Technology Developments Integrating a Space Network Communications Testbed

Winston Kwong[*], Esther Jennings[†], Loren Clare[‡], Dee Leang[§]
Jet Propulsion Laboratory/Caltech, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

As future manned and robotic space explorations missions involve more complex systems, it is essential to verify, validate, and optimize such systems through simulation and emulation in a low cost testbed environment. The goal of such a testbed is to perform detailed testing of advanced space and ground communications networks, technologies, and client applications that are essential for future space exploration missions. We describe the development of new technologies enhancing our Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation (MACHETE) that enables its integration in a distributed space communications testbed. MACHETE combines orbital modeling, link analysis, and protocol and service modeling to quantify system performance based on comprehensive considerations of different aspects of space missions. It can simulate entire networks and can interface with external (testbed) systems. The key technology developments enabling the integration of MACHETE into a distributed testbed are the Monitor and Control module and the QualNet IP Network Emulator module. Specifically, the Monitor and Control module uses web services interface mechanism to centralize the management of testbed components. The QualNet IP Network Emulator module allows externally generated network traffic to be passed through MACHETE to experience simulated network behaviors such as propagation delay, data loss, orbital effects and other communications characteristics, including entire network behaviors. We report a successful integration of MACHETE with a space communication testbed modeling a lunar exploration scenario.

I. Introduction

IN the vision of future manned and robotic exploration missions to the Moon, Mars and beyond, it is essential that NASA develop a low cost method to validate and verify communications architectures for a robust and continuously available space communications network. Towards this goal, the Exploration Systems Research and Technology Program funded the initial phase of the Space Communications Testbed (SCT) project that started in April 2005. JPL's involvement in this project ended in late 2005. The SCT team comprised Viasat Laboratories as the project lead; the other team members were Glenn Research Center (GRC), Jet Propulsion Laboratory (JPL), Goddard Space Flight Center (GSFC), and Langley Research Center (LaRC).

The SCT concept is a geographically distributed system initially located across Laboratories and the other participating NASA Centers. Such a testbed is needed to perform detailed testing of advanced space and ground communications networks, technologies and client applications that are essential for future exploration missions. The testbed allows combinations of real and emulated equipment to support end-to-end performance and functional testing of space communications and networking such as interplanetary links, in-space and surface operations, intra-vehicle networks, and deep-space navigation. The testbed also supports interoperability of both custom and COTS hardware and software used to emulate space networks.

---

[*] Technical Staff, Digital signal Processing Research Group, Communications Architectures & Research Section.

[†] Communications Networks Systems Engineer, Communications Networks Group, Communications Architectures & Research Section.

[‡] Group Supervisor, Communications Networks Group, Communications Architectures & Research Section.

[§] Software Engineer, Flight Software Applications & Data Product Management Group, Flight Software & Data Systems Section.

Expected benefits from the use of SCT include flexible platforms to evaluate Earth ground systems, Crew Exploration Vehicles (CEV), launch vehicles, ISS, payloads, Lunar and Mars rovers and surface assets, Extra Vehicular Activity (EVA), Earth/Lunar/Mars relay satellites, MCCs, lunar/Mars bases, validation and refinement of assumed requirements of space network architectures, reduced mission risk, and testing of interoperability of “plug-and-play” communications and navigation systems.

There are two main components in the SCT architecture: the Emulation Subsystem and the Monitor and Control Subsystem. The Emulation Subsystem contains the emulation of real-world functional components as well as communications links (space and ground) among the components. The Monitor and Control Subsystem provides management functions of the testbed. The Monitor and Control Subsystem is a hierarchical structure with a Master Controller that interfaces with the emulated components through messaging services. It manages processes to configure, control, report, and store qualitative data for the SCT system. The structure utilizes web services and database technologies for messaging, data collection, and system status reporting.

II. Network Simulation Tool

At JPL, we have developed a Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation[2] (MACHETE) tool for analyzing the performance of existing and emerging communications protocols and services in the context of space exploration. The basic software architecture for MACHETE consists of four general systems: (1) orbital and planetary motion kinematics modeling, (2) link engineering modeling, (3) traffic load generation and protocol state machine modeling and execution, and (4) user interface systems spanning all of these three core elements. The resulting combination provides an essential tool for quantifying system performance based on comprehensive considerations of different aspects of space missions. Using this tool, technology researchers and mission designers can (1) determine system resource requirements such as bandwidth, buffer size, and schedule allocations, (2) characterize performance benefits of new or alternative protocols, services, and operations, (3) validate new technologies for mission infusion, and (4) enhance mission planning and operations.

At the core of the MACHETE network simulator is a discrete event simulator QualNet[**]. QualNet is the commercial product of GloMoSim and was developed as part of the DARPA Global Mobile communications networking project. QualNet contains a full contingent of conventional protocols, such as the IEEE 802.11/WiFi and Internet protocol standards. JPL models space protocols with the QualNet development environment and modelling toolsets. The protocols include the complete CCSDS protocol stack: Proximity-1, Packet Telemetry and Telecommand (TM/TC), AOS Space Data Link Protocol, Space Communications Protocol Specification (SCPS) and CCSDS File Delivery Protocol (CFDP). The most recent additions are models for Bundle Protocol (BP) and Licklider Transmission Protocol (LTP), built according to specifications of IETF drafts.

The integration functionality of MACHETE makes it a useful tool for space networking technology research and mission design. Currently, MACHETE provides two external interfaces for network simulation: IP packet sniffing using the IP Network Emulator (IPNE) and TCP/IP using a TCP-bridge. In SCT, we integrated MACHETE using the IP Network Emulator.

III. Testbed Implementation and Benchmarking

A. External Data Interfaces

As aforementioned, MACHETE supports two external interfaces, either at the transport layer through a TCP-bridge or at the network layer through the IP Network Emulator. The TCP-bridge is a software application that connects testbed components together using TCP/IP with a custom application protocol. The advantages of using the TCP-bridge are: TCP provides reliable transmission and the TCP-bridge is relatively easier to use in a testbed. The disadvantages of using a TCP-bridge are: the applications program must adhere to the custom bridging protocol (additional header information etc.), and the organization of the testbed (i.e. the way testbed components can be connected together) is limited. TCP-bridge uses sockets to connect to external entities and the nature of the TCP protocol provides reliable end-to-end transport. In addition, a custom application protocol allows metadata such as timestamps to synchronize data flow in the testbed. However, to connect components into the testbed, they must be equipped with this custom application protocol. COTS software cannot be used out-of-the-box without modification.

The advantages of interfacing at the IP layer are: compatibility with any application running IP, and the ability to accommodate a vast number of testbed network topologies. The disadvantages of interfacing at the IP layer are:

[**] QualNet software is commercial software developed and maintained by Scalable Network Technologies, Inc.

best-effort transmission (packet loss may occur), and network configuration settings can be complex depending on the testbed organization (i.e. require routing table modifications). The IP-based architecture allows virtually any IP-ready hardware or software application to interface with SCT. Companies who wish to test their IP-ready device in a space environment can plug into the data plane network and pass data through the emulated space network. Such hardware devices may include routers, switches, channel emulators, VOIP devices, and radios. Similarly, software applications designed to run over the Internet can be used without modification. In our testing, we have used Netmeeting[3], VLC[4], and IPerf[5] as end-to-end applications in generating external traffic through a simulated space scenario.

In SCT, we chose the IP interface design because there were hardware-only components such as routers where only the lowest three layers (physical, data-link and network layers) are implemented. Furthermore, the IP interface enables SCT to interface with the large number of hardware and software components that already exist.

B. Messaging Service

For remote monitor and control, a messaging service is needed. Web services provide a flexible mechanism to manage the testbed components. Since components may have different sets of management functions, web services allow those functions to be customized and tailored to specific requirements. Through the use of Simple Object Access Protocol[6] (SOAP) and XML, the management server feeds and listens for data messages to and from the testbed components. Changes to web services interfaces can be published to directories, which provide a dynamic and flexible infrastructure for sharing web service interface descriptions between the distributed components in the testbed. Most importantly, web services provide an interface mechanism to support centralized testbed monitor and control operation.

C. Network Synchronization

A common challenge to distributed systems is synchronization. SCT mitigates this problem by allowing testbed components to be started and stopped synchronously with web services. However, there are currently no tight synchronization mechanisms (lock-steps) to maintain testbed synchronization during runtime. Several difficulties arise in runtime synchronization. The most prominent asynchrony stems from the non-deterministic arrival of IP packets in the data plane. IP packets delivered to the data plane are not time synchronized with the simulation. The inherent asynchronous property of IP packet flow implies that simulation runs will be different and it is improbable to repeat the experiment where all the packets will be delivered in the same order or time sequence. Assuming the testbed network is isolated and no component is a performance bottleneck, then, packet jitter, delays, drops, and other errors should be small enough to be averaged out over a large number of simulation runs.

D. Testbed Equipment Performance Limitation Measurement

In general, systems produce errors (noise) that aggregate with output results. The error levels may or may not be acceptable depending on the required tolerance levels. Thus it is imperative to measure the inherent error levels of testbed components to ensure results are meaningful. The performance limits of the testbed components (i.e., network interface cards, processor speed, network bandwidth, software, etc) impose a limit on achievable data throughput in simulation. If this limit is exceeded, the performance will degrade (e.g. dropping data packets to produce a higher error floor in the simulation results).

We attempted to characterize the effect of MACHETE on network throughput by comparing the throughput of the network without MACHETE (data not passing through MACHETE) with the throughput achieved by passing data through MACHETE. We used IPerf to measure both TCP and UDP throughput to independently (independent from SCT) analyze the performance capacity of the MACHETE system. MACHETE was installed on a Pentium 4, 1.6 GHz, Windows 2000 workstation with 2 GB of memory. A virtual network scenario consisting of two nodes connected by a 1

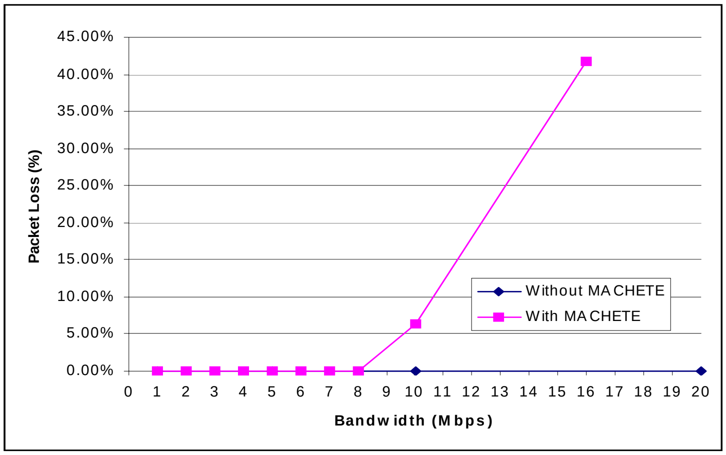

Figure 1. IPerf UDP Packet Loss

GB Ethernet link is loaded into MACHETE. The IPerf server and client are run on two separate, identically equipped machines. The client is configured to connect and send data to the server through the virtual network in MACHETE. Since the bandwidth of the virtual network (1 Gbit) is larger than the physical testbed (100 Mbit) network, only the MACHETE system performance, and not the virtual network performance, is measured. All tests are run with the default IPerf options except measurements are extended to one minute (from the default 10 seconds). In addition, in the UDP tests, “-b <bandwidth>” option was used to vary (limit) the UDP bandwidth.

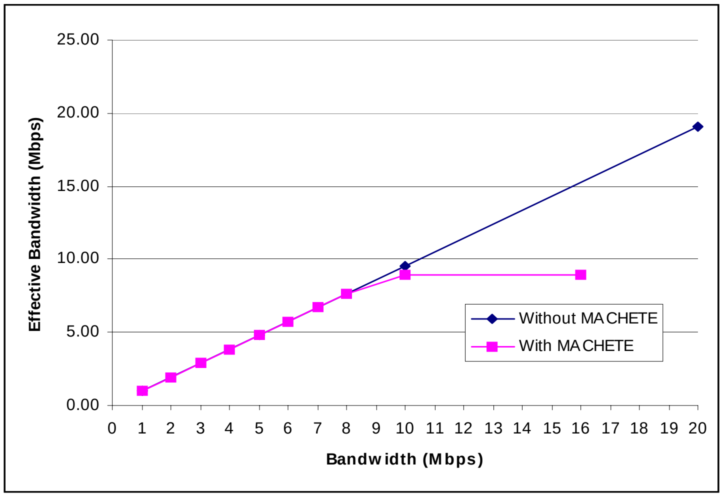

Figure 2. IPerf UDP Effective Bandwidth

The first test obtained a baseline performance without MACHETE. The IPerf server and client are connected directly using TCP and the bandwidth was measured to be 94 Mbps. This number is consistent with the test setup since the machines are on a 100 Mbit network. When the server and client connect through MACHETE, TCP throughput dropped significantly to 5.93 Mbps. A possible explanation for the sharp decline in bandwidth may be due to TCP’s reliable transport property. Then, we ran UDP transfers to measure throughput and observe packet loss[††]. Figure 1 shows that UDP suffers significant packet loss when IPerf data traffic passes through MACHETE. Without MACHETE, there is no significant packet loss when bandwidth is increased to 20 Mbps; but with MACHETE in the system, packet loss becomes significant beyond the 8 Mbps bandwidth.

From these experiments, we observe that both the TCP and UDP effective bandwidths are affected by MACHETE. For UDP without MACHETE, the effective bandwidth matches the bandwidth set with the “-b” option[‡‡]. The effect of packet loss starts at 8 Mbps which correlates to the point where the effective bandwidth approaches a ceiling. Figure 2 shows that the UDP throughput is affected by MACHETE.

IV. Experiment Scenario and Setup

The testbed was set up to demonstrate communications networking experimentation for the NASA Exploration Systems Mission Directorate (ESMD). Four elements were used to construct the Earth to Moon network scenario: Deep Space Network (DSN), Lunar Relay Satellites (LRS), Crew Exploration Vehicle (CEV), Lunar Proximity Network (LPN). The four elements were connected to form the network shown in Figure 3. Each line represents a communication link between node pairs. MACHETE is responsible for simulating the LPN including the links between LRS to CEV (2), LRS and LRS (4), and LPN and LRS (5). LPN includes several surface nodes as shown in Figure 4.

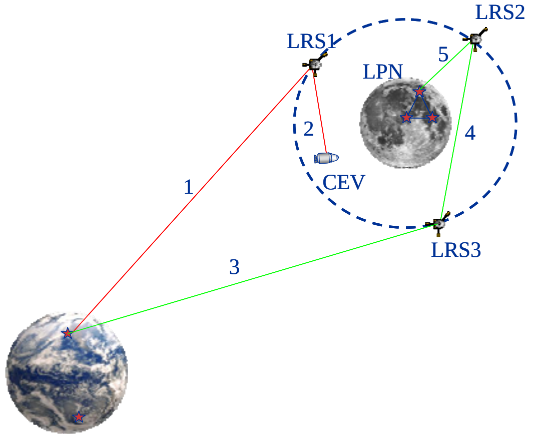

Figure 3. Earth to Moon Network Scenario

The test scenario involved two external end-to-end communication paths. An astronaut on the lunar surface will send video to Earth. Data will originate from the moon and travel to LRS2, and relayed to Earth. The green path (EVA3 to MOC) in Figure 4 illustrates this path. The second communication path is between a Lunar Roving Vehicle (LRV) and CEV orbiting the moon. They will run a CFDP[7] application for file transfer. This is shown as the red path (LRV2 to CEV) in Figure 4. MACHETE

[††] The datagram size is 1470 (default value), small enough to fit into a single packet to measure packet loss.
[‡‡] When the IPerf server and client are connected directly without MACHETE, we observed a bandwidth ceiling at 65 Mbps.

simulated multi-hop paths running different protocols over each. IEEE 802.11 is used between lunar-to-lunar surface elements, Proximity-1 is used between a lunar surface element and LRS, and AOS is used between LRS and DSN.

The topology of the testbed network is shown in Figure 5. Each rectangular box in the figure represents a physical machine. The Link Emulator (LE) simulates the communication characteristic between LRS and DSN. A machine (VLC_RX) is connected to the LE to create Network A. This machine represents the end users on Earth running one end of the video application[§§]. Network B consists of MACHETE, VLC_TX, CFDP_TX, and CFDP_RX. MACHETE will be simulating the various links on and around the moon. VLC_TX will run the other end of the video application by streaming data to Earth. CFDP_TX and CFDP_RX are end-to-end applications for the LRV to CEV link.

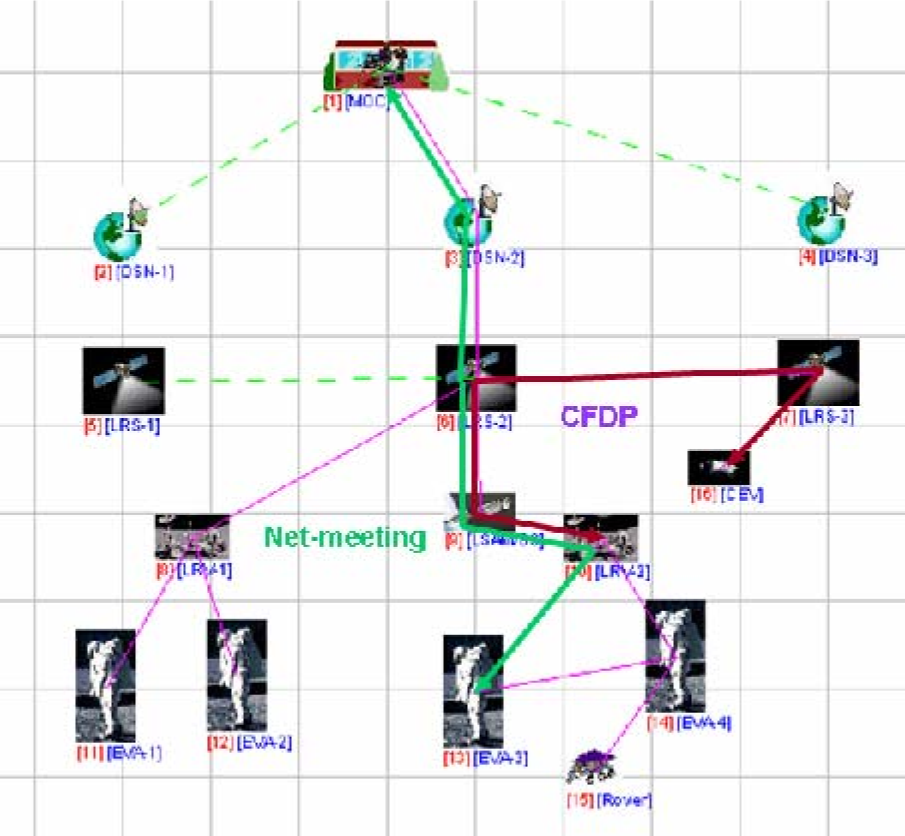

Figure 4. Communication paths for SCT experiment

There are two separate planes in the SCT network: data traffic plane, and monitor and control (MC) plane. The MC plane (Network C) provides a path for testbed management through web services. Currently MACHETE can handle start and stop commands as well as statistics collection. Logically, we separate the data and MC planes to prevent their operation from interfering with each other. Also, web services facilitate a common interface to be developed for all components on the MC plane.

The data plane is the path for data traffic exchange between testbed components. The end-to-end applications (VLC and CFDP) will generate the data to be used in the testbed. The interface to the data plane is the IP protocol. In other words, the basic element of transport in the data plane is the IP packet. This architecture enables IP-ready hardware systems to plug-in easily to the testbed.

MACHETE interfaces with the testbed data plane with a QualNet module called IP Network Emulator (IPNE)[***]. Each network interface inside MACHETE is assigned a virtual IP address. MACHETE is configured to map corresponding IP addresses in the real-world to virtual IP addresses in the simulation-world. By addressing data to virtual IP addresses, IPNE can sniff data packets, inject them into MACHETE for simulation, and passed back onto the network. The operation is transparent to the source and destination applications except for the

---

[§§] The VLC media player application is used to stream a DVD movie across the network.
[***] IPNE is an addon module to QualNet developed by Scalable Network Technologies.

addressing to virtual IP addresses. This mode of operation is called IPNE-NAT because IPNE performs network address translation between real and virtual IP addresses.

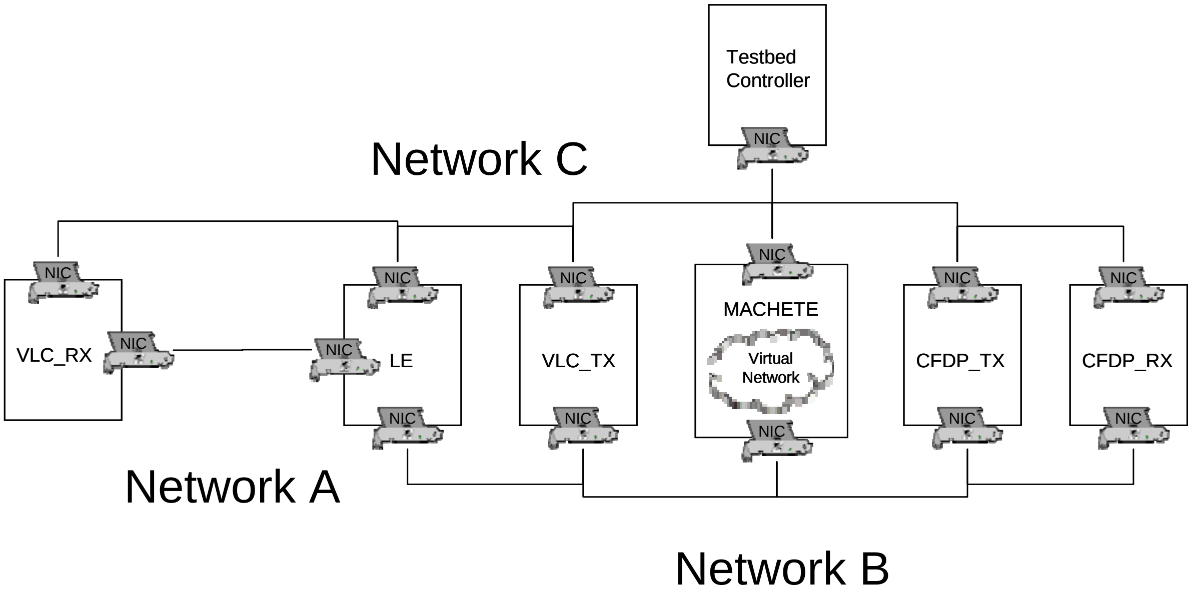

Figure 5. Space Communication Testbed Topology

V. Conclusions and Future Work

MACHETE was successfully integrated and functionally tested with SCT. There is a clearly recognized need for the development of such a capability for NASA[8]. This effort has provides valuable insights on distributed testbed design and implementation. In addition, we have developed sufficient capabilities to support studies for risk analysis, requirements, and other system definitions.

The SCT testbed is constructed from multiple distributed systems connected together by an IP network. While a maximum number of distributed systems was not specified, the IP interface design is capable of supporting many components to form complex testbed topologies. It supports a large number of existing hardware and software COTS products that are IP capable. However, the IP-oriented design trades off precision and control for compatibility and scalability. Synchronization is a known difficult problem in large scale distributed systems. It is a challenge to synchronize data flow with a “global clock” view of simulation time. Additionally, the IP protocol is an unreliable transport mechanism and can introduce errors into the simulation.

For smaller testbeds, TCP/IP offers the precision and control not available by IP. Customized protocols can be developed to synchronize a testbed with time stamped metadata and other time processing mechanisms. Since TCP/IP is a reliable transport mechanism, packet jitter, delays, and other testbed network-related behaviors should have no significant impact on the fidelity of the simulation results where precision is required. Of course, plugging into the testbed is only possible if the component implements the customized protocol. This may not be feasible for devices that do not implement the entire network stack..

Web services provide a convenient mechanism to monitor and control testbed components from a centralized location. Future development includes defining a set of interface functions which allow components to plug-in to testbeds and advertise themselves and their capabilities to the Master Controller.

In the SCT environment, we have performed the initial integration and testing of hardware and software components in a space network simulation testbed. As a continuation of this work, we need to further characterize the performance of MACHETE. MACHETE has performance limitations and we need to identify and understand the causes. One possibility is that the IPNE (packet sniffer software) is unable to sniff packets quick enough from the network resulting in packet drops. We will run MACHETE on a faster computer to see if performance can be improved. Additional work includes validating simulation results with real network systems. We need to determine how well a testbed integrated with MACHETE can model real space network performances and behaviors.

Appendix

Bandwidth (Mbps)	Total Packets Transferred	Packet Loss	Packet Loss (%)	Jitter (ms)	Data Transferred (MB)	Effective Rate (Mbps)
1	5704	0	0.00%	0	7.16	0.95
5	25512	0	0.00%	0	35.8	4.77
10	57022	7	0.01%	0	71.5	9.53
20	102043	19	0.02%	0	143	19.07
30	153064	0	0.00%	0	215	28.67
40	204030	131	0.06%	0	286	38.13
50	255322	0	0.00%	1.167	358	47.73
60	306124	71	0.02%	0.955	429	57.20
70	357145	3	0.00%	1.897	501	66.80
100	343163	0	0.00%	0.716	481	64.13
200	347389	33	0.01%	0.661	487	64.93

Table 1. UDP Transfer Performance without MACHETE

Bandwidth (Mbps)	Total Packets Transferred	Packet Loss	Packet Loss (%)	Jitter (ms)	Data Transferred (MB)	Effective Rate (Mbps)
1	5105	0	0.00%	0.983	7.16	0.95
2	10207	0	0.00%	0	14.3	1.91
3	15308	0	0.00%	0	21.5	2.87
4	20410	0	0.00%	0	28.6	3.81
5	25513	0	0.00%	0	35.8	4.77
6	30615	0	0.00%	0	42.9	5.72
7	35717	1	0.00%	2.102	50.1	6.68
8	40819	4	0.01%	3.103	57.2	7.63
10	51023	3223	6.32%	3.59	66.9	8.92
16	81634	34077	41.74%	2.34	66.7	8.89

Table 2. UDP Transfer Performance with MACHETE

Acknowledgments

The Space Communications Testbed is being developed by a team comprised of ViaSat Laboratories as the prime contractor and Glenn Research Center (GRC), Jet Propulsion Laboratory (JPL), Goddard Space Flight Center (GSFC), and Langley Research Center (LaRC) as subcontractors. A portion of the research described in this paper was carried out at the Jet Propulsion Laboratory under a contract with (NASA).

References

[1] Rieser, J., Berry, K., Clare, L., Slywczak, R., "The NASA Space Communications Testbed (SCT)," *IEEE Aerospace Conference*, Big Sky, MT, March, 2006.

[2] Gao, J., Jennings E., Clare, L. SeGui, J., Kwong, W., "MACHETE: A Tool for Architectural Modeling, Performance Characterization, and Technology Infusion of Space-Based Networks," AIAA International Communications Satellite Systems Conference (ICSSC), Rome, Italy, September 2005.

[3] Netmeeting: http://www.microsoft.com/windows/netmeeting/ (November 2005). NetMeeting is a solution for Internet conferencing for Windows users with multi-point data conferencing, text chat, whiteboard, and file transfer, as well as point-to-point audio and video.

[4] VideoLAN Client: http://www.videolan.org/vlc/ (November 2005). VLC (initially VideoLAN Client) is a highly portable multimedia player for various audio and video formats (MPEG-1, MPEG-2, MPEG-4, DivX, mp3,

ogg, etc.) as well as DVDs, VCDs, and various streaming protocols. It can also be used as a server to stream in unicast or multicast in IPv4 or IPv6 on a high-bandwidth network.

[5] http://dast.nlanr.net/Projects/Iperf/ (November 2005). Iperf is a tool to measure maximum TCP bandwidth, allowing the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss.

[6] Simple Object Access Protocol: http://www.w3.org/TR/soap/ (November 2005)

[7] CCSDS 727.0-B-3. "CCSDS File Delivery Protocol (CFDP)". Blue Book. Issue 3. June 2005.

[8] R. Miller, Space Exploration Communication and Navigation Status and Panel. IEEE Aerospace Conference, Big Sky, Montana, March 2006.