

Minimizing Mission Risks Through Emulating Space Communications Architectures

Rich Slywczak, Frances Lawas-Grodek, Diepchi Tran, Thong Luu, Allen Holtz, Brenda Ellis

NASA/Glenn Research Center (GRC)

21000 Brookpark Road, Cleveland Ohio 44135; (216) 433-3493

{Richard.A.Slywczak, Frances.J.Lawas-Grodek, Diepchi.T.Tran, Thong.Luu-1,
Allen.P.Holtz, Brenda.L.Ellis}@nasa.gov

ABSTRACT: The goal of launching breakthrough missions with a minimal amount of risk at a reasonable cost is achievable regardless whether the mission is large, such as NASA's International Space Station, or small, such as CHIPSat. To meet these goals, satellite missions must rely on new tools that detect any liabilities to the project during pre-launch testing.

NASA/Glenn Research Center (GRC) is currently developing an emulation testbed to assist missions with validating requirements and resolving issues, whether science or communication, before moving to an operational status. The Space Communications Emulation Facility (SCEF) will serve as a nationally accessible NASA facility. In the testbed, mission managers can emulate complete missions under typical space-based scenarios or researchers can emulate specific components of a satellite mission.

The goal of this paper is to explore SCEF by discussing the architecture of the hardware and software of the emulation testbed. In addition, the types of emulations and using SCEF to minimize risks will be highlighted.

SCEF will provide missions with the tools that they can use to resolve issues earlier than traditional methods. The end result will be a realization of savings, in time and money, as they move from mission concepts to launch.

INTRODUCTION

With the President's New Vision for Space Exploration¹, NASA has been aggressively addressing the goals of the agency for the next decade and beyond. Within the next twenty years, the United States will see the retirement of the Space Shuttle, the development of the Crew Exploration Vehicle and the return of man to the lunar surface. While developing and satisfying these goals, NASA will also focus on travel to Mars by both manned and robotic flights. During this time, the small satellite community will be able to play a pivotal role by developing missions that will use cutting edge technology with shorter schedules than the major institutions.

These new missions will be more complex and require new and diverse ways of managing space-based resources. In the near future, they will challenge the traditional thinking about typical satellite missions that simply record measurements and then transfer the data to predetermined ground stations. Communications between satellites will become essential to not only route data between them

but also to dynamically share information so that satellites can make more effective measurements in real time. But, before these missions will be successful, both a communications backbone and a series of ad-hoc networks must exist for these missions to exchange information. All missions, whether NASA, university or industry, must be able to connect and use these dynamic ad-hoc networks².

NASA/Glenn Research Center (GRC) is currently developing an emulation environment that will allow mission planners from government, universities and private industries to model satellite missions. The Space Communications Emulation Facility (SCEF) provides a multi-satellite communication testbed that researchers and mission planners can access as a tool to test and validate their designs during the early phases of development. Missions inside of SCEF are created by designing a scenario that is based on multiple parameters, such as the number of satellites within the mission, customized on-board satellite components, and a number of relay satellites and ground stations. During the emulation, complex high-end graphics displays not only satellite orbital positions but also the communication patterns between satellites and between satellites and ground

stations. SCEF has heritage in the Space Based Internet (SBI) which was originally developed by the University of Kansas³.

The objective of SCEF is really two-fold. First, it can emulate complete satellite mission architectures. During the emulation, mission planners can determine how effective the satellites communicate within themselves and amongst other satellites based on position and time. Secondly, researchers can emulate components of a satellite or aspects of a mission to improve on-board processing and communications. This will be a significant advantage for researchers, since it allows them to focus on their specific components while integrating their research into a typical mission and mission managers can adopt newer technologies sooner by knowing how they will function in a realistic environment.

While SCEF is an exciting new technology to plan and develop missions, it also provides another inherent benefit which is the mitigation of risks. Every project must handle risks and, usually, provide a risk mitigation plan so that most liabilities can be analyzed beforehand and have a prepared solution should they arise on the project. SCEF provides a tool to help minimize risks. While there are specific issues associated with each project, this paper will discuss a few generic liabilities that can be studied and mitigated through SCEF.

This paper will discuss both the hardware and the software architecture of the SCEF with special emphasis placed on the node architecture. The benefits of using an emulation systems and, specifically, SCEF are discussed along with the different types of emulations that a typical user can run in the environment. Tools, like SCEF, will be essential to develop these challenging new missions, since they will require new technologies that have not previously been integrated into satellites. It is through an emulation environment that these technologies will be validated and that missions will be able to launch sooner and within costs.

TYPES OF SATELLITE MISSIONS

As previously mentioned, emulation systems will help to define and validate these newer missions. This section will identify some of these missions and highlight the scope that will drive their development. These missions are as follows:

- *Single Satellite Missions:* Most current missions are single satellite missions, but the newer missions will need to extensively communicate with relay satellites which will support constant communications with ground stations. These communications requirements will enable scientists to retrieve data from the satellite at any time.
- *Constellations:* Researchers have been exploring the concept of having a group of satellites taking measurements together. The group will consist of a specified number of satellites which must cooperate and function as a unit. Their orbits will need to be studied along with the communications requirements between them and external satellites.
- *Ad-Hoc Networks:* Missions must contend with ad-hoc networks in space and on terrestrial surfaces. A terrestrial ad-hoc network might be a set of rovers on Mars; a space-based ad-hoc network might be two satellite sharing information from different missions.
- *Sensor-web:* Sensor web is a NASA concept where a number of different platforms containing sensors can be mixed into one network. It will contain space-based sensors, such as satellites; air-based sensors, such as airplanes and balloons; and land-based sensors, such as stationary measurement systems and rovers.

SCEF

The following section will provide a description of the SCEF architecture in terms of the software and hardware. The hardware environment will describe the architecture that will host the SCEF emulations system and the responsibilities of the machines. The software architecture will describe the major components of the software and describe the node architecture in detail.

HARDWARE ARCHITECTURE

The SCEF hardware environment is based on a cluster of computer systems on which the emulation system will execute; in the cluster, there are 32 nodes and 1 controller. A schematic of the hardware architecture can be seen in Figure 1. The 32 nodes will represent 31 satellites and a single ground station. The capabilities of the machines are as follows:

- *Controller:* The controller machines are 900 Mhz Pentium III class machines that contain 4 GB of memory. Each of the controllers contains 4 processors and about 200 GB of on-line storage. The controllers are responsible for running the emulation manager software and the commercial package Satellite ToolKit (STK) by Analytical Graphics.
- *Nodes:* The nodes (i.e., each satellite or ground station) are 3.06 Ghz Pentium IV class machines that contain 1 GB of memory. Each node contains a single processor and has 80 GB of on-line storage. The nodes are responsible for emulating the software components of the satellite with one node also serving as the operations node. Each node is compiled with a customized kernel to include Virtual Ethernet (Veth) devices and the latency and bit error rates (BERs) modules.

All machines are running the Fedora Core Linux operating system.

The above machines are connected through two separate networks, which are comprised of gigabit Ethernet connected to a Cisco Catalyst 4506 Switch. Each machine has two network interfaces that support two separate networks:

1. *Management Network:* The management network is used for communicating mission information to a node, in terms of orbital definition, resource allocation, and commands that are not part of the current mission scenario.
2. *Data Network:* The data network collects data from the emulation as they are transmitted to another relay satellite or ground station node. This network is where the data pipe size, bit errors, and latency, per definition from the commands via the management network, is applied to the transmitting data stream before they are relayed by the network.

The two networks are necessary to distinguish between the operations that command the emulation separate from the transmission of mission data being passed among the nodes. This allows for isolating what is occurring on a satellite and ground station (nodes on the data network) from what is controlling the emulation scenario (emulation manager on the management network).

SOFTWARE ARCHITECTURE

As described in the hardware architecture, SCEF is comprised of two major components, the controller and node software. To represent the space-based communications link between satellites, an Ethernet connection is implemented between each of the nodes and an IP-based protocol will serve as the packet format on this link. Emulating multiple antennas with a single physical Ethernet card is made possible by the use of Virtual Ethernet (Veth) Devices. A Veth device is a kernel modification that handles the assignment of one or more IP Addresses, each representing an antenna, to a single physical Ethernet device. Internally, when data for a Veth device arrives at the machine with a virtual IP address, the network layer will route the packets back through the physical device with the physical IP address for normal processing. The Veth device provides a convenient and flexible mechanism for emulating multiple IP-based devices on a single computer without installing multiple physical Ethernet cards. Veth devices can be assigned on the “fly” during the emulation execution.

The SCEF software architecture, as shown in Figure 2, will consist of the following modules:

- 1 Hardware Environment: Controller
 - a) Emulation Manager – provides a GUI for user input/output and controls the flow of information between the nodes and the controller machine.
 - b) Node Emulation – responsible for resource allocations and assigning Veth devices.
 - c) Communication Emulation – contains the code for the Veth devices and the latency, BERs and Quality of Service (QoS) modules to emulate space communications.
- 2 Hardware Environment: Node
 - a) Operations Node – software modules that permits the central operational node to control the orbits, adaptive routing, and instruments scheduling.
 - b) Satellite Node – software that resides on each satellite or ground station node.

Emulation Manager

The emulation manager provides the interface to SCEF, as shown in Figure 3; the user can select the specific scenario file and can start/stop the emulation through the Graphical User Interface (GUI). In addition, it will provide a number of output reports including period, position, range, access and lighting data. As part of the graphical output, the user can see an emulation of the satellite orbiting the Earth in 2D

and 3D graphics. The reports and graphics are provided by STK, a commercially available package.

The emulation manager also contains a number of modules that will provide communication services to the nodes. Some of the modules are as follows:

- *Operations Channel Module*: responsible for the communication between the Operations Node (which contains the Instrument Scheduling and Routing) and the Nodes. It passes Instrument Scheduling and Routing information to the appropriate individual node.
- *Node Control & Configuration Module*: selects and configures nodes for the specified scenario; communicates with the node emulation software to control the emulation such as sending dynamic attribute updates to all nodes; and collects network traffic statistics from all nodes.
- *Emulated Operation Channel*: serves as a medium between the Instrument Scheduling and Routing Software in the Operations Nodes and the node software in each node. It passes the Instrument Scheduling and Routing information from the Operation software to each of the nodes.
- *Operation Attribute Interface*: communicates with the Operations Node Attributes to receive requests of attribute updates and sends the results obtained from STK to each of the nodes.

Node Emulation

The node emulation software provides client services to the Emulation Manager. It runs as a daemon and provides services during the lifetime of the emulation. It consists of two modules:

- *Node Control Module*: responsible for communicating with the Emulation Manager via messages to configure the network environment (i.e creating and configuring the physical Ethernet devices, Virtual Ethernet devices, and the QoS services) and to start/stop the Node program
- *Networking Statistics Module*: responsible for gathering networking statistics from the Communication Emulation Module and sending them to the Emulation Manager.

Communication Emulation

The communication emulation software contains the required modification for the Linux kernel to support Virtual Ethernet devices and additional QoS elements, such as bit error rate, delay, etc. These modifications are compiled into the kernel and will be available on the nodes when the emulation is starting. If the services are desired they must be specified in the scenario file during the setup of the emulation.

Operations Node

The Operations Node is responsible for controlling all satellite operations. The Operation Node computes orbit information, determines network topology, configures the routing table of each node, and computes when instruments on satellites should be turned on or off to collect data. The central operation software is comprised of the following three modules:

- *Operation Attributes*: queries the satellite attributes information from STK via the Emulation Manager and passes this information to the Instrument Scheduling and Routing modules.
- *Operation Instrument Scheduling*: responsible for determining when the satellite instrument should be activated and passes this information to the satellite nodes.
- *Operation Routing*: periodically computes the topology of the scenario and the routing tables for each node and passes them to the individual nodes.

Satellite Node

The Satellite Node provides an emulation of a satellite customized to the user specification in the scenario file. The architecture of a typical node is shown in Figure 4. When the Emulation Manager executes a scenario, a user specified number of satellites are created and inserted into the emulation with orbital elements that will define the desired trajectories. The satellite node will represent one of the satellites defined in that particular scenario. The overall goal of the satellite node is to support the transition from centralized satellite operations and limited on-board decision making to autonomous satellite operations and enhanced on-board decision making.

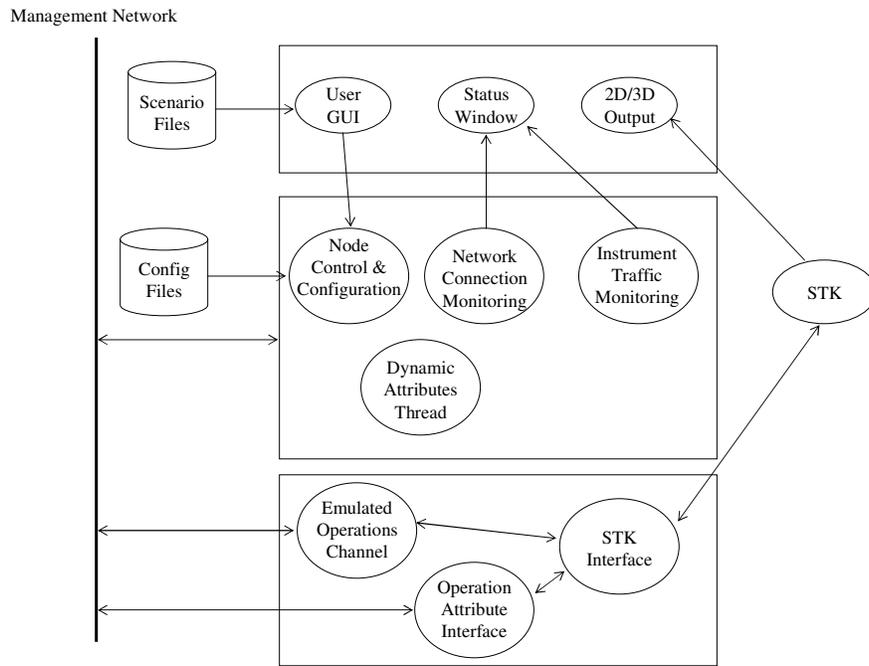


Figure 3. Emulation Manager Architecture

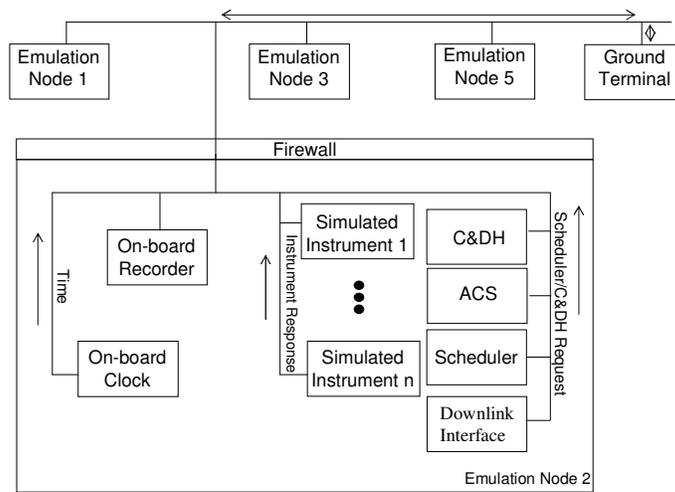


Figure 4. Satellite Node Architecture

The components that are part of the emulated satellite node and their default operations are as follows:

- *On-board Clock*: responsible for maintaining an accurate centralized time source. Any process on the satellite bus can query the clock for the current time.
Default Operation: retrieves the current time from the host computer and reports the time back to the querying process.
- *Simulated Instruments*: defines a simulated science instrument that will record random “measurements”. The instrument will simulate a science instrument on-board the satellite.
Default Operation: creates a stream of random data that conforms to the data boundaries specified by the user. The stream will be maintained in memory until it is transferred to the on-board recorder or returned to the querying process.
- *Command and Data Handling (C&DH)*: provides a controlling mechanism for satellite operations and executes commands send from the ground.
Default Operation: implements a version of satellite quality C&DH software that is currently under development.
- *Antenna and Control System (ACS)*: responsible for controlling the antennas on the satellite. Any process can request an antenna change, but the C&DH along with the Scheduler will make the decision of which changes and, at the specific time, they can happen.
Default Operation: based on the user input that describes the antenna operations. The process will simulate an antenna movement and then it can be queried to determine the state of the request.
- *Scheduler*: responsible for executing the commands in the queue for the satellite or science instruments.
Default Operation: default scheduler will execute commands as they are found in the queue at the specific time indicated by the command.
- *Uplink/Downlink Interface*: responsible for transferring data from either the satellite to the ground station or from the ground station to the satellite. It will implement the data rate for the transfer that was set by the user.
Default Operation: downlink interface will emulate transferring the data from the satellite to a ground station at a data rate

specified by the user. The requesting process can query the downlink interface to determine whether the transfer has been completed.

- *On-board Recorder*: responsible for storing the measurements taken by the instruments.
Default Operation: centralized recorder where the science and satellite instruments will pass packets of data and have them stored until a request for download has been received.

The satellite node is designed so that each component will be an individual process or, in other words, each component will be individually addressable by an IP address. As shown in the figure, the communications aboard the satellite will be based on the TCP/IP protocol. This will promote the idea of the “Internet in Space” or “IP in Space”⁴ concept that is envisioned for future missions. Each of the components will be emulated by a UNIX process and function as a service. For example, if the scheduler needs access to the time, it queries the on-board clock for the time. Another example would be for an instrument to request an antenna to turn a certain number of degrees. Since the antenna might turn slowly or the request could be postponed given more urgent requirements, the instrument would need to re-query the ACS component to determine when the antenna has completed the turn.

TYPES OF EMULATED SCENARIOS

One objective of SCEF is to emulate complete mission scenarios under conditions that the satellites will encounter when they become launch ready. But, SCEF has another, more fundamental, objective which is to emulate any component on the satellite for research or mission purposes. This section will provide a brief description of some of the categories that can be emulated in SCEF. These emulation scenarios can be summarized, as follows:

- *Mission Scenarios*: Each satellite-based mission has its own special requirements and the testbed permits the definition of different mission scenarios based on orbital parameters such as semi-major axis, eccentricity, inclination, argument of perigee, right ascension of ascending node (RAAN), epoch time, and orbital period for each satellite node. With these parameters, mission planners can create and emulate scenarios which include multiple or single GEO, MEO, or LEO satellites.
- *Research Algorithms*: The testbed provides the flexibility for researchers to replace and/or modify algorithms on-board the satellite. Such algorithms include command and data handling,

data storage, antenna control, clocking, and scheduling. If an algorithm is not modified, the default version will run upon algorithm execution.

- *Security:* With the use of the IP-based protocols, security will become a major component in satellite development. Since the testbed uses the common network protocol IP, any advanced security features (eg, IPSEC⁵, VPNs⁶) can be integrated into the testbed.
- *Communications:* Communications is becoming an important research topic as researchers seek out flexible and seamless communication mechanisms among satellites. The testbed allows modifying protocols stacks and testing with advanced protocols. For example, to improve the throughput, TCP⁷ with modified congestion control algorithms or the rate-based protocol in the SCPS⁸ protocol suit may be implemented.
- *Networking:* Routing in space is a challenging area for networking due to the dynamics of the environment. Users have access to a combination of minimum spanning tree and link limiting algorithms with the Discrete Time Dynamic Virtual Topology Routing (DT-DVTR) method as the default routing or can integrate and test a customized routing algorithm.

MINIMIZING RISKS THROUGH EMULATION

Emulation can be a very powerful tool which combines the advantages of simulation with protocols, measurements, etc. tested under a real system workload⁹. When testing a theory or scenario involving space missions, it will become essential to perform laboratory testing to produce more effective and efficient results. An environment that could lend itself to extensive testing can reduce costs and schedule.

Space has always been a risky environment given that, once most missions launch, there are very few opportunities to replace or modify components. One modification to satellites that can be implemented from the ground concerns software problems. The option exists to upload new revisions of the software providing that a catastrophic error has not occurred. SCEF provides two significant advantages. The first is that satellites can be tested under normal scenarios to determine if there are any problems with the normal day-to-day operations of the mission. The

second is that they can also be tested under anonymous scenarios to develop a risk mitigation plan. Through the plan, the missions can be prepared to handle problems that might arise during the mission with a tested solution through the emulation environment.

While risks can be specific to an individual mission, this section will highlight a few of them that can be applied to a number of missions. These can be summarized, as follows:

- *Satellites that don't communicate:* Already highlighted in this paper, communications will become one of the key issues in future missions. They will need to operate seamlessly and constantly with other missions and ground stations. Routing in space and relay satellites will become significant players in the communications segment. Without constant and cooperative communications, future missions will not be considered successful.
- *Satellites/Constellations that have inaccurate orbits or orbital information:* Of the different missions types, constellations will play a major role in future missions. There can be two different types of constellations. First, in the loosely coupled arrangement, the satellites will act autonomously and collect data to have it processed on the ground. Secondly, in a tightly couple arrangement, satellites must have knowledge of the other satellites and exchange orbital information. This will ensure that they are taking measurements in concert with each other. The orbits/orbital information of these satellites must be finalized before launch to ensure that they can have adequate ground coverage and satisfy mission objectives once they launch.
- *Satellite functionality in deep space:* The satellite community is becoming more interested in deep space missions. But, there is a host of risks that can occur in deep space and these issues must be resolved before launch. For example, a mission might want to study landing characteristics or de-orbit burns which are significant issues.
- *Using new components/technologies on satellites:* There is concern and hesitation to adopt new technology on satellites over the "tried and true". How this technology will function in an environment far removed from our manual involvement and intervention presents a significant risk.

While there are risks to satellite missions, an emulation system can help to minimize these and develop solutions

for problems, if needed. Specifically, SCEF has a number of characteristics that can be useful, as follows:

- *Run the actual code/components in the environment:* In SCEF, a mission can insert actual code into the emulation environment. For example, if a researcher develops an improved on-board clock or scheduling algorithm, a mission can validate the component by testing the actual code. While there might be some interfacing routines to the rest of SCEF, this is probably the most direct method of exercising the actual code prior to launch.
- *Model satellites based on orbital parameters:* When satellites are specified in the emulation environment, they can be specified by their orbital parameters (e.g., semi-major axis, RAAN, argument of perigee, etc.). During the emulation, missions can determine how, and if, these orbits will impact the mission.
- *Model missions based on space environment parameters:* The space environment can be modeled based on QoS parameters, such as latency and bit error rates. Since these parameters can become large at long distances, such as Lunar or Mars, projects will be able to study issues such as communications¹⁰.

FUTURE WORK

NASA/GRC has been aggressively working on the SCEF emulation system and this paper has outlined the results of that effort. While a significant amount of the code has been implemented, SCEF is still considered beta software where testing is occurring but there are still bugs in the system. An environment like SCEF will never be completed due to the complexity of the software and the upgrades for new missions and requirements; it will continue to be an evolving system. There are a number of upgrades that are continuing as follows:

- *Expand the node definition:* The goal is to have the satellite node look as realistic as possible. There are still a number of components to implement (e.g., power subsystems, up-link interfaces, etc) and some of the current implementation are rudimentary. They serve as a first implementation of the subsystems.
- *Improve the integration of research algorithms:* The SCEF team is continually looking for better ways to integrate

customized code into SCEF. Currently, SCEF has defined an Application Programmers Interface (API) to integrate the customized code into the environment. The goal is to make the integration as seamless as possible and have the integrator incur as little overhead as possible during the integration process.

- *Improved user interfaces (GUIs):* Most software developers believe that they can improve the GUIs for a software system and the same is true for SCEF. While a GUI does exist, it does not provide an interactive point-and-click method for developing scenarios. As more testers use the systems and provide feedback, the GUI will become a more polished product.
- *Provide a remote access capability:* Currently all scenarios must be defined and executed on the computer cluster at NASA/GRC. In the future, it would be ideal to have an organization attach a computer to the network and, with a little configuration, be a member of the cluster. They could, essentially, define a satellite on their local system and be part of a constellation where the rest of the members could be running either on the cluster at NASA/GRC or remotely.
- *Dynamic allocation support across missions:* Currently, most of the work in SCEF is oriented within a specific mission. An infrastructure for cross mission sharing of resources and allocations does not exist. The goal is to provide a secure and efficient utilization, fair bandwidth allocation and dynamic support for a large number of missions.
- *Rate control testbed nodes:* The SCEF testbed is comprised of PCs that have the latest technology with CPU speeds of 3.06 GHz. But, this does not reflect the typical CPU power found on satellites. A study will determine whether it is possible to rate control the nodes so that they can represent the typical speeds found on-board satellites. This could be user settable.

CONCLUSION

The goal of this paper was to outline SCEF, the emulation system that is currently under development at NASA/GRC. Projects will rely on emulation systems to determine how satellites, systems of satellites and basic components will adapt to the space environment. As outer space continues to be explored and private enterprises become part of the community, there must be a mechanism to test these systems before they are ready to launch.

Projects will use emulation systems to aid in reducing the foreseen liabilities in the project and test missions and components in the mission once they become operational. SCEF provides a number of benefits. Being able to integrate custom, and missions ready, code into the environment will be a tremendous benefit, since missions will be able to see how the code used in the project will function with mission parameters. In addition, being able to define satellites based on orbital parameters and missions based on space environment parameters will provide a realistic emulation run. Each of these provides benefits to the missions.

NASA/GRC is continually developing and improving the SCEF software by testing with new missions. While the software is still considered beta, SCEF is looking for missions and researchers that would like to test in the emulation environment. Other projects will provide the most benefits to SCEF by testing the software and determining where upgrades are needed and features must be added. SCEF will continue to be the system which future missions use to validate their requirements and goals, but missions must also consider SCEF to be a tool that will help them meet their requirements and encounter as few problems as possible and enjoy a successful mission.

REFERENCES

1. New Vision for the Space Exploration Program: Remarks by the President on U.S. Space Policy, NASA Headquarters, Washington, D.C., January 14, 2004.
2. Bhasin K. and Hayden J.L., "Developing Architectures and Technologies for an Evolvable NASA Space Communication Infrastructure", AIAA, ICSSC, May 12, 2004
3. Minden G., Evens J., Searl L. and Wagh P., "Architecture and Design for a Space Based Internet Emulation System", University of Kansas.
4. Operating Missions as Nodes on the Internet (OMNI), <http://ipinspace.gsfc.nasa.gov>, Accessed June 14, 2004
5. Kent S. and Atkinson R., "Security Architecture for the Internet Protocol", RFC 2401, IETF, November 1998.
6. Gleeson B., Lin A., Heinanen J., Armitage G. and Malis A., "A Framework for IP Based Virtual Private Networks", RFC 2764, IETF, February 2000

7. Postel J., "Transmission Control Protocol", DARPA Internet Program Protocol, RFC 793, 1981.

8. Space Communications Protocol Standards, <http://www.scps.org>, Accessed June 14, 2004.

9. Svobodova L. and Mattson R., "The Role of Emulation in Performance Measurement and Evaluation", Proceedings of the 1976 ACM SIGMETRICS Conference on Computer Performance Modeling Measurement and Evaluation.

10. Allman M. and Eddy W., "Advantages of Parallel Processing and the Effects of Communications Time", NASA.