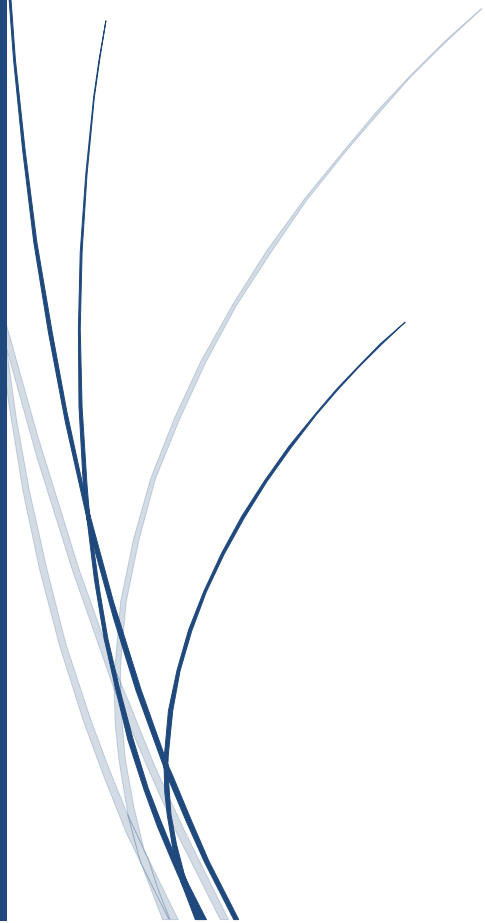
FAE TELECOM
DEFINE · DESIGN · DEPLOY

Whitepaper

Simulation Network Environment for use in SDN/5G migrations supporting Cloud use for support of an all-purpose network transport (APTN) design.

Introducing MILSIMNET: the simulator for Military networks



Contents

NOTE: MilSimNet and GridSimNet are marketing names, for our simulation software, with focus in two market areas. Military networks are MilSimNet, and the features of simulation are the same, but the user focus is different. GridSim, is aimed at the Power Industry and their Challenges in Substation modernization. There are examples of Power Grid applications throughout our white paper. These are to showcase the Power Industry use of the tool. Most Military uses are not publicized.

A Changing Landscape

Simulation, AI, and Testing in the NaaS of the Future

"Virtual networking or Network-as-a-service (NaaS) makes connectivity easier, but it complicates automation of lifecycle processes. The problem is that when "services" are created on top of connectivity rather than through the same devices that provide connectivity, you lose some insight into service behavior. You also lose pretty much all the traditional ways of doing management. There's talk about "monitoring", "analytics", "simulation", and "artificial intelligence" to fill the gap, but little detail about how that might work.

*The basic principle of automated service lifecycle management is that there are a series of generated **events** that represent changes in conditions. These events are then **analyzed** so that a cause and correction can be determined, and then a **remediation process** is launched to restore normal (or at least optimal under current conditions) behavior. This is the process into which we must fit any new tools we hope to adopt.*

*The tools? These also need some definition. **Monitoring** is the real-time examination of network behavior to identify deviations from the desired operating state. Such deviations are then coded as **events**, things that represent alerts that must be handled in some way. **Analytics** is the broader process of historical trend analysis or correlation analysis, aimed at deepening the insight you gain from traditional monitoring, usually to detect things earlier than monitoring would. **Simulation** is the process of modeling the behavior of the network under a specific set of conditions, and it can be used either to help predict how current monitoring/analysis conditions will play out (giving you yet more advanced notice) or predict how effective a given remedial process will be. **Artificial intelligence** is similarly a tool to interpret our "events" to either predict how a problem will unfold, or what reaction to the problem is most likely to be successful. Got it?*

*We can apply our tools to our lifecycle model now, starting with the "event" generation. Let's say that the network has a **goal state**, which is implicit in nearly all operations processes. There's a network plan, perhaps a capacity plan, that is based on assumptions about traffic and connectivity. That plan implicitly defines a network behavior set that's based on the plan, and that's the goal state. If the current network behavior deviates, then it would be expected that the plan would be invalid, and some remediation would be required. Monitoring and analytics are the traditional and leading-edge (respectively) ways of detecting a deviation.*

Simulation could be a powerful tool at this point, because when the network's presumptions on connectivity and traffic are known, those presumptions could be used to set up a simulation of normal network behavior. What simulation brings to the table is the opportunity to model detailed conditions in the network as traffic and connectivity demands change. That provides specific ways of generating early warnings of conditions that are truly significant, versus just things that won't make much of a difference or will be handled through adaptive network behavior.

AI can also be injected at this point. If we have a good simulation of "normal" behavior and we also have a simulation of how the current set of conditions is likely to impact the network, we can then use AI to establish a preemptive (or, of we're late in recognizing things, a reactive) response. That response could then be fed into the simulator (if there's time) to see if it produces a satisfactory outcome. We

could also use AI to define a set of possible responses to events, then simulate to see which works best. That could then be fed back into AI (machine learning) to help rule out actions that shouldn't be considered.

In the response area, it's AI that comes to the fore. Machine learning is often represented as a graph, where we have steps, conditions, and processes linked in a specific way. Think of this as an augmentation of normal state/event behavior. Because all service lifecycle processes should (nay, I would say "must!") be based on state/event handling, augmenting that to provide for more advanced analysis of the state/event relationship would be expected to improve lifecycle automation.

That doesn't mean simulation isn't helpful here, either to supplement AI by providing insight into the way that proposed responses will impact services, or by offering a path to predict network behavior if conditions fall outside what AI has been prepared to handle. Most useful "process automation AI" is likely a form of machine learning, and simulation can help with that by constraining possible solutions and by offering something that's perhaps almost judgment-like to solve problems for which specific rules haven't been defined.

It should be clear here that the combination of AI and simulation serve in lifecycle automation almost as a substitute for human judgment or "intuition". The more complex the network is, the more complex the services are, the more difficult it is to set up specific plans to counter unexpected conditions—there are too many of them and the results are too complex. It should also be clear that neither simulation nor AI are the primary tools of lifecycle automation. We still need good overall service modeling (intent modeling) and we still need state/event or possibly graph-based interpretation of conditions.

All of this, we should note, is likely to be managed best at the facility level rather than at the virtual network level. NaaS is a wonderful thing for operations, but less so as the target of remediation. Real faults can't be corrected in the virtual world; the best you could do is to rebuild the virtual structure to accommodate real conditions. That could create the classic fault avalanche, and it would be easier and less process-intensive in any event to remediate what can really be fixed—there's less of it and the actions are more likely to be decisive.

There's an impact on testing here, too. The industry has been struggling with the question of whether you should do "connection network" testing in even IP networks, where finding a traffic flow means examining routing tables to see where it's supposed to go. In NaaS-modeled networks, we have a clean separation of connectivity and the associated underlayment, but even that underlayment may be based on IP and adaptive behavior.

One possibility for testing is the "deep dive" model; you go to the bottom layer that's essential for everything else and test the facilities there. This is difficult to separate from monitoring, though. Another possibility is to maintain connection-level testing, but that's going to be much more difficult unless you have a standard mechanism for setting up an overlay network. Today in SD-WAN, for example, we have dozens of different overlay encapsulation approaches; can we expect test vendors to adapt to them all?

The best approach is probably to provide what could be called a "testing API", which probably doesn't need to be anything more complicated than a way to establish a specific pathway then examine how traffic flows through it. Since this pathway would look like a standard connection to the NaaS layer, you'd be able to get it in a standardized way even if the underlying encapsulation of the overlay network differed from vendor to vendor.

There's still our reality of fixing virtual problems in the real world to contend with. Do we really want to do "testing" anywhere except at the connection edge of a NaaS or at the physical facilities level? What do we expect to learn there? That's something testing vendors will need to think about, particularly as we expand our goals for automation of service lifecycle management. There's little point to that if we then expect networking personnel to dive into bitstreams.

Again, simulation and AI can help. We need to think more in terms of what abnormal conditions would look like and how they should be remediated, and less in terms of geeks with data line analyzers digging into protocols. The rational goals for network automation, like the implementations of network modeling, should focus on intent-modeled elements that self-remedy within their scope. If we set that goal, then AI and simulation are going to prove to be very powerful tools for us in the coming years."
Tom Nolle – CIMI Corp

Today's networks are changing due to IP transition, along with SDN and 5G planning for deployment are all driving major re-architectures of the networks. Any one of these by itself presents the telecom engineers with major design issues, put all five in play and the need for new tools to help are clear.

Our First release version of MILSIMNET will help and more powerful and useful tools with Modeling and AI and Machine learning will be included in each new version developed and brought to market.
The telecom engineers managing the production network need to be able to initially understand what equipment is installed in the network and then be able to model the re-architecture and test the planned changes before they go live on the production network.

Network Re-architecting

This is a complicated task considering that there are new challenges appearing every day. For most mission critical infrastructure it is not practical to have a test environment that is the same size as the production network, especially with thousands of smart devices being installed, creating networks with an exponential number of endpoints. This makes the task of recreating different design scenarios not only hard but in many situations impossible. How do you assess the impact of 25,000 smart devices? Buy and install them in a warehouse to recreate the network? How do you test the IP transition plans, with new IP based equipment, to really know what impact that equipment will have, or even which design option is the better? Trial and error on a small test network? How does that scale? We are encouraged by interest in the simulation tool in the Power Industry.

To reconstruct the issues of a scalable product network, the only viable solution is to supplement the lab test environment with thousands of simulated devices, using a Network Simulation Environment MILSIMNET) simulator.

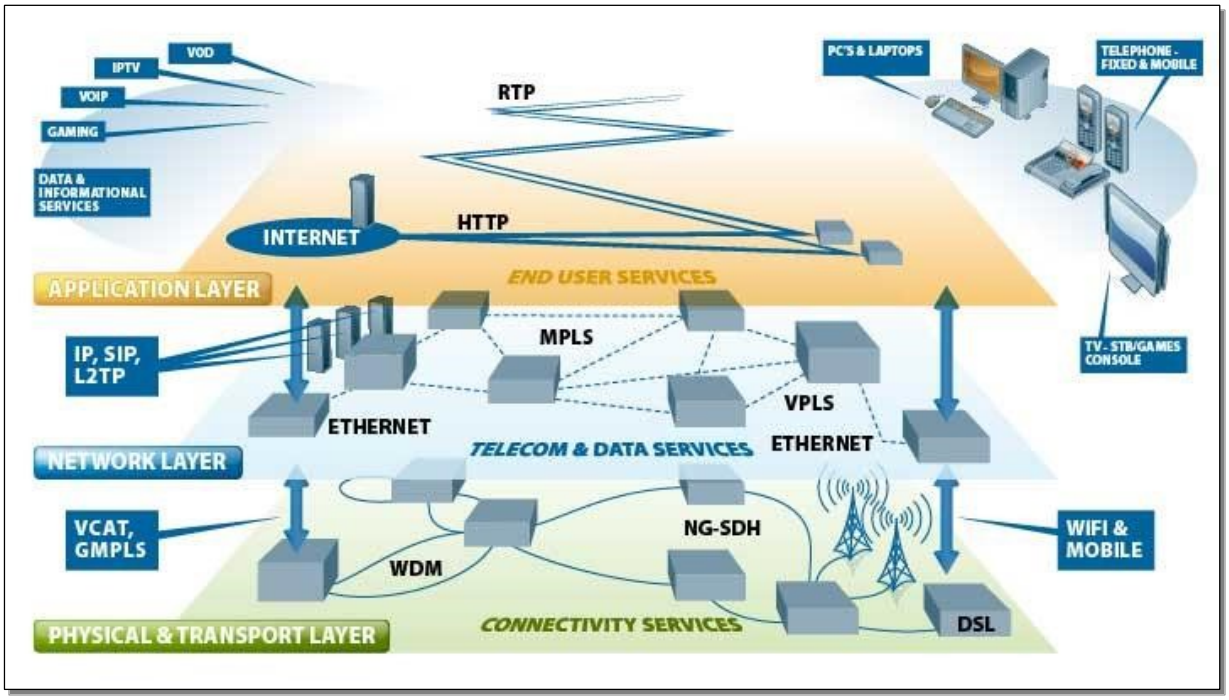
MILSIMNET simulators enable the telecom engineers to setup a scalable simulated network with 100,000s devices. They can then simulate different combinations of routers, switches, firewalls, WLAN controllers from many different manufacturers, without having to purchase one or impact the production network.

Think of network simulation in the same way pilots use a flight simulator—you can crash the simulation as many times as you want without causing any damage or interruption to the production network.

Converged Networks

Converged networks integrate networks that have historically been separated into a single, common infrastructure. The integration of these resources has two main objectives:

- Reduce the overall cost of the network – allowing the sharing of operations, administration, maintenance and provisioning.
- Allow the deployment of new network based critical applications that utilize data, voice and video to exploit the synergies that this creates.



This diagram illustrates the key areas of a Software Defined Networks (SDN) that is designed to support both voice and data applications.

However, one of the fundamental issues with designing and implementing an SDN is that there is no single solution for all regions of the network. This means that every design/implementation of an SDN is different – presenting significant challenges.

This uniqueness is a result of several issues:

- Multiple converging technologies that are not always required in every network.
- Multiple service level requirements, with different ways of ensuring Quality of Service (QoS).
- Multiple suppliers and carriers using different technology to compete.
- Multiple standards and specifications from the "standardization" bodies.

Utilizing FAE Telecom's All-Purpose Transport Network (APTN) design philosophy, the components of a NGN are going to be drawn from many different suppliers and technologies. Ensuring quality and compatibility of all these components before, during and after installation is going to be critical to the successful operation of an SDN.

Providing effective, cost efficient frameworks to support the network deployment life cycle is -- a critical concern for network dependent businesses. Furthermore, as new protocols and services are deployed with complex interactions to existing sets, the need to simulate these networks increases significantly. This white paper examines the use of Simulation Network Environments as the best way to effectively design, deploy and support SDNs.

What is a Simulation

Virtual Network Environment simulations leverage network management and simulation technologies to create a "virtual" computer-based picture of a network. A simulation is the act of exporting MIB object instances and values, just as a real-world manageable device does, but without the actual physical device. The network management application interacts with the simulations within the MILSIMNET just as it would with real-world devices. The most common is a "realistic" simulation, i.e., it attempts to duplicate the behavior of a real- world device.

For a device, any number of simulations can be run, just as in the real world there are several scenarios in which a device can be involved. For example, the user can create a router simulation of a lightly loaded device or an overburdened device, or any range of scenarios in between. Or, the user can simulate an RMON probe on a healthy network segment, or a probe that is monitoring a segment with either a high traffic load or failing devices. From a network management perspective, the difference is seen merely in the instances and values of returned MIB objects.

Simulations with randomness have their limitations in the case of product testing because the values of MIB objects are unpredictable. For this use, a MILSIMNET allows the creation of "constant" simulations. This type of simulation makes counter objects return a value with a constant rate.

MILSIMNET are more extensive than can be created with any physical network test environment. A MILSIMNET is ideal for developing and testing network plans for scalability, robustness, performance and effective policy implementations. It is possible to easily extend a physical environment via the simulation of many thousands of manageable network nodes. MILSIMNETs cost-effectively provide network designers, testers, implementers, and trainers with a private, virtual network. This eliminates the overhead and headache associated with having to purchase, maintain and administer physical equipment – which could be physically and financially prohibitive for large-scale networks. MILSIMNETs significantly increase the efficiency designing, implementing and supporting large NGNs.

MILSIMNETs combine real device hardware with simulated devices to produce a larger, more complex configuration that is more flexible and more cost effective as a tool for network traffic simulation, device test, disaster planning as well as training. Multiple devices can be configured in such a way that when you have an effective simulation of a large-scale network, network management applications cannot tell the difference between real device hardware versus simulated. Also, with the proper software configuration tools, these simulated devices can be configured more easily, and in bulk, so as not to disturb the real network backbone. Simulated devices serve much better as a test and training network in that anything that is modified on a simulated device does not risk a network "crash" or inhibit the production network performance.

MILSIMNETs can serve as robust grounds for testing multiple devices without having that device in its physical form, therefore, saving much of the budget and valuable resources for other team members in other capacities. The key to maintaining and configuring a MILSIMNET is having configuration software that can handle multiple devices easily and efficiently.

Increased ROI with reduced cost of network engineering

MILSIMNETs reduce capital costs of a network since there is no need to purchase a variety and quantity of hardware from different vendors as part of the design process. Investing in network simulation over hardware results in significant savings in capital resources, engineering labor and operations (e.g. estimated savings for 100 managed network devices: 98% in capital resources; 96% in engineering labor; and 64% in operations). MILSIMNETs can be used as a testing ground for device hardware still in the evaluation phase so that when changes are made to a network design, it is not required to have the physical hardware in the network.

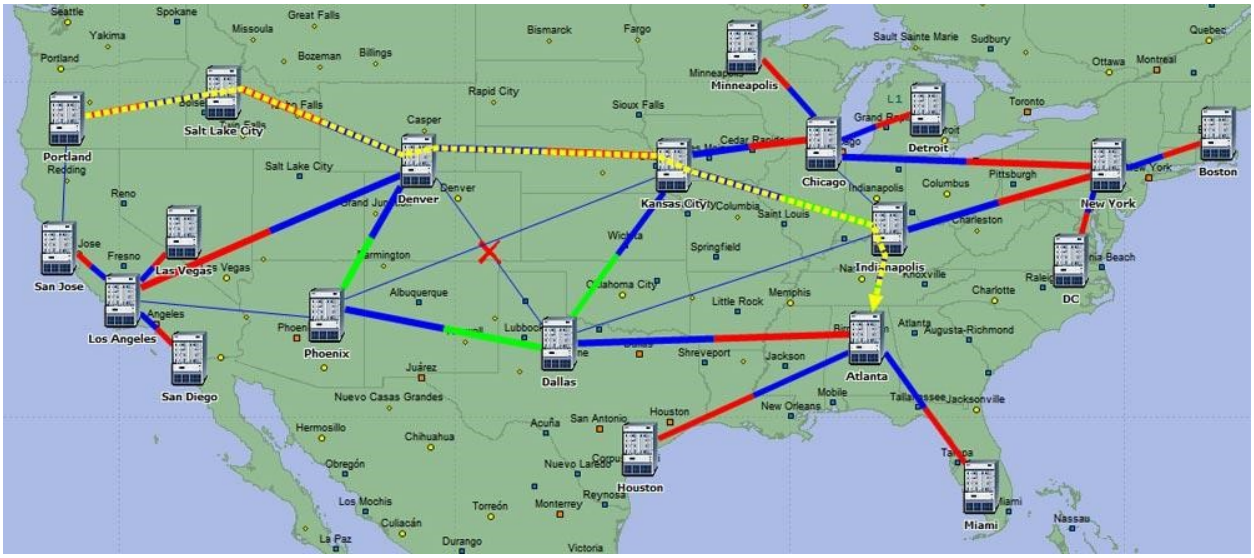
Simulation Network Environment Capabilities

Real-Time Fault Provisioning and Performance Management

Quickly install and configure new devices or entire network segments, individually or in groups. Schedule network "scenarios" that change a multitude of variables, start or shut down devices, or generate traps

and events. Multiple network management tools can query the same simulated devices, each with its own unique network view. Scalability of simulated networks is a key feature, here, in that the setup time required is considerably less than that of a physical network.

Network Configuration "capture and playback"



Easily reproduce actual network configurations. The MILSIMNET recorders can take snapshots of network objects and then "replay" them instantly in the MILSIMNET. Make controlled adjustments to variables to recreate problems or test configuration variations. It is possible to re-create a full production network in the MILSIMNET! With this methodology, important recreations, such as customer escalation or event issues, can be monitored and easily reproduced in such an environment.

Customization

With the intuitive graphical interface or flexible scripting language, you have complete control of the MILSIMNET. Change variable values, such as status, arrival rate and utilization rate for a single device or an entire group. Extend object structures to create prototypes of brand-new object and devices.

Extensibility

Supplied libraries provide out-of-the-box support for leading data network equipment vendors. In addition, a MIB Compiler imports new and custom MIBs to provide complete simulation of any SNMP compatible device management agent.

Scalability

A MILSIMNET supports many thousands of objects on a single server, thus enabling simulation for large-scale carrier and enterprise networks. Multiple MILSIMNET servers can supply an unlimited level of scalability.

Ease of use

A MILSIMNET provides both a graphical user interface (GUI) and scripting. Several application wizards in the GUI facilitate initial setup of the MILSIMNET, while scripting automates regression testing and demos. The goal is to present the entire MILSIMNET within a graphical user interface that is complete and descriptive.

Applications of Virtual Network Environments

Architecting New Networks

As discussed, today's utility telecommunications networks are undergoing the most extensive redesign seen in decades.

These changes are being driven by multiple factors:

CIP Compliance

Complying with the current (and future) Critical Infrastructure Protection (CIP) mandates from NERC and FERC are requiring utilities to adopt new design philosophies and implement new hardware on their networks.

Understanding the current network, what equipment is installed and how that network behaves is the fundamental building block for being able to start to build an effective CIP compliance plan. Using the MILSIMNET Recorder capabilities allows the MILSIMNET to "walk" the entire network, discovering all attached devices and building a permanent record of the baseline network.

Developing "CIP scripts" then allows network engineers to create critical infrastructure issues and see how the current network would react. These scripts can then be developed to model new designs and new equipment options on the simulation, without having to invest in the new equipment or impact the production network.

As network changes are implemented, new recordings of the network can be made and used to demonstrate CIP compliance, with scripts designed to show necessary test scenarios the network responding as required.

IP Transition

As the carriers discontinue offering the traditional TDM based services used by critical infrastructure in thousands of substations, the transition to IP based next generation networks has accelerated. As there is no single answer to replacing a TDM service, with geography, bandwidth and application support all being factors that impact the replacement technology for an individual substation.

Understanding what the options are and how they impact the network is very difficult without physically building test networks, which is costly, time consuming and resource intensive. Using the MILSIMNET simulators capability to add "virtual" devices to the MILSIMNET, enables multiple

test scenarios to be built and run in the simulator, without the need for physical equipment to be purchased and installed.

With a MILSIMNET, other products can be simulated and compared for operational characteristics and usability much more efficiently and faster. Examples of this are comparing switch devices. How does the device compare along the lines of switch management? Is the command line interface the same? These sorts of questions can be explored more efficiently with simulated network device hardware.

Support: Localize an environment

MILSIMNETs decrease the frequency and duration of visits to network sites. Problem environments can be recorded, and then replicated in a virtual network to help facilitate the efficient diagnosis and resolution of issues.

The support of network-based applications is a challenging business. As the use of Next Generation Networks continues to increase, the ability to troubleshoot issues has become more difficult than ever.

Fortunately, MILSIMNETs can assist technical personnel in determining root causes and reduce unnecessary visits to remote locations. A MILSIMNET can be used to record the network behavior. A simulation of the production network is then available for the technicians to troubleshoot – they can effectively see for themselves exactly how various applications and devices are interacting within the network and pinpoint the root cause of a problem.

One of the most time-consuming parts of diagnosing problems is reproducing it, which often requires setting up traffic generators and network analyzers. A MILSIMNET's ability to record a network allows technicians to 'replay' scenarios and re-create them virtually – without disturbing the network and setting up physical equipment. Actual scenarios can be run forward, backward, fast forward and fast backward at will – like a VCR! Imagine the time saving in diagnosing problems, especially when the problems are the type that appears at certain times – say every 30 days. With simulation there is no need to wait for the problem to reoccur – technicians simply fast forward 30 days, and the problem is there for them to work on!

Disaster Recovery Planning

The best approach to disaster recovery focuses primarily on planning and prevention and a MILSIMNET allows many typical disaster scenarios to be analyzed in detail.

Traditionally effective simulation of a network disaster such as a server farm going off-line, a router going out of service, a back-hoe cutting through the fiber optic cable down the road or a major terrorist attack has had to be done in one of two ways:

- on the live network out of hours and physically taking offline areas of the network to "see what the effect would be" or
- with pencil, paper and a slide rule trying to calculate what the knock-on effect would be.

Both scenarios are costly in terms of manpower and time, resulting in minimal what-if scenarios being tested. A MILSIMNET of the entire network can run unlimited what-if scenarios enabling a fully comprehensive disaster recovery plan being implemented that address all possible incidents.

Application Support & Capacity Planning

MILSIMNET technology is a key tool that can help predict the behavior and impact of new applications that will be deployed in the network environment. Running applications on a MILSIMNET illustrates how the application operates in a specific environment without the need to install it on the physical network. What-if scenarios allow network designers to see the impact on traffic and other applications as a variety of end user scenarios (such as hundreds of users all staring the application at once at the start of day, or large data input at specific times of day).

Engineering Training: Flexible Training Facility

MILSIMNETs alleviate the need for instructors to import hardware or connect to existing live networks. A MILSIMNET enables students to investigate routine and unforeseen network management scenarios via a network-in a-box.

Training is essential for keeping abreast of rapidly evolving technology, as well as for professional development, critical to ensuring the most effective learning experience is a sufficiently configured, state of-the-art lab. Hands-on training is also important for students to fully understand the intricacies of complex applications and network management solutions. Given the rapid pace of technological evolution, lab equipment quickly becomes obsolete. Therefore, maintaining an up-to-date training lab to satisfy expectations for meaningful and real-life exercises is costly, time-consuming and impractical.

The solution, once again, is to leverage MILSIMNET technology. MILSIMNETs can provide virtual devices that can be used to set up training lab networks. The virtual lab network becomes as extensive and complex as instructor's desire – allowing students to fully learn the capabilities of complex network software and network management applications. Having access to an unlimited number of virtual networking devices allows the trainer to properly represent complex application and network capabilities in labs and demonstrations. With a MILSIMNET, students get the opportunity to practice positive and negative 'real world' scenarios that would be impractical with a physical network.

Bringing it all together –MILSIMNET Network Environments in the Real World

The following are practical applications for a Virtual Network Environment and what benefits it can provide to individuals who are responsible for maintaining network connectivity.

- **Performance Management** – How is the Wide Area Network performing? What does it look like? How much data are you transferring from site to site or from site to HQ?

- **Configuration Management** – How are the routers and switches configured? Do you have the configurations backed up on a central server? Do you have secondary routers and switches online and ready to take over in case of a primary failure?

- **Accounting Management** – How are the local offices budgeted for high-speed leased line access? Are the costs incurred charged back to the local offices? How are these costs tracked?

- **Fault Management** – Are you aware when a router or switch experiences a problem? What scenarios do you have accounted for in simulation and what are there remedies? Do you have a methodology for training local IT people on these fault resolution scenarios?

- **Security Management** – Do you know who is accessing your network and what they are doing? What information are they seeing? How are passwords and user access maintained? Who has access to your switching and router closets?

One Liberty Lane Suite, Box 31 Hampton NH 03842
info@FAETelecom.com | www.FAETelecom.com |

Used with Permission of Tom Nolle – CIMI Corp from his blog.

Taking a Model-Side View of Ops Automation

Earlier this week I blogged about virtualization and infrastructure abstraction. I left the modeling aspect of that to the end because of my intentional focus on the abstraction side. Now I'd like to look at the same problem of virtualization from the modeling side, so see where we end up and whether conclusions from that direction support the conclusions you get when you start with the notion of an "abstraction layer".

Operations automation, whether we're talking about applications or service features, is usually based on one of two models that were popularized in the "DevOps" (Development/Operations) automation activity that started about a decade ago. One model, the oldest, is an extension of the "shell scripts" or "batch files" routinely used on computer systems to execute a repetitively used series of commands. This is now called the "prescriptive" or "imperative" model; it tells the system what to do explicitly. The other model, the "declarative" model, describes the goal state of a system and then takes automatic (and invisible) steps to achieve it.

What we call "intent modeling" today is an expansion of or variation to the declarative approach. With an intent model, an application or service is divided into functional elements, each of which is represented by a "black box" that has externally visible interfaces and properties, but whose interior processes are opaque. These elements don't represent the functionality of the pieces in the data plane, but rather the lifecycle processes associated with them, which we could call the management or control plane. If you tell an intent-modeled element to "deploy", for example, the interior processes (that, you recall, are invisible) do what's needed to accomplish that goal or "intent". Any element that meets the intent is equivalent to any other such element from the outside, so differences in configuration and infrastructure can be accommodated without impacting the model at the high level.

This raises a very important point in modeling, which is that the model represents an ***operations process set*** *associated with lifecycle automation. There are process steps that are defined by the model, and others that are embedded within model elements, opaque to the outside. Model elements in a good operations model could be hierarchical, meaning that inside a given element could be references to other elements. This hierarchical approach is how a general function ("Access-Element") might be decoded to a specific kind/technology/vendor element. In most cases, the bottom of any hierarchical "inverted tree" would be an element that decoded into an implementation rather than another model hierarchy.*
The qualifier "in most cases" here arises from the fact that there's an intrinsic limit to how low a model hierarchy can go. A management system API, which is probably the logical bottom layer of a model, could represent anything from the specific control of an individual device to a broad control of features of a device community. Remember the old OSI management model of element/network/service management layers? If the API you're manipulating in the end is a service-level API, then you can't expect to model devices—they're inside the bottom-level black box.

This means that there are really two levels of "automation" going on. One layer is represented by the model itself, which is under the control of the organization who writes/develops the model. The other is represented by the opaque processes inside the bottom model element, which is under the control of whoever wrote the implementation of those opaque processes. In our example, that would be the implementation of the service management API. Whatever isn't in the bottom layer must be in the top.

Another implication of this point is that if different products or vendors offer different levels of abstraction in their management system APIs, there might be a different modeling requirement for each. If Vendor A offers only element management access, then services must be created by manipulating individual elements, which means that the model would have to represent that. If Vendor B offers full service management, then the elements wouldn't even be visible in that vendor's management system and could not be modeled at the higher level.

This same issue occurs if you use a model above a software tool that offers a form of resource abstraction. The Apache Mesos technology that's the preferred enterprise solution for very large-scale virtualization and container deployments has a basic single-abstraction API for an arbitrarily large resource pool, and all the deployment and connection decisions are made within the Mesos layer. Obviously, you don't then describe how those decisions should be made within your higher model layer. If you don't use something like Mesos, though, you need to have the modeling take over those deployment/connection tasks.

The cloud community, who I believe are driving the bus on virtualization and lifecycle automation, have already come down on the resource abstraction side of this debate, which to me means that for service lifecycle automation the network operator community should be doing the same thing. That would align their work with the real development in the space. To do otherwise would be to expect that network/service lifecycle automation and application/cloud automation will follow different paths with different tools.

My dispute with the NFV ISG, with ONAP, with the ETSI ZTA stuff, revolves around this point. When we try to describe little details about how we'd pick the place to put some virtual function, we are discarding the experience of the cloud community and deliberately departing from their development path. Can the networking community then totally replicate the cloud's work, in a different way, and sustain that decision in an ever-changing market? Can they do that when a hosted application component and a hosted virtual function are the same thing every way except in nomenclature?

That so many in the operator world are now infatuated with "cloud-native" functions only make the separation of network functions and cloud applications sillier. How cloud-native are you if you decide to build your transformation on developments that are in little or no way related to what's going on in the cloud you're trying to be native to? It makes you wonder whether the term is anything more than fluff, an attempt to give new life to old and irrelevant initiatives.

We are, with developments like containers, Kubernetes, Mesos, Istio, and other cloud-related tools, reaching the point where the cloud has defined the ecosystem within which we should be considering future network operator transformation. Surely everything above the connection layer of services is more like the cloud than like the network. Much of what's inside the connection layer (the hosted

elements) are also cloud-like, and the way that connection/network devices relate to each other and to hosted components is totally cloud-like as well. Why the heck are we not using these tools? What ETSI should be doing now is not spawning new groups like ZTA, or continuing old groups that are only getting further off-track daily as the NFV ISG is. They should be aligning their goals to the cloud, period.

One critical step here is the tension between modeling and resource abstraction; in fact, abstraction in general. We could accomplish a lot even in the short term if we could convince vendors in the networking space to think about "abstraction APIs" and how these APIs could integrate with modeling tools like TOSCA. This kind of thing would not only benefit the network transformation efforts of operators, but also the cloud itself. Resource abstraction is increasingly a part of the evolution of cloud tools, after all.

Along the way, they can raise some valid issues for the cloud community. There is already cloud work being done in data-plane connectivity for hosted elements; it should be expanded to address the needs of virtualization of network data-plane elements like routers and switches. There are constraints to hosting location or to "proximity" of components to each other or to common failure sources that have been pointed out in networks and are only now becoming visible in cloud services. This can be a give-and-take, but only if everyone is part of the same initiatives. Let's get it together.

Contributing Authors:

Tom Nolle
President of CIMI Corporation
Sstrategic consulting firm specializing in telecommunications and data communications since 1982.

Dwight Linn
CEO & Founder
FAE Telecom
DwightL@FAETelecom.com

Dr John Pan
Board Member & Advisor - FAE Telecom, Inc
28 Years w/Bell Labs, MIT- PHD: EE, Co-Founder of Loop Telecom
Holder of over 100 Patents, last one issued in 2004

Edward A. Schowalter
VP Sales, FAE Telecom
EdwardS@FAETelecom.com