MACHETE: Environment for Space Networking Evaluation

Esther H. Jennings[1], John S. Segui[2], Simon Woo[3],
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

Space Exploration missions requires the design and implementation of space networking that differs from terrestrial networks. In a space networking architecture, interplanetary communication protocols need to be designed, validated and evaluated carefully to support different mission requirements. As actual systems are expensive to build, it is essential to have a low cost method to validate and verify mission/system designs and operations. This can be accomplished through simulation. Simulation can aid design decisions where alternative solutions are being considered, support trade-studies and enable fast study of what-if scenarios. It can be used to identify risks, verify system performance against requirements, and as an initial test environment as one moves towards emulation and actual hardware implementation of the systems. We describe the development of Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation (MACHETE) and its use cases in supporting architecture trade studies, protocol performance and its role in hybrid simulation/emulation. The MACHETE environment contains various tools and interfaces such that users may select the set of tools tailored for the specific simulation end goal. The use cases illustrate tool combinations for simulating space networking in different mission scenarios. This simulation environment is useful in supporting space networking design for planned and future missions as well as evaluating performance of existing networks where non-determinism exist in data traffic and/or link conditions.

Nomenclature

AOS = Advanced Orbiting System
BP = Bundle Protocol
Kbps = Kilo bits per second
CCSDS = Consultative Committee for Space Data Systems
CFDP = CCSDS File Delivery Protocol
DTN = Delay Tolerant Networking
IETF = Internet Engineering Task Force
IND = JPL's Interplanetary Network Directorate
IPN = Inter-Planetary Network
LTP = Licklider Transmission Protocol
MACHETE = JPL's Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation
NI&E = Network Integration and Engineering
SCaN = Space Communications and Navigation
SOAP = this acronym is used for two entirely different objects in two different contexts
(1) Satellite Orbit Analysis Tool;
(2) Simple Object Access Protocol (a web services tool)
STK = Satellite Tool Kit
TOAST = Telecomm Orbital Analysis Tool

1. Introduction

NASA's mission statement clearly spells out the goals "to understand and protect our home planet, to explore the universe and search for life, and to inspire the next generation of explorers…as only NASA can". Space exploration is an important component of the NASA vision. The success of a mission depends on every single component of the system to be functioning correctly and that the integrated system works correctly as a whole under potentially adverse environments. A mission system is indeed complex and each subsystem (component) needs to be designed and tested carefully. In our work, we focus on space-based networking technologies which involve communication

---

[1] Senior Technical Staff, Communications Networks Group, 4800 Oak Grove Drive M/S 238-343, Pasadena, CA 91109

[2] System Engineer, Communications Networks Group, 4800 Oak Grove Drive M/S 238-343, Pasadena, CA 91109

[3] System Engineer, Communications Networks Group, 4800 Oak Grove Drive M/S 238-420, Pasadena, CA 91109

sub-system and communication protocols for Space exploration missions. When considering various alternative network architectures and protocol combinations, it is essential to evaluate the feasibility and performance of these alternatives in a cost-effective manner. While using a mathematical analysis method, we may need to make simplified assumptions of the system to obtain closed-form solutions. As more details on practical constraints of the system become available (to replace simplified assumptions), the system may become too complex to be analyzed mathematically and we would need to use simulation for performance evaluation. For example, fluctuating channel conditions or dynamic traffic profile may be nonlinear and intractable to mathematical analysis which requires simulation.

As early as 2000, the Communications Networking Group at the Jet Propulsion Laboratory (JPL) saw the need to develop a simulation environment for testing and evaluating Space-based network protocols which we named "Multi-mission Advanced Communication Hybrid Environment for Test and Evaluation" (MACHETE). MACHETE is not a single simulation tool but an environment containing various tools that can be selected and combined to perform end-to-end simulation of complex systems. Various tools exist each having its merits in different types of analysis; there is not a single tool that can (or should) suit all analysis. Thus, it is essential to have an environment where the system analyst/engineer has the freedom to choose the appropriate combination of tools to achieve the desired integrated end-to-end system simulation/emulation. MACHETE is tailored to the unique characteristics of space networks, to facilitate mission design and technology development; it comprises various tools in its toolbox, some are custom built and some are commercially available. The tools in MACHETE range from orbital analysis tools, mathematical tools (e.g. Matlab), to discrete event network protocol simulator. MACHETE is poised to analyze a variety of mission operations and to perform architecture studies. As NASA is expanding its Space Communications and Navigation (SCaN) capabilities to support planned and future missions, building infrastructure to maintain services and developing enabling technologies, we see an important role for MACHETE in all of these areas – to analyze and evaluate future SCaN architectures during design phase. To motivate the usefulness of MACHETE, we describe various use cases of MACHETE and lessons in various mission scenarios including near Earth missions, lunar missions, Mars missions and distributed simulation of integrated testbed. In each use case, we also discuss the tool combination used to achieve simulation objectives.

2. MACHETE Architecture

MACHETE is an environment containing various tools and interfaces: (1) orbital and planetary motion kinetics modelling tools, (2) link engineering modelling tools, (3) discrete event network simulation tool, and (4) interfaces among various tools. Examples of orbital analysis tools used by MACHETE are the Satellite Orbit Analysis Program (SOAP) by Aerospace Corporation, the Satellite Tool Kit (STK) by AGI, JPL's Telecommunication Forecaster Predictor (TFP), and JPL's Telecomm/Orbital Analysis Tool (TOAST). Link engineering modeling tools are generally custom Matlab library programs. When we have access to link profile data from actual missions, these recorded link profiles are used as input to the simulation. At the core of MACHETE's network tool is a discrete event simulator QualNet by Scalable Networks, Inc., with extensive wireless and Internet suite libraries. The specific space protocol models developed at JPL are built upon QualNet; these include CCSDS protocols such Proximity-1, Advanced Orbiting System (AOS) data link protocol, CCSDS File Delivery Protocol (CFDP), as well as IETF draft protocols such as Bundle Protocol (BP) and the Licklider Transport Protocol (LTP). We recently added routing algorithms being considered for Delay Tolerant Networks (DTN) such as Contact Graph Routing (CGR) and Delay Tolerant Link State Routing (DTLSR).

Throughout the years, we have collaborated with other NASA centers when cross-center projects called for collaborative tool development and accreditation. From 2007 to 2009, we collaborated with Glenn Research Center (GRC) under SCaN Network project in developing custom models of the SCaN network simulator. GRC provided the CCSDS Encapsulation Service (ENCAPS) and AOS Multiplexing Protocol Data Unit (M-PDU) models while JPL provided TDRSS physical model, traffic generation models and ISS-mission specific link budget libraries.

Interfaces among the various tools and testbed/emulator can be file-based or interactive. QualNet can be interfaced in real-time hybrid emulation-simulation at the transport layer or at the network layer.

3. Mars Mission Scenarios

Mars missions are representative of deep space missions. A Mars network has certain challenging features: asymmetric communication, variable quality links, long delays, and intermittent connectivity. MACHETE has supported several Mars mission studies and analysis. These involved evaluating the performance of CCSDS protocols for Mars missions, determining the latency and buffer storage needed, and comparing the performance of alternative protocols such as CFDP and BP. These are described in further detail below.

3.1 CCSDS Proximity-1 Protocol Performance Evaluation of Lander-Orbiter link

The purpose of the study in [1] was to characterize the performance of CCSDS Proximity-1 protocol when it was used on the rover-orbiter (Mars Exploration Rover to Odyssey orbiter) link in the context of Mars exploration. A rover-orbiter communication link was a complex dynamic system involving orbital geometry and link budget calculation, multi-path fading and Mars atmospheric effects. In addition, different types of data traffic with varying Quality of Service (QoS) requirements were multiplexed on the same link. When reliable science data was multiplexed with expedited operations data, we were interested in the maximum supportable data rate for science data while meeting the QoS requirements of expedited data.

For this study, we selected the following tools from MACHETE: (1) the orbital analysis tool (SOAP) was used to model geometry, to calculate of slant range and connectivity of the rover-orbiter link, and to compute signal strength as a function of time; (2) Matlab link-engineering program was used to model physical channel characteristics, incorporating channel gain fluctuation due to multipath fading, atmospheric scintillation and background noise where link engineering parameters such as modulation and data rates were chosen according to Odyssey orbiter's design document; (3) MACHETE's network simulator (based on QualNet) took the output from Matlab (a stochastic profile of signal-to-noise ratio) as link conditions and simulated data traffic and protocol behaviors.

Two types of data traffic were generated: expedited operational data and reliable science data. Operational data (e.g. engineering, navigation) were expedited with high priority for channel access; the data rate was fixed at 5 kbps. Link margin values were configured at: 5dB, 3.04 dB, 2 dB, 1.03 dB and -1 dB. Reliable science data were sent with automatic repeat request (ARQ); we assume that of science data was generated by the rover and was queued into the transmission buffer prior to the beginning of the pass. Bit error rate (BER) profiles were computed according to various link margin values and various data rates. Metrics of interests were throughput in terms of the number of received data frames and QoS in terms of gaps, errors, latencies and buffer size at the rover. Simulation result showed that the reliable data stream can still be delivered above the BER = 10^{-6} threshold due to ARQ. So, one can relax the link margin to maximize throughput as long as BER is not too large to maintain physical channel synchronization. From the simulation result, we plotted waiting time distribution for expedited data stream of various data rates where the x-axis was the waiting time in second and the y-axis was the waiting time probability. From this chart, we could answer questions such as "what is the supportable expedited data rate while maintaining a latency of less than 600 ms with 99% probability?"

3.2 CCSDS File Delivery Protocol Performance Evaluation to Characterize Latency and Buffer Requirements

In [3], we evaluated latency and storage requirement of CFDP through mathematical analysis and simulation in a Mars scenario involving the Mars Science Laboratory (MSL), Mars Telecommunications Orbiter (MTO), and the Deep Space Network (DSN). Since CFDP provides file-based data management, store-and-forward relay and reliable versus expedited data transfer. Files were segmented into protocol data units (PDUs) to be sent. In deferred NAK mode, the sender will send all the PDUs of a file in sequence in the first round, followed by an end-of-file (EOF) PDU. If all PDUs were received correctly, the receiver would respond with a Finish (FIN); otherwise, the receiver would respond a Negative Acknowledgement (NAK) identifying the sequence numbers of missing or corrupted PDUs. The sender would then resend the identified PDUs and the process would continue in rounds (spurts) until the file was received correctly and completely. We were interested in characterizing CFDP's performance in terms of throughput, latency distribution, the number of rounds needed for reliable file transfer and buffer requirement to support daily data volume. In analyzing network capacity in this scenario, we need to include both communications link bandwidth and storage because space links can be intermittent thus the need for store-and-forward. We were also interested in determining the feasibility of using reliable CFDP over Ka-band for the MTO to DSN link, operating at 85 to 90 percent weather. We were interested in the buffer size needed for reliable data transfer.

The tool selected for this study was MACHETE's network simulator, in addition to mathematical analysis. In this scenario, low priority science data were relayed via reliable CFDP and high priority operational data were relayed via expedited CFDP. Since the dominating distance is between MTO and DSN, we considered distances ranging from 0.54 au at the nearest to 2.44 au at the farthest. Note that the size of the communication pipe is a product of bandwidth and propagation delay. The maximum pipe occurs at a high data rate, but shorter distance; the data rate goes up as inverse distance square, thus dominating the bandwidth-delay product. In our simulation, we used 5 distance(data rate) values ranging from 0.54 au (10 Mbps) to 2.44 au (0.5 Mbps), 4 file sizes from 1 MB to 10 MB, and 4 BERs from 10^{-5} to 10^{-8}; there were 80 combinations total. We used a multi-dimensional Markov chain buffer utilization model and ran Monte Carlo simulation of 50,000 file transfers of 5 different file sizes, 6 different PDU sizes, and 5 different data completeness requirements 95-99.99%.

Simulation result showed that latency was sensitive to BER but not data rate. Using deferred NAK, at BER=10^{-7}, about 1/3 of the files were received completely in one round. Most files were received completely in two rounds. File size had a moderate impact on latency; smaller files (1 MB) took one round to complete at BER=10^{-7} while only 1/3 of the larger files (10 MB) were received completely in one round at the same BER. Analysis was made by estimating the number of rounds needed to complete a file transmission; latency was estimated as a function of the number of rounds and propagation delay. The simulation result on latency confirmed the analytical estimates. We also verified that immediate NAK (per PDU) would not improve latency since it was most beneficial when used with low bandwidth-propagation delay product link and the MTO-DSN link had long delay. For buffer requirement, we observed that increasing the completeness requirement from 95-99.99% required a factor of 3 increase on the buffer space. The buffer size required ranges from 25-30 times the daily data volume where Ka-band pass outage was the main consideration for buffer requirement. An interesting observation was that if we used incremental custody transfer (release buffer when PDUs were received correctly), one may reduce the buffer size requirement to about 6 times the daily data volume.

3.3 Evaluating Alternative Protocol Stacks

While CCSDS CFDP provided reliable file transfer with functions spanning multiple OSI layers, IRTF developed the Licklider Transport Protocol (LTP) to handle reliable file transfers as a function of the transport layer. To compare various combinations of protocol stacks, we implemented the Bundle Protocol (BP, with store-and-forward capability) and LTP (with reliable data transfer function) in the MACHETE network simulator so that the simulation tool can support alternative protocol stacks using reliable CFDP or BP over LTP. In [5, 6], we benchmarked performance of BP in terms of overhead and latency; the conclusion was that BP added minimal overhead with respect to lower layer protocol delays.

Using reliable CFDP store-and-forward had the limitation that it requires full delivery of a file to an orbiter before the file can be transmitted to Earth. It did not allow for parts of a file to be transmitted to one orbiter and parts to another orbiter. In [6], we evaluated the feasibility of using an alternative protocol stack BP over LTP. BP/LTP allowed for a single file to be transmitted along multiple paths. The scenario was a Mars relay network with 2 rovers on Mars, 2 relay orbiters, 3 DSN ground stations and Mission Operation Center. We used MACHETE's network simulator tool to run stand-alone simulations where traffic were generated by the simulator. We also integrated the simulator with JPL's Protocol Technology Lab to validate our simulation by running live traffic. Historical mission data on link availability and data rates were used. We assumed a simple "First Contact" as communications scheduling algorithm, i.e., use the first available contact for communication. Experimental results were correlated with our analysis on average and maximum latency according to the specific schedule. We verified that the experiment results matched our expectation which was an indication that the BP/LTP alternative is feasible for future missions.

In [13], we compared the performance of LTP versus reliable CFDP over a single link. We used MACHETE's network simulator to simulate a Mars-Earth scenario with one link between a spacecraft on Mars and the DSN. The one way transmission time was 860 seconds and file size was 1 MB. The files were sent at a rate such that each file can clear the 1024 kbps link. Various BER values ranging from 10^{-5} to 10^{-8} were used. While LTP's heritage can be traced to CFDP, LTP used a message driven retransmission system whereas CFDP used a combination of message and timers. Using a message driven (check point) ARQ simplified management because time-driven ARQ required knowledge of latency (which increased management overhead). In simulation experiments with a high error rate, LTP had a higher latency compared to CFDP; this was due to the loss of check point messages. The additional latency was the time needed for the sender to notice the loss of a check point message and to resend it again.

4. Near Earth Mission Scenarios

Under the Space Communications and Navigation (SCaN) Program Office and NASA's Constellation Program, we investigated an IP-based network architecture for the Crew Exploration Vehicle Orion to the International Space Station (ISS) mission. Although rendezvous with the ISS via Space Shuttle is not something new, the use of IP-based protocol stacks for such missions is a new concept. President Obama's recent proposal to cancel the Constellation program leads to further needs to re-plan and to re-evaluate future mission concepts and architectures;

MACHETE can be used to investigate these concepts in the new era.

4.1 Performance Evaluation of Space Communications and Navigation Network for ISS Mission

In [9], we simulated an Orion to ISS scenario using IP over generic space link models with appropriate propagation delay. The focus of this initial study was to investigate the feasibility of using an IP-based protocol stack; TDRSS switching was initially modeled as IP-switching due to lack of bent-pipe model in the tool. We subsequently developed the TDRSS bent-pipe model. The tools used were MACHETE's network simulator and JPL's TOAST. The network scenario involved Orion (Crew Exploration Vehicle), Ares (Crew Launch Vehicle), TDRSS, ISS, NASA's Integrated Services Network (NISN), White Sands Ground Terminal, and mission control center (MCC) at Johnson Space Center. We considered S-band link with a return link bandwidth of 192 kbps and a forward link bandwidth of 92 kbps. We used a historic 14-day shuttle trajectory as representing Orion trajectory and evaluated IP-protocol stack performance where the topology of the network underwent dynamic changes. The traffic streams were one 8 kbps constant bit rate of command (high priority) from MCC to CEV, two gamma distribution voice streams (high priority) with peak rate at 19.8 kbps and mean conversation length of 10 minutes, one constant bit rate telemetry stream (medium priority) at 152.6 kbps and one delay-tolerant stream (low priority) at 30 kbps. We ran one simulation where the traffic streams were assigned different priorities and another simulation where all traffic had the same priority. On the return link, adding the peak rates of the two voice streams and telemetry results in 192.2 kbps; which would saturate the return link. However, since there were times when the voice stream was quiet, the bandwidth could be used for other types of traffic. By adding another 30 kbps of low priority delay tolerant traffic, the total peak traffic was 222.2 kbps which was 15.7% in excess of the 192 kbps bandwidth. With prioritization, we could fit another 27 kbps of low priority traffic without loss to any other traffic. Without prioritization, we lost 4% of telemetry and approximately 2% on each of the voice streams on the return link. However, comparing the total bit loss in both cases, the loss was approximately 23 megabits with data prioritization and 59 megabits without data prioritization. Thus, using data prioritization could increase the total throughput.

As we further developed the tool, Glenn Research Center (GRC) contributed additional protocol models. This collaboration produced a library used to simulate Orion to ISS mission, called the SCaN Network Integration and Engineering (NI&E) simulator. In this simulator, we added the TDRSS bent-pipe model, link budget library (based on the Master Link Book [17]), and traffic model for video and audio; GRC contributed the CCSDS ENCAPS and AOS MPDU models. We built an automation framework that contained 28 baseline nominal scenarios described in the Constellation Computing System Architecture Design Document (CSADD) [18]. These scenarios were simulated with prescribed values for QoS and appropriate data volume for each data stream. We used average distances from trajectory analysis for each phase of the mission. Propagation delays were verified to match computed analysis based on average (static) distances among objects. We verified that the data volume prescribed for nominal scenarios were supportable (without loss) with link budget based on the Master Link Book. We also verified expected behavior of QoS (by adding low priority data beyond nominal data volume). Our tool development, its use in distributed test laboratories and simulation results were reported in [11, 12, 15].

In [19], we added further details to our simulation tool for NASA's Constellation Program. Constellation's Data Exchange Message protocol model was added for sending command and telemetry data. We added RTP overhead to the application used for generating audio and video traffic, and extended the link budget library to include Ares links. Instead of using average distance from trajectory analysis, we built an interface to read in trajectory as STK profiles so that the user can either select to use the link budget library or STK for link budget calculation. Using this new setup, we analyzed protocol overhead, throughput, delay and jitter of each traffic type for various phases of the Orion to ISS scenario (Ascent, Low Earth Orbit, Rendezvous, and Return to Earth). We simulated Ares to the Air Force Telemetry Processing Facility (TEL-4) site and Jonathan-Dickinson Missile Tracking Annex (JDMTA) site and verified the behavior of link outage between Ares and TEL-4.

4.2 Performance Evaluation of H.264 Variable Bit Rate Video over IP-based Space Networks

In [16], MACHETE's network simulator was used to evaluate the performance of H.264 variable bit rate video in the Orion to ISS Low Earth Orbit scenario. The purpose was to determine whether the IP-based SCaN network can meet Constellation's requirements on latency and jitter for video traffic.

The experiment setup included Orion video downlink transmission where the Orion to Space Network was Ka-band at 15 Mbps bandwidth. The one-way propagation delay was estimated to be at 276 milliseconds. Code word error rate (CWER) after ½-rate Low Density Parity Check (LDPC) error correction code was 10^{-7}. H.264 video codec's network application layer (NAL) was used to generate data traffic. The protocol stack included NAL, RTP, IP, CCSDS ENCAPS, CCSDS AOS MPDU and link budget library as physical layer. We used two HDTV video sequences. One was the New Mobile Calendar MPEG test sequence (the Vasa ship, moving train with color toys, and background with two types of wallpaper) with a spatial resolution of 1280 by 720, at 50 frames per second; the video was 10 seconds. The second video was taken when the Expedition 13 crew members were performing a series of experiment sin the Destiny laboratory of the ISS. Three H.264 video sequences were generated: (1) expedition-13 sequence compressed to 8.45 Mbps, (2) new mobile calendar sequence compressed to 9.31 Mbps, and (3) new mobile calendar sequence compressed to 8.69 Mbps. Leaky bucket traffic shaper was used with different peak rate constraints. Although the simulation result was not conclusive, we were able to make several observations: (1) the QoS performance that the end user experiences depend on the burstiness of the encoded bit stream and the average data rate for variable bit rate video; (2) the network transport delay for the video downlink was dominated by IP queuing delay and propagation delay; (3) jitter was mainly introduced by IP queuing; (4) using traffic shaping reduced both delay and jitter.

5. Surface Network Scenarios

Under the SCaN and Constellation Programs and the Interplanetary Network Directorate Technology Program, we investigated technologies for surface exploration missions.

5.1 Lunar Mission Scenario

In [14], we considered the suitability of IEEE 802.11, IEEE 802.16 or a combination of both of these protocols to be used for lunar surface exploration. The tool used was MACHETE's network simulator. The lunar surface scenario involved two teams of collaborating astronauts, one base station and one rover. The base station and the rover had the capability to act as relays. The surface area was bounded by 500 by 650 meters containing all astronauts and landed assets.

There were 3 experiment setups: (1) using IEEE 802.11g only with and without retransmission; (2) using IEEE 802.16e only; (3) using a combination of IEEE 802.11g and IEEE 802.16e. In the combined protocol scenario, the long haul link (base station and rover) used IEEE 802.16, other short-range networks used IEEE 802.11. There were two short-range networks, one connecting two astronauts with the base station, and the other connecting the other two astronauts with the rover.

Different traffic types of various volume and distributions among different communicating pairs were multiplexed, including voice, command, telemetry, and caution-and-warning. Voice traffic was from astronaut to astronaut; the QoS for voice required low jitter and low loss. Caution-and-warning was sent among the following pairs: rover to base station, base station to astronaut and astronaut to base station. Telemetry was sent among the following pairs: rover to astronaut, astronaut to base station, astronaut to astronaut. Command was sent from base station to astronaut, base station to rover, and astronaut to rover. File transfer occurred from base station to rover and from base station to astronaut.

Our preliminary result showed that 802.11g (with ARQ) satisfies voice QoS; 802.16 had higher delay and jitter due to all data must go through the base station. For caution-and-warning, traffic on the uplink from subscriber (astronaut, rover) to base station suffered longer delay (on 802.16 network) and higher packet loss (on 802.11 network) as compared to the downlink traffic (base station to astronaut or rover). Even with channel separation, 802.11 outperformed the 802.11/802.16 combined network in terms of delay and jitter. Disabling link layer retransmissions (ARQ) on 802.11g further improved delay. The preliminary study was made under very short period of time and limited funding; the result was not conclusive but did show interesting trades between the protocols considered.

5.2 Multi-Modal Sensing and Tracking

In a matching task to the Hybrid Simulation Environment for Space-based Networks task under Interplanetary Network Directorate Technology Program, we extended MACHETE's network simulation tool to include both sensing and communication models in a discrete-event simulation environment. Using this tool, we simulated vehicle detection (sensing) and information dissemination (information fusion) via wireless communications under different parameters and scenarios [2, 8]. Although the scenarios described in [2,8] were in a military setting, the same tool can be used on planetary surface where the vehicles can be exploration rovers.

In [2] we evaluated spatial lay down of sensors and experimented with a mixture of sensors of different modalities for vehicle detection and tracking. Since vehicle objects were of different types possessing different attributes, sensors of different modalities were used. We assume that sensor system did not affect the behavior of the sensed object.

For tracking a moving vehicle, we simulated a field of simple omni-directional sensors where the vehicle moved in pre-determined trajectories. The sensors were spatially uniformly distributed in a 1000 by 1000 meter region; the number of sensors was less than 50 and the sensing range was 30 meters. When a sensor detected the vehicle, it reports time and position of the vehicle. A track is then generated by combining the sensor reports. Tracking accuracies were quantified by computing the tracking error which was the average distance between the actual track and the sensor detection estimated track. As expected, we observed decreasing tracking error when more sensors were used.

Multi-modal target detection simulation assumed two types of vehicles with different attributes. In this scenario, we fixed the total number of sensors used and vary the number of sensors of each modality. For different mixtures of sensor modalities, 10,000 tracks were generated. Experimentation showed that the overall detection probability is 98% or better when there was a more balanced mixture of sensors. The main purpose of this work was not to answer a specific question but to provide a tool, when given enough details on the scenario parameters and assumptions, that can determine potential optimal solutions with respect to certain metric of interest. By varying the mixture of sensor modality, we can also answer questions about the duration of n-exposure of the vehicle, where n is an integer. N-exposure means the vehicle is being detected by n sensors for a specific duration.

5.3 Evaluation of Sensor Network Data Fusion Schemes

In [8], we focused on alternative sensor network architecture for information fusion. The performance metrics were vehicle detection and false alarm rate. We extended the tool to include terrain embedding; line-of-slight computations were implemented in the Physical layer.

Using the extended tool, user could define mission environment including terrain features that may affect both communications signal and sensor signal propagation. Directional, omni-directional sensing and multi-hop communications were modeled. We investigated 3 different sensor architectures for information fusion. Sensors were organized into disjoint clusters where each cluster had a cluster-leader. The sensors within clusters had a smaller communication range compared to cluster-leaders. The fusion methods were: (1) localized fusion, (2) hierarchical fusion and (3) distributed fusion. In localized fusion, micro-sensors only communicated with other sensors in its own cluster. In hierarchical fusion, sensors communicated with their respective cluster-leaders and cluster-leaders may communicate with other cluster-leaders. Distributed fusion allowed sensors and cluster-leaders communicate with sensors in other clusters either directly or multi-hopped. Modified majority voting with threshold and window size was used for information aggregation. We examined tradeoffs of different fusion architectures under various sensing sliding window size and false event occurrence rate.

Simulation result showed that localized fusion had limitation on vehicle detection because the vehicle was detected only when its trajectory passed through the local cluster covered area. Hierarchical fusion enabled the vehicle detection information to be disseminated to other clusters, but it took time to propagate information to other clusters through cluster-leaders. Distributed fusion allowed faster vehicle detection (lifting the constraint of information dissemination through cluster-leaders), but it increased communication load and could consume more energy. As the window size was increased, we also observed increased false alarm rate. As expected with information dissemination, distributed fusion was more susceptible to false alarm, hierarchical fusion had less false alarms and localized fusion had the least amount of false alarm.

6. Distributed Simulation – Integration with Testbed

We described the use of MACHETE's network simulator in stand-alone network simulations in previous sections. In addition, MACHETE's network simulator had also been used to support integrated simulation of simulators and emulators. In the earliest work [4], we integrated the simulator with emulators in the Protocol Test Lab (PTL) at JPL where the simulator simulated long delays on the link and data loss on less than perfect channels. The network simulator allowed an application/protocol to send real data traffic over the simulated link. Another use of the simulator was when the actual implementation of a lower layer protocol was not available in the emulator; we used the simulator to simulate lower layer protocol behaviors.

6.1 Integrated Testbed for Lunar Mission

In [7], we integrated MACHETE's network simulator with ViaSat's emulation testbed through the simulator's IP Network Emulator (IPNE) and web-based Simple Object Access Protocol (SOAP) to simulate lunar scenarios involving Earth ground stations, Lunar Relay satellites, Crew Exploration Vehicle (CEV) and Lunar surface assets. MACHETE's network simulator simulated the Lunar surface assets and links between Lunar Relay satellites and between satellites to CEV. The Exploration Systems Research and Technology Program funded the initial phase of the Space Communications Testbed (SCT) project (April 2005) where ViaSat Laboratories was the lead; other team members were Jet Propulsion Laboratory, Glenn Research Center, Goddard Space Flight Center and Langley Research Center (LaRC). The distributed simulation used web services to testbed components. The testbed was setup 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 and Lunar Proximity Network (LPN). The component simulated by MACHETE was the LPN element. The LPN involved 4 astronauts, two landers, one rover, a base station. IEEE 802.11 was used between communicating surface elements. CCSDS Proximity-1 was used between surface element and LRS, generic space link protocol was used between LRS and DSN. External traffic streams were injected from hardware at the astronauts and at CEV.

In a test laboratory environment with integrated simulators and emulators, it was important to benchmark testbed equipment performance limitations. The hardware performance depended on the central processor unit, network interface cards and random access memory speeds. Software performance depended on operating systems and other software libraries such as *libpcap* and MACHETE's simulator libraries. It was important to identify and measure performance overhead that may unintentionally influence simulation results in terms of latency, packet loss and throughput degradation. We measured end-to-end effective bandwidth with *IPerf* and observed throughput degradation at 10 Mbps with the network simulator in the middle; without the network simulator, throughput degradation occurred at 70 Mbps. Thus, the use of a network simulator placed a constraint on the supportable data volume. The major overhead was due to the filtering and processing of each IP-packet between simulator and emulator. Another limitation was that IPNE did not synchronize IP packet flow between emulator and simulation time, thus the experiments were not repeatable to reproduce the same results. We considered interfacing the simulator to an emulator either at the Transport layer or at the Network layer. Interfacing at the Transport layer required finer granularity and provided reproducible results, but it required writing custom interface software. Interfacing at the Network layer worked with off-the-shelf Internet applications, but did not provide reproducible results. MACHETE's network simulator was successfully integrated and functionally tested with SCT. Although we did not conduct any extensive analysis on the simulation results from SCT (due to funding constraints), we had proven the concept of integrating distributed simulator and emulators and gained valuable insights on distributed testbed design and implementation.

6.2 Integrated Testbed for ISS Mission

In [10, 11, 12], we participated in various distributed simulation of NASA's Orion spacecraft and Ares launch vehicle in a mission to the International Space Station. We worked with both Integrated Mission Simulation (IMSim) project and Distributed Systems Integration Laboratory (DSIL). IMSim's project goal was to research, develop and deploy technologies, processes and simulation which support the collaborative, interoperable and distributed simulation of complex space systems in support of NASA's Exploration Initiative. DSIL's focus was on early integration and interoperability; checking that the different components can exchange data successfully (format issues etc). In these distributed environments, truth data pertaining to spacecraft status, position, velocities were exchanged via Transmission Control Protocol (TCP); mission data such as telemetry and command were exchanged via User Datagram Protocol (UDP).

The IMSim/DSIL architecture contained the following simulated components: Orion crew exploration vehicle (CEV), Ares I crew launch vehicle (CLV), Space Communications and Navigation (SCaN) Network, Ground Systems, International Space Station (ISS), and mission control center (MCC), where these are interconnected through the IEEE 1516 High Level Architecture (HLA) using TCP. IMSim/DSIL consisted of multiple testlabs that were geographically located at different parts of the United States. These testlabs were connected through NISN. The NISN backbone and the externally accessible IP addresses through the facility network formed the NASA Distributed Simulation network (DSNet) which was part of NISN. The testlabs that were connected through DSNet were located at Jet Propulsion Laboratory (JPL), Marshall Space Flight Center (MSFC), Johnson Space Center (JSC), and Kennedy Space Center (KSC).

Distributed simulation was run in an HLA framework where the HLA Run-Time Infrastructure (RTI) defined how each simulated components interact with each other. In HLA terms, each simulated components was a federate and the Federation Object Model (FOM) specified the relationships among data that were exchanged in a simulation execution. In the IMSim and DSIL distributed simulations, the federates were: Crew Exploration Vehicle federate and Mission System federate (at Johnson Space Center), Mobile Launcher federate and Launch Control System federate (at Kennedy Space Center), Crew Launch Vehicle (at Marshall Space Flight Center), TDRSS federate and Space Communications and Navigation Network federate (at Jet Propulsion Laboratory). The role of the SCaN federate was to subscribe to truth data and use spacecraft position to compute propagation delay. Representative telemetry data was sent from the CEV federate to MS through SCaN federate where SCaN federate simulated latency through TDRSS bent-pipe with respect to spacecraft trajectories. Through our multi-center tests, we observed the bandwidth usage of various data, and gained a better understanding of the time synchronization dependencies and limitations among distributed simulations. We verified that SCaN correctly relayed the data between federates and verified the measured latency against the expected latency from analysis. The benchmark showed that HLA did not incur significant overhead and the mission representative telemetry data did not dominate DSNet bandwidth. The majority bandwidth used was for video teleconferencing among centers during the tests for simulation coordination.

7. Conclusion and Future Work

Most of the commercial tools available to date specializes in simulation of terrestrial networks or near Earth (satellite) networks; thus the need to develop a unique simulation environment for space networking supporting NASA mission systems. In this paper, we described the development and use cases of the MACHETE in supporting various mission studies and analysis. By looking at the missions planned from now until 2025, there will be increased number of spacecraft and communications capability involving optical communications. Mission systems are complex due to the various constraints on trajectories, antenna pointing, power, processing and storage capabilities etc. The different assets and resources need to be orchestrated carefully to meet mission objectives while optimizing resource utilization. It is important to have a simulation environment with flexible tool selection and interfaces so that users can choose the tool combinations that best suit his experiment goal. MACHETE was built with the intention to provide a flexible simulation and testing environment, and we had been extending its capabilities continually.

In this paper, we illustrated the use of MACHETE in various simulation scenarios, evaluating protocol performance when multiple types of data with different quality of service requirements were multiplexed. We used the tools to characterize latency and buffer requirements for specific protocols, and compared performance of alternative protocol stacks. The test environment was also used to evaluate performance of custom space networking protocols (e.g. CCSDS protocols) in specific mission scenarios. A novel customization to the network simulation tool was the combined sensing and communications models used to detect and track surface vehicles in exploration missions, and comparison of different data fusion schemes.

The usefulness of the test environment is not limited to stand-alone simulations. It can be interfaced to assets in test laboratories in a hybrid simulation / emulation setting. This has been demonstrated in the various distributed simulation experiments supporting IMSim and DSIL.

From the various use cases in the past, and the flexibility of the environment leveraging on the strength and specialization of various tools, we envision that MACHETE will be a useful test and evaluation environment for future missions. Specifically for NASA's SCaN network architecture design and performance evaluation of network protocols and services.

Acknowledgement

The research described in the paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with National Aeronautics and Space Administration. We also acknowledge the contribution from NASA GRC to the SCaN NI&E simulator version 2 and that GRC organized the process for the accreditation of the simulator. We acknowledge the collaboration with other NASA centers on the IMSim and DSIL experiments.

References

[1] Jay Gao, “Simulation tools for Protocol Design and Evaluation”, JPL’s IND Technology & Science News, May 2003, Issue 17.

[2] Loren Clare, Esther Jennings, and Jay Gao, “Performance evaluation modeling of networked sensors”, Proceedings of the 2003 IEEE Aerospace Conference, Big Sky, Montana, March 8-15, 2003.

[3] Jay Gao and John Segui, “Performance Evaluation of CCSDS File Delivery Protocol – Latency and Storage Requirement”, IEEE Aerospace Conference, Big Sky, Montana, March 2005.

[4] Jay Gao, Esther Jennings, Loren Clare, John Segui, Winston Kwong, “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.

[5] John Segui, Esther Jennings, “Delay Tolerant Network Protocol Simulation in MACHETE”, Second International Conference on Space Mission Challenges for Information Technology (SMC-IT), Pasadena, California, July 2006.

[6] Christopher Krupiarz, Esther Jennings, John Segui, Jackson Pang, Joshua Schoolcraft, Leigh Toergerson, “Spacecraft Data and Relay Management Using Delay Tolerant Networking”, AIAA International Conference on Space Operations (SpaceOps), Rome, Italy, June 2006.

[7] Winston Kwong, Esther Jennings, Loren Clare, Dee Leang, “Technology Developments Integrating a Space Network Communications Testbed”, AIAA International Conference on Space Operations (SpaceOps), Rome, Italy, June 2006.

[8] Simon Woo, Esther Jennings and Loren Clare, “A Simulation Tool for ASCTA Microsensor Network Architecture”, IEEE Aerospace Conference, Big Sky, Montana, March 2008.

[9] Esther Jennings, David Heckman, “Performance Characterization of Space Communications and Navigation (SCaN) Network by Simulation”, IEEE Aerospace Conference, Big Sky, Montana, March 2008.

[10] Esther Jennings, Michael Blum, Juan Busto, Victoria Chung, Edwin Crues, Dan Dexter, David Hasan, Joe Hawkins, “Integrating Collaborative Distributed Simulations for Space Exploration Missions”, AIAA Modeling and Simulation Conference, Honolulu, Hawaii, August 2008.

[11] Polly Estabrook, Michael Cheng, Esther Jennings, Laura Hood, William Othon and David Hasan, “Real-time, Integrated, Testing of Complex Space Architectures via Constellation’s Distributed Systems Integration Laboratory (DSIL)”, 59th International Astronautical Congress, Glasgow, Scotland, September 2008.

[12] Esther Jennings, Sam Nguyen, Shin-Ywan Wang, and Simon Woo, “Interfacing Space Communications and Navigation Network Simulation with Distributed System Integration Laboratories (DSIL)”, AIAA International Conference on Space Operations (SpaceOps), Heidelberg, Germany, May 2008.

[13] John Segui and Esther Jennings, “Simulating Autonomous Telecommunication Networks for Space Exploration”, AIAA International Conference on Space Operations (SpaceOps), Heidelberg, Germany, May 2008.

[14] John Segui, Esther Jennings, Hemali Vyas, “Evaluation of IEEE 802.11g and 802.16 for Lunar Surface Exploration Missions Using MACHETE Simulations”, AIAA Modeling and Simulation Conference, Chicago, Illinois, August 2009.

[15] Esther Jennings, Richard Borgen, Sam Nguyen, John Segui, Tudor Stonescu, Shin-Ywan Wang, Simon Woo, Brian Barritt, Christine Chevalier and Wesley Eddy, “Space Communications and Navigation (SCaN) Network Simulation Tool Development and its Use Cases”, AIAA Modeling and Simulation Conference, Chicago, Illinois, August 2009.

[16] Hua Xie, John Segui and Esther Jennings, “Trace Driven Simulation of H.264 VBR video transmission using the SCaN Simulator”, JPL Interoffice Memorandum, July 2009.

[17] CxP 70022-03 “Command, control, communication, and information (C3I) interoperability standards book, volume 3: master link book”, C3I20Vol3, revision A.

[18] CxP 70078, “Constellation Program Computing Systems Architecture Description Document (CSADD)”, Rev. A, 21 March 2008.

[19] Hemali Vyas, Erica Lieb et. Al, “Communications and Network Modeling Activity – SAVIO-03-1-13, SAVIO-03-1014”, Constellation SAVIO Internal report (white paper), January 2010.