White Paper

Test & Evaluation of Network-Centric Systems

The Purpose of Emulation

What is Network Emulation?
- Network Emulation vs. Simulation
- Example Emulation Architecture
- Why Not Test Directly on Physical Testbeds?

Emulation with Physical Devices: Best of Both Worlds
- Emulation-based Analysis

Requirements of Emulated Network-centric Systems
- Why Some Simulators Can’t Support Emulation

Case Study 1
- Performance Evaluation of a Wireless Ad Hoc Network

Case Study 2
- Test and Evaluation of a Multi-Role Tactical Common Data Link Network
The Purpose of Emulation

Developers of wireless network-centric communication systems face a huge challenge in developing new technologies that raise the bar in terms of performance while supporting legacy systems. Validating designs from the component level to the system-of-systems level is necessary at numerous stages in the design process. Test and evaluation of these systems mitigates a variety of risks, including:

- program failure,
- technological obsolescence, and
- life and limb of the soldiers communicating through the system.

Comprehensive testing of actual network prototypes is a costly but necessary practice prior to deployment. Not just the hardware, but also the software and systems must be tested, validated and verified. The problem with testing has been that the two methods commonly employed – simulation and physical testbeds – are only of value or viable at the very beginning or very end stages of development. Early stage simulations are not always accurate, whereas late-stage testbeds are very expensive. There is a real need for better, cheaper testing in the middle and later stages of product development.

Real-time network emulation bridges the gap between early stage, highly abstracted simulations and late-stage highly detailed physical tests on prototypes.

In this paper, we will define emulation, describe how it is achieved, and discuss benefits by citing real-life uses of the technology.
What is Network Emulation?

An emulator duplicates something so that it behaves like the original thing. In Computer Science, emulation means the ability to replace “live” physical components of a complex system with counterpart representations that can be deployed rapidly and at a fraction of the cost.

Emulation is achieved when any player in the system— hardware, software or even human — is not able to discern any difference between a real system component and its emulation replacement. Therefore, all emulations are by definition real-time, because they must fill in for the system they represent without other connected systems noticing the difference.

Real-time network emulation uses a network simulator to run high fidelity models of protocols and applications in real-time. All aspects of the network are emulated at the highest fidelity except the physical hardware, environmental effects and mobility, which are simulated.

Network emulation blends the flexibility and low cost of simulation with the highly accurate, highly detailed results of a testbed. It is valuable throughout the life-cycle of the network-centric system and can be employed in increasing level of detail and realism as the project demands.

Network Emulation vs. Simulation

Network simulation models very complex systems. Over the last 20 years, scientists have developed models of physical networks or networked systems, resulting in many solutions, all or most of which abstract, or simplify, the system. Abstraction is practiced in network simulations because complex networks can take a prohibitively long time to generate simulation results, even on fast computing platforms.

Due to the predominance of abstraction in network simulation, it has been viewed historically as an impractical method of testing extremely complex systems. When the model is abstracted, it can mask the very phenomena that are the enhancements being developed. Much of the challenge in network modeling resides in building high fidelity yet efficient radio, channel and mobility models. Network models that fail to take into consideration details like Radio Frequency propagation, channel interference and precise node location may render experimental results meaningless.

When abstraction isn’t used, though, many simulator are overwhelmed by the computational load required of them. Simulation results may take weeks to be generated for a one-hour experiment. Thus, for testing complex networks, simulation is only valuable if there is no abstraction (or negligible amounts), and the model results are generated quickly.

An emulation, on the other hand, is an exact representation of such a system, only in software form. An emulation must be a ‘highest fidelity’ model of the physical system. To ascertain whether the emulation is of “highest fidelity”, a destination network cannot discern if the traffic is coming from a real network or an emulated network. Only under these circumstances, can

Figure 1: Schematic of an Emulated Network Linked to Physical Prototypes
an emulation be used as a replacement for a system, whereas a simulation is only used to analyze or make predictions about that system.

**Example Emulation Architecture**

To give an example of how emulation is used, Figure 1 shows a network emulation construct in QualNet. The QualNet emulation environment is shown with a grid underneath. Two laptops, labeled as Video Source A and Video Destination, send and receive video traffic, respectively. That video traffic flows through an emulated network, which adds many hops, mobility details, wireless environmental effects, and all other details of the wireless network conditions to the traffic. On top of that, another video stream from within the emulation flows towards the Video Destination.

The Video Destination displays two video streams: one originating outside the emulated environment and one originating within the emulation. Both video streams are degraded due to network conditions, and the Video Destination cannot distinguish between the traffic from a real network versus traffic from an emulated network.

True emulation allows traffic sources and sinks to exist anywhere within the real/emulated environments. It also allows packets to cross between physical testbed device and the emulated network without any trace of doing so.

**Why Not Test Directly on Physical Testbeds?**

The purpose of prototypes are to:

- verify that a system provides the advertised capabilities,
- ensure that these capabilities are accessible by users, and
- allow random, possibly unexpected events to reveal deficiencies of the technology.

When building a network system, the most accurate way to test the design is through testing of the actual device. However, there are many limitations with physical testbeds, most notably the cost. An early stage military radio prototype can cost upwards of $100K. Plus, if the design is allowed to reach the prototyping stage without proper testing in the design phase, the cost to go back to the proverbial drawing board is even greater.

Other problems associated with physical prototyping include hardware management. Monitoring nodes and maintaining them, as well as managing experimental controls is difficult and time-consuming. Configuring topology, applications, mobility and experiment details is nontrivial when you are moving tanks, UAVs and soldiers around a full scale field test.

In addition, to produce a scientifically rigorous study, many trials must be performed. Repeatability is an important requirement for testing. Being able start and stop trials, as well as pause and modify parameters on the fly is also necessary to efficiently conduct experiments. All these are difficult to do with physical testbeds.

**Emulation with Physical Devices: Best of Both Worlds**

Best of breed solutions incorporate hybrid simu-
lation capabilities that allow integration of simulated, physical (real) and emulated components into a single evaluation platform for scalable and realistic network experimentation. This can provide the experimenter with the best trade-off in terms of realism, scalability and repeatability depending on the target network scenario.

Figure 2 is a cost analysis for physical testbeds alone vs. emulation with physical testbeds. The cost estimates include hardware and software costs but do not include labor costs for setup of prototypes, which adds considerably to the Physical Device Only setup.

Setting up prototypes is a time-consuming exacting science. Following are some examples of network constructs you could test with emulation that would be cost-prohibitive with physical devices:

- urban vs. rural environments,
- weather effects, such as the influence of snow or rainfall on network performance, and
- compromised or destroyed nodes on a large scale.

**Emulation-based Analysis**

The V-chart in Figure 3 shows the effect of emulation on the development cycle of network-centric systems. Development progresses from left to right. In the System Design stage, or left half of the V, first broad ideas are developed, and then the detailed ideas / component level details are fleshed out. Flaws in the designs, when caught in this phase, are relatively inexpensive to rectify.

Once the System Design phase is over, System Testing takes place, which is depicted as the right half of the V. During System Testing, first the components are evaluated with respect to their original requirements, followed by increasingly larger chunks of the solution, culminating in the test and evaluation of the entire system itself. Flaws caught in the System Testing phase are far more costly to correct than in the System Design phase.

Before emulation, developers of network technologies could use network simulation in the early system-level design phase, and then physical testbeds in the later system testing phase. Emulation allows many more opportunities to test and evaluate designs throughout the maturity-realism continuum. In other words, emulation allows testing that’s better, cheaper and at a variety of granularities. The end result is network-centric systems can be better tested and understood more deeply at much earlier stages in the development cycle.

With emulation, network and component designers need build only a few prototypes for most or all of the testing and evaluation process. Protocols running on actual hardware interface with unabstracted high fidelity protocol models in the simulation. These high fidelity protocol models are emulations of the protocols running on the real hardware. The simulation construct allows prototype hardware nodes to interact with emulated nodes in various scenarios in the
simulation construct, thus validating protocol behavior with real hardware in the design phase.

Design flaws can be corrected at early stages in the design process, saving the network designer considerable time and money versus if these flaws were detected later in the development cycle. We term this emulation-based analysis.

Emulation has huge value outside of the design and development process for testing and evaluation, such as integration and training. Emulations can be used in a system of systems environment by networking existing systems to systems already under development, and new systems to be developed.

Requirements of Emulated Network-centric Systems

Real-time

For a model to perform exactly the same way as the physical system, it must run in real time, all the time. The physical sub-network must not be able to distinguish a physical device from its real-time emulation.

Real Packets

Packet contents inside the emulation must be the same as packet contents outside the emulation. This allows for packet capture anywhere within the network, regardless of whether a node is emulated or real.

Transparency

Since are the packets are real within the emulation, we also need transparency to take advantage of this fact. Transparency allows network designers to interact with the network live. Examples of ways network engineers might interact with the network include:

- passive monitoring, such as collection of state information,
- traffic insertion, such as introduction of traffic to measure achievable bandwidth or delay
- triggers, such as conditional commands that notify a network engineer when a particular node moves into a particular region, or
- execution speed control, such as pausing, slowing down, or even fast forwarding uneventful periods in the scenario.

Real-time emulations using real packets that allow dynamic interaction are extremely powerful

Figure 3: QualNet Emulation Environment has Full Packet Fidelity and Allows Visibility Into and Interactivity with Emulated Network.
tools. They are true replacements for networks and allow a myriad of third party applications, such as Semi Automated Forces (SAF) applications, HP Open View, and Ethereal to interact with the emulation. Examples of ways that third party applications can broaden the capabilities of a QualNet emulation include:

- applications that locate the source of a bottleneck can assist in solving network design problems, and
- applications that suggest alternate network topologies and constructs.

**Why Some Simulators Can’t Support Emulation**

Historically, simulation was considered unsuitable for testing and evaluation of network-centric systems because of speed and fidelity concerns. Even on the fastest computing platforms, some simulators can’t emulate the network because they can’t support real packets inside the emulated environment, nor support full-scale networks in real-time.

One way that slow simulators try to work around the scalability issue is to daisy chain smaller (e.g. 50 node) networks instead of grouping nodes together in a large-scale (e.g. 500-node or more) network. Under the daisy chained scenario, the smaller networks are not time synchronized, reducing the realism of the emulation.

Another way that other simulators cut corners to stay in real-time for emulation is to take the contents of the packet out of the emulated environment. So-called “Process Models” convert real network traffic into events for the simulation engine, effectively stripping the packets of their contents once they enter the emulation. The packets then recover their contents as they leave the emulation. This poses a number of problems, including:

1. **Symmetry Requirement.** If traffic originates outside the emulation, it must terminate outside the emulation; traffic originating inside the emulation must terminate inside the emulation. This is a big limitation, as the primary purpose of the wireless tactical network is to provide a means of communications among local actors to solve local
problems. Any 2,3 or n number of users should be able to form a network, exchange information among themselves and collaborate to solve problems, regardless of whether they exist in the emulated or real domain.

2. Processor Load. It is very processor-intensive to rewrite every packet, stripping the contents out and then replacing them back, but there is no other way for slow simulators to support real-time emulation without abstraction of packet contents.

3. Lack of Transparency. The purpose of network emulation is to replace networks with perfect models that integrate with a larger system seamlessly. The test and evaluation community uses emulation to root out and diagnose problems with proposed network constructs. One such way these problems are discovered and solved is by tapping into any location of the emulated network and drawing out real packets. A network emulation, like the network it models, allows for interactivity at any layer in the stack and any location.

QualNet is a network simulator that supports emulation. QualNet supports the scalability and real-time requirements of network-centric systems through parallel execution. QualNet also supports full packet fidelity, detailed RF propagation, channel interference, and complex terrain. Unlike other simulators, QualNet provides a flexible, interactive complete replacement for a real network. QualNet is a true emulator.

**Case Study 1**

*Performance Evaluation of a Wireless Ad Hoc Network*

In this use case, QualNet was used as an emulation engine for a network of 23 tanks, each containing a radio. One tank is streaming video...
to another, and the traffic is routed through the network of 23 nodes. The performance of the network was emulated under four scenarios:

- static tanks in a rural environment
- mobile tanks in a rural environment
- static tanks in a rural environment with interfering network traffic, and
- static tanks in an urban environment.

The scenario was set up in a laboratory environment where the video source, destination, and one intermediate hop nodes are represented as laptops A, B, and D, respectively. The rest of the network was emulated by QualNet running on Laptop C.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Average Throughput (Mbps)</th>
<th>Average Delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Tanks, Rural</td>
<td>1.84</td>
<td>0.328</td>
</tr>
<tr>
<td>Mobile Tanks, Rural</td>
<td>0.85</td>
<td>0.482</td>
</tr>
<tr>
<td>Static Tanks, Rural with</td>
<td>1.40</td>
<td>0.336</td>
</tr>
<tr>
<td>Interfering Traffic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static Tanks, Urban</td>
<td>1.61</td>
<td>0.325</td>
</tr>
</tbody>
</table>

Figure 7: Quantitative Performance Metrics
Figure 6 shows how the scenario was created using the hardware-in-the-loop capability of QualNet.

The first scenario, Static Tanks in a Rural Environment, was considered the control. Any military radio could be used in this QualNet experiment, such as Link-16, Soldier Radio Waveform (SRW), Enhanced Position Location Reporting System (EPLRS) or Single Channel Ground and Airborne Radio System (SINC-GARS).

When mobility was introduced, the tanks were allowed to move randomly at 5 meters per second. When network interference was introduced, Constant Bit Rate (CBR) traffic was generated by QualNet and flowed within the emulated network at 1.3 Mbps. When the scenario was moved from a rural to an urban setting, CTDB8 terrain format was used with the QualNet Advanced Propagation: Urban model to account for environmental effects.

Performance metrics of the scenarios are included in Figure 7. Snapshots of the streaming video under the four scenarios are shown in Figures 8-11. In each case, the video at the Source Node is shown on the left and the video at the Destination Node is shown on the right.

The QualNet emulation showed that the worst performance was measured in the mobile network in the rural setting, followed closely behind by the static network in a rural setting with interfering traffic. From the perspective of video quality, the end-to-end delay was far more critical to application performance than end-to-end throughput.

**Case Study 2**

*Test and Evaluation of a Multi-Role Tactical Common Data Link Network*

L-3 Communications is using QualNet to develop the MR-TCDL (Multi-Role Tactical Common Data Link) Network. MR-TCDL is a Wireless, IP-based, Mobile Ad-hoc Networking System that provides high bandwidth capacity over very long ranges.

QualNet is used to validate MR-TCDL applications and hardware and refine system designs. Components of the system modeled by QualNet include:

- Inter-Platform Communications Manager (IPCM), a network control services suite. IPCM is embedded on the L-3 WMR router and provides automated, proactive, Ad-hoc wireless network and physical layer management, and
- CDL (Common Data Link), very high bandwidth, long range (L-3 CSW) radio communications system

IPCM acts as a cross-layer network-centric control service. It is designed to optimize the physical and network interconnections based upon many discriminators, such as distance between nodes, QoS, traffic profiles, radio frequencies, power, data rates, and platform/terrain blockages.

IPCM runs in a distributive manner, where nodes join the network and become part of a collective artificial intelligence (AI). An MR-TCDL/IPCM network elects Peer Group Leader (PGL) nodes and functions as an intelligent Ad-hoc cluster of nodes.

IPCM is designed to provide dynamic real-time responses to changes in the RF media and the associated network layers. This in turn optimizes the interconnections and maintains a stable physical layer and network connectivity.

*Pre-Test Flight Emulation*

Prior to test flights featuring real aircraft, L-3 connects real WMR/MRT-CDL prototypes to an emulated fleet of aircraft in QualNet. The pre-test-flight validation process, in system design and development phases, is intended to reduce the risk of system or network failure during system testing and actual flight test acceptance phases.

With real hardware and actual algorithms integrated into the simulation process, protocol service verification takes place throughout various stages of product development, not just at the final step. QualNet-based simulation is critical to understanding and refining the performance of IPCM, network protocols and other network services.

Once the test flights are underway, QualNet is used to provide flight test support with real-time 3D visualization of link status and telemetry data. By providing operational-level validation of networking software and hardware, emulation
helps L-3 build a better solution faster and more inexpensively. Figure 12 is an example of the emulation architecture with flight testing. Real, emulated and simulated nodes are noted where applicable.

Because it is relatively inexpensive and quick to set up, emulation can be used to validate a large variety of design constructs, from protocols to bigger picture flight-test and mission planning requirements. Beyond the initial analysis, critical tuning of the system can be used to evaluate and find optimized configurations for a final solution.

QualNet’s integrated physical layer modeling, combined with its high fidelity radio signal strength degradation effects provide additional capabilities that enrich the base functional testing services of the hardware emulator. The high fidelity PHY models also provide comparative analysis for integration testing. Working together, these different validation methods form a comprehensive (layered and cross-checked) approach to reducing technical failure risks prior to scheduled flight tests.

The following summarizes a typical pre-flight test emulation-based validation process for a next-generation device such as the L-3 WMR router:

- Determine the wireless networking equipment configurations. Identify hardware, software, and platforms.
- Define required performance metrics and acceptance criteria for simulated flight tests. Define the emulation test scenarios that will run concurrent to actual flight tests.
- Perform flight test simulations and collect instrumented simulation run data, i.e., network, IPCM, and physical layer statistics.
- Perform results analysis on simulation run data, and compare against desired acceptance criteria (i.e. apply standard “statistical control process” quality assurance methods).

Figure 12: Sample emulation architecture with flight testing. The scenario includes real, emulated and simulated nodes.

Figure 13: Sample output from the emulation environment using AGI’s Satellite Tool Kit.
Sample Test Flight

Once the emulation-based analysis is performed, flight tests involving deployable hardware and software are planned. Figure 14 shows a typical flight test scenario. Other critical phenomena at play that are represented in the emulation are antenna blockage due to wing shadowing, as well as terrain and weather conditions. Both play a key role in determining communication success.

Beyond the design phase, emulation is expected to play a key role throughout the product development and qualification testing phases, such as:

- Mitigating failure risk by iteratively employing emulation-based validation during all phases of a wireless networking system’s lifecycle.
- Use simulation-based validation processes to establish a realistic networking system “performance envelope” (numerical data based) prior to initial flight demonstrations. A known “performance envelope” reduces failure risk, and can provide a valuable outline of expected flight test results.
- Perform critical closed-loop analysis between simulation predicted trials, and actual flight test performance metrics by comparing actual FlightView results, against targeted flight test (subset) simulation scenarios.
- Use observed data sets to derive actual flight test-based scenarios to build a CONOPS scenario simulation reference library. In other words, earlier emulation efforts are used to spawn a different...
Network-centric emulation through QualNet provides the ability to rapidly test a networking system under a wide variety of operational conditions. It reduces a huge complex problem space into a reasonable problem space via the data reduction and programmable analysis in the post run analysis tool kit. The simulator then provides the architecture and tools to rapidly diagnose, understand, and resolve problems once they are detected.

Emulation provides a cost effective, repeatable, controlled systematic validation process to ensure the performance, reliability, and quality of next-generation network-centric warfare systems.

**Summary**

New techniques employing highest fidelity, real-time network emulations are changing the way that network-centric systems are tested and evaluated. After early stage, abstracted simulations and before late stage full prototype test beds, emulation plays an important role in evaluating network constructs. From the component level to the system-of-systems level, emulation enables network designers to test and evaluate the system, greatly reducing the cost and effort required to thoroughly test and validate throughout the design cycle.

Not all network simulators are capable of true emulation, however. To be a true replacement for a network, an emulation must support real-time speed, full packet fidelity, and provide transparency.

QualNet is a network simulator that supports true network emulation. With parallel execution and highest fidelity models, QualNet has the scalability and interactivity required to test and evaluate the most advanced network-centric systems.
References


About Scalable Network Technologies

Headquartered in Los Angeles, California, Scalable Network Technologies is the leader in parallel processing technology for network performance evaluation. The company develops and supports high-fidelity evaluation software tools used for predicting the performance of computing and communications networks and network devices.

SNT has created a new category of evaluation tools for today’s sophisticated networks that meets the demand for real-time, real-network performance testing. Widely recognized for its flagship product, QualNet, the company’s customers include major aerospace and defense contractors, the US Department of Defense, mobile network operators, as well as research agencies and universities.

For more information or a free 2-week trial of QualNet Developer software, please visit:


6100 Center Drive
Suite 1250
Los Angeles, CA 90045

http://www.scalable-networks.com
http://www.qualnet.com

For additional information, please send email inquires to: sales@scalable-networks.com.