduce more prototype nodes into the network replacing the emulated nodes and thus transition smoothly from part prototype part emulated nodes network to a full prototype network.

The following points highlight the benefits of employing emulation-based analysis:

- In emulation-based analysis, the network designer can interface real world applications with the emulated network via hardware-in-the-loop technologies. The network environment can be varied to set up various scenarios for detailed analysis. This allows for extensive performance evaluation and analysis of the target network in the design phase with the applications that are to be run on it, ensuring better predictability of the applications' performance on the implemented network.
- Existing networks and systems can be interfaced with emulations of next generation networks via hardware-in-the-loop to conduct interoperability testing and analysis in the design phase itself, rather than in the implementation and testing phase.
- Without emulation-based analysis, the implementation and testing phase requires the network designer to produce a large number of prototype nodes to carry out detailed testing of the network on the physical testbed. Tests of scalability might require hundreds of prototype nodes and carrying out controlled, repeatable tests is extremely expensive and resource intensive. Using the emulation based analysis approach, network designers need to only build a few prototypes with the protocols running on actual hardware and interface these with un-abstracted high fidelity protocol models in the network simulator via hardware-in-the-loop. The simulator can then allow the designer to scale with the emulated nodes and have the prototype hardware nodes interact with numerous emulated nodes in various scenarios in the simulator, thus validating protocol behaviour with real hardware during the design phase itself. .

NETWORK SIMULATOR REQUIREMENTS FOR EMULATION-BASED ANALYSIS

A network simulator must satisfy a number of stringent requirements to be used for emulation-based analysis. These requirements are non-trivial, since they heavily depend on the architecture of the simulator and the fidelity of the protocol models in the simulator. The requirements can be summarized as follows:

Protocol Stack Emulation

To support emulation-based analysis and interaction with physical devices, it is essential that the simulator include protocol emulations. In other words, the fidelity of the protocol representation in the simulator must be identical to that of the protocol implementation on a physical device. For protocols based on standards (e.g.Request for Comments (RFCs) or Internet drafts if the standard is not yet formed) the high-fidelity models must be fully compliant with the standard with respect to the packet format as well as protocol logic. This is imperative, since any abstractions in the packet format or the protocol logic in the simulator model will lead to errors and inconsistencies when the emulated network interacts with the physical prototype network, eventually affecting the analysis results.

Simulator Execution Time Synchronization with Real-time

The network simulator 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 emulated network nodes and the physical prototype nodes. 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. An ideal simulator for emulation-based analysis should have the simulation time vs. real-time graph as shown in Figure 3, where the simulation time can continuously keep up with real (or physical) time.

In case of scenarios that involve a large number of emulated radios, 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 and as illustrated in Figure 4, 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. With such support available, the simulator can then easily support scalability in emulated nodes with real-time operation as indicated in Figure 4.

Realistic Physical Environmental Effects

The simulated physical environmental effects have to be modeled at high fidelity in order to appropriately include effects due to fading, terrain, and weather among other environmental effects. This is very important for a simulator since significant abstractions of these effects will lead to inaccurate analysis.

Hardware-in-the-loop Interface

The simulator should have a hardware-in-the-loop capability to enable interfacing the physical network nodes with the emulated network running in the simulator. This hardware in the loop interface must have the capability to inject packets from the physical network nodes into the simulation and vice versa preserving the packet format and contents of the packets as the packets move from one domain to the other.

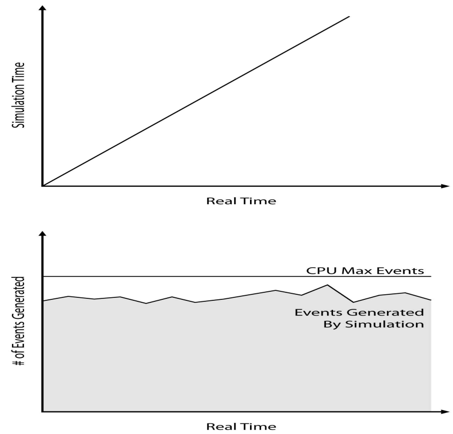

Figure 3: Ideal Simulation Time vs Real Time Graph of Network Simulator for emulation-based analysis

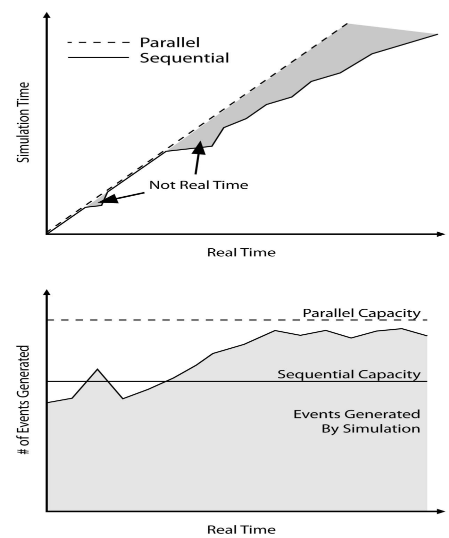

Figure 4: Parallel Execution Capability Mitigates Scalability Issues

Summarizing, a network simulator to be used for emulation-based analysis should have emulation-level models for the network protocol stack, realistic physical and environmental effect representation, an efficient hardware-in-the-loop interface, and finally the ability to scale to thousands of nodes running these high fidelity models with real-time execution. One network simulator that does satisfy these stringent requirements is the QualNet network simulator [7]. QualNet has an extensive set of high fidelity models ranging from the physical layer all the way to the application layer to enable modeling a variety of emulated networks. QualNet accurately models a wide range of physical environmental effects and has the capability to include detailed terrain models in the simulation. QualNet's parallel execution capability ensures that in scenarios containing hundreds of emulated nodes running high fidelity protocol models, the large numbers of events that are generated are processed in an efficient manner by running QualNet on dual core, quad-core processor machines or on parallel computing cluster. Hence simulation time will keep up with clock time all through the simulation for such scenarios. QualNet also has an efficient hardware in the loop interface to couple the physical network and the emulated network in the simulator. QualNet's scenario generator tools allow easy topology configuration and its analysis tools serve well for performance analysis of the emulated network.

CASE STUDY: DESIGN AND DEPLOYMENT OF A MANET USING EMULATION-BASED ANALYSIS

This section presents a case study for design and deployment of a MANET using emulation-based analysis. Consider a situation where a MANET is to be designed which is to be formed between groups of next generation radios and is to be deployed in an urban scenario consisting of dismounted soldier divisions, tank divisions and a humvee patrol interconnected through a low flying UAV as indicated in Figure 5. Applications like streaming video and voice over IP (VoIP) are to be run on the network. Appropriate choice of protocols, protocol parameter settings would be the end result of the design & analysis phase and a successful deployment would be the end result of the implementation & testing phase.

The first step is to use simulation based analysis to identify appropriate candidate protocols for the radios or evaluate simulation models of designed protocols and get the initial performance metrics for the network. Once the initial measures of performance with the desired protocols are obtained, the emulation based analysis phase begins. In the pre-prototyping stage, various scenarios are set up with nodes running emulations of the protocols along with high fidelity models for the urban environment effects.

As shown in Figure 6, using QualNet's hardware-in-the-loop interface, the application sources of streaming video and VoIP are interfaced with the simulator, so that the traffic from these real world application sources can be injected into the nodes of interest in the scenarios and are routed towards identified client nodes. The output from these client nodes is observed on real world clients to check if the application performance meets the desired standards. Protocol parameters can be fine tuned, or protocol emulations can be modified if needed to ensure that the application performance standards are met.

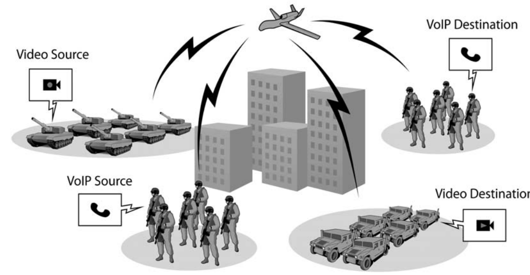

Figure 5: Deployment Scenario for MANET

The network designer now proceeds to construct few prototypes based on the findings of the pre-prototyping stage. The post-prototyping stage involves the use of QualNet to emulate one part of the target network as indicated in Figure 7 with some part of the target network actually realized by the constructed prototype radios.

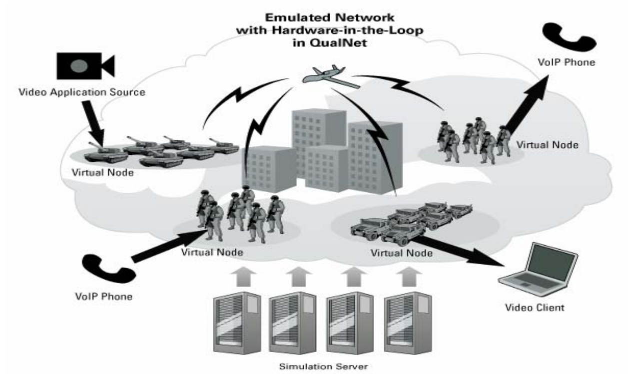

Figure 6 : Emulation-based Analysis in the Pre-prototyping stage

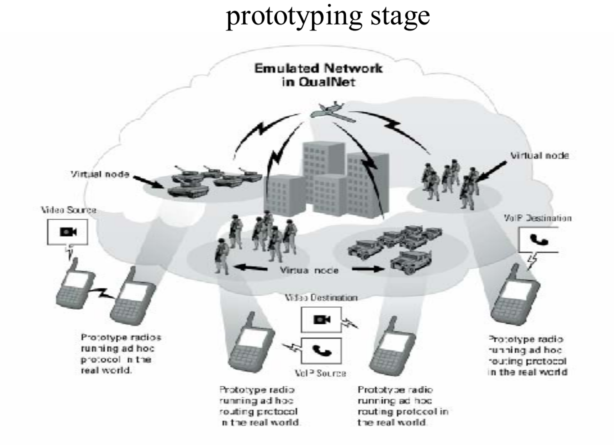

Figure 7: Emulation-based Analysis in the Post-prototyping stage

Figure 7 indicates an analysis in the preliminary stage, where only the video source radio, a router radio in the tank platoon, video destination radio, VoIP source radio and VoIP destination radio are set up as actual prototype radios, while the rest of the network nodes exist as emulations in the QualNet simulator. The prototype node that directly interacts with nodes in the emulated network has representation in the emulated network as a virtual node. These virtual nodes act as gateways between the real and emulated worlds.

The real world prototype network and the QualNet emulated network seamlessly communicate using the real-time emulation capability of QualNet and its hardware-in-the-loop interface; the prototype node does not realize that it is communicating with an emulated node and vice-versa. The routing protocol daemons running on the prototype radios exchange routing protocol information packets with the emulated network. Routing updates due to topology changes caused due to mobility and physical environment effects in the simulation are propagated to the prototype network and are reflected in the prototype routers' routing tables. The prototype video source node runs the streaming video application and the effects of the target network can be observed on the video quality at the destination prototype node. Similarly, the network designer can make a phone call through the prototype node acting as the VoIP source and the voice quality of the call at the VoIP destination node can be assessed as desired changes occur in the target network in terms of mobility based on QualNet mobility scripts. The designer can very easily modify mobility patterns and even change the emulated network parameter settings to assess the effect on the applications' performance.

As analysis in the post prototyping stage continues, any design defects or parameter settings discovered are fixed. Gradually, emulated nodes can be replaced by actual prototype nodes. Exercises can be conducted using live and virtual units; the virtual units being the emulated nodes in scenarios which are not easily replicable for the exercises, e.g. UAVs in urban environments. Once the desired performance level and performance predictability is achieved, the emulation-based analysis phase is deemed completed, resulting in a prototype network that can be further tested on a full scale testbed or actual deployment.

We employed emulation-based analysis to design a network of 100 MANET nodes with 802.11b radios running a link state based routing protocol which is eventually to be deployed in an urban scenario and needs to support a streaming video application. We used simulation based analysis using a simulated constant bit rate application to first determine the end to end metrics that could be achieved with a generic link state routing protocol model available in QualNet

100 nodes were uniformly distributed in 6000 x 6000 m area, and the random waypoint model was used to generate mobility with a maximum node speed of 5 m/s. The average hop count between a source and destination pair was 3. The experiments were repeated 30 times with different random seeds. Each simulated packet size was 1470 bytes and the bit rate for the 802.11b radios was set to 11 Mbps. Table 1 outlines the results obtained.

Table 1: Simulation Based Analysis Results

Protocol	Goodput (Mbps)	Average End-to-End Delay (s)
Generic Link State Routing Protocol	0.94	0.477

The second step was to conduct emulation-based analysis in the pre-prototyping stage and ensure that the emulated network can achieve the desired performance levels of goodput and delay along with acceptable user experience for the streaming video application.

For the performance analysis in the entire emulation-based analysis phase, real world applications and network performance measurement utilities were used. Iperf [8] was used for goodput and end-to-end delay measurement and VideoLAN Media Player (VLC) [9] was used as the streaming video application source. A snapshot of the video being streamed at the source is shown in Figure 8. The goodput and delay measurement results are indicated in Table 2 and the snapshot of the video received at the destination node is shown in Figure 9. As can be seen in the figure, it was observed that the emulated network running the generic link state routing protocol did not provide an acceptable user experience for the streaming video application and an enhanced version of link state routing protocol was needed.

Figure 8: Snapshot of Video being streamed

Figure 9: Snapshot of Video at destination with Generic Link State Routing Protocol during Pre-Prototype Emulation-based Analysis

Table 2: Pre-prototype Emulation-based Analysis Results with generic link state routing protocol

Protocol	Goodput (Mbps)	Average End-to-End Delay (s)
Generic Link State Routing Protocol	0.92	0.8

We repeated the pre-prototype emulation-based analysis using an emulation model of Optimized Link State Routing Protocol (OLSR) [2], and our results (Figure 10 and Table 3) indicated that the emulated network running OLSR with the tuned parameters values of OLSR in the emulated model (stated in Table 4) did provide an acceptable video quality at the destination node. Hence the pre-prototyping stage resulted in OLSR as the protocol of choice with the tuned parameters as the settings for the protocol.

Table 3: Pre-prototype Emulation-based Analysis Results with OLSR

Protocol	Goodput (Mbps)	Average End-to-End Delay (s)
OLSR	1.1	0.4

Figure 10: Snapshot of video at destination with OLSR during Pre-Prototype Emulation-based Analysis

Table 4: Tuned OLSR parameter values

Protocol	Parameters	Default Values (s)	Tuned Values (s)
OLSR	HELLO_INTERVAL	3	2
	REFRESH_INTERVAL	3	2
	TC_INTERVAL	7	5

The next stage in the emulation based analysis was the post-prototyping stage where actual prototype nodes were interfaced with emulated nodes to validate the network behavior with prototype radios. We interfaced 3 prototype nodes with 97 emulated nodes in QualNet. The OLSR routing protocol daemon (olsrd) [6] running on the prototype radios had its OLSR parameter values set to the tuned parameter values obtained in the pre-prototyping stage. The analysis resulted in the following performance metrics in Table 5 and the resulting video quality at the destination in Figure 11, which are consistent with the pre-prototype emulation-based analysis results.

Table 5: Post-Prototype Emulation Based Analysis Results with OLSR

Protocol	Goodput (Mbps)	Average End-to-End Delay (s)
OLSR	1.1	0.4

Figure 11: Snapshot of video at destination with OLSR during Post-Prototype Emulation Based Analysis

Thus, the results from emulation-based analysis indicated that employing OLSR as the routing protocol with tuned parameter values enabled the target network to achieve the desired performance level.. More emulated nodes could now be replaced by prototype nodes to slowly evolve the network into a full prototype network which then could be tested on a physical testbed or deployed, with the performance of the network being accurately predicted using the emulation-based analysis results. Additional analysis with network loading and scalability with thousands of nodes can be performed in a similar manner in the emulation-based analysis phase, saving on resources to do these tests with prototypes on a full scale testbed.

CONCLUSION AND FUTURE WORK

It can be concluded that including an emulation-based analysis phase in the network design and implementation process shall lead to better predictability of network performance during the process. Introducing emulation-based analysis in the design phase following simulation-based analysis allows the network designer to conduct analyses typically done in the implementation and testing phase for evaluating protocol scalability, interoperability of the target network with existing networks and the performance of the real-world applications that are intended to run on the target network. Flaws which are discovered during emulation-based analysis can now be fixed in the design phase itself, avoiding the cost of redesign during the implementation and testing phase.

Work is ongoing to develop more advanced tools in QualNet to enable troubleshooting protocol behaviors during emulation-based analysis and enable for human-in-the-loop interaction with the emulated scenario that allows the network designer to change network parameters on the fly and analyze the resultant network behavior.

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] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR) RFC 3626", available at http://www.ietf.org/rfc/rfc3626.txt, 2003.

[3] 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.

[4] Z. Ji, M. Marina, M. Varshney, Z. Xu, Y. Yang, J. Zhou and R. Bagrodia, "WHYNET: A Hybrid Testbed for Large-Scale, Heterogeneous and Adaptive Wireless Networks, " UCLA Computer Science Department Technical Report CSD-TR060002, January 2006.

[5] G. Judd and P. Steenkiste, "Repeatable and Realistic Wireless Experimentation through Physical Emulation," Proceedings of HotNets 2003, 2003.

[6] OLSR Routing Protocol Implementation version 0.4.9. 2006. Available at http://www.olsr.org

[7] Scalable Network Technologies Culver City CA, "QualNet User Manual v 4.0." http://www.qualnet.com, 2006

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

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

[10] J. Zhou, Z. Ji and R. Bagrodia, "TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications", Proceedings of IEEE INFOCOM 2006 , 2006

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