Traffic Modeling for Deep Space Network in the Human Mars Exploration Era[1]

Kar-Ming Cheung[2], Douglas Abraham[3], Marc Sanchez-Net[4], Kristy Tran[5], and Carlyn-Ann Lee[6]
Jet Propulsion Laboratory, California Institute of Technology, Pasadean, CA 91109

In this article we describe the analysis and simulation effort of the end-to-end traffic flow for the Deep Space Network (DSN) in the Human Exploration Era, when DSN will provide communication and navigation services for human missions to distant celestial objects like the Moon, asteroids, and Mars. Using the network traffic derived for the 30-day period within July/August 2039 from the Space Communications Mission Model (SCMM), we simulate the bandwidths of the ground links and the buffer profiles of the network nodes. We also use a 2-state Markov scheme that models the store-and-forward mechanism that regulates the ground network traffic. The network traffic modeling and simulation generates ground bandwidth and buffer statistics, which in turn are used to formulate the future DSN ground network bandwidth and storage requirements.

I. Introduction

THE Deep Space Network (DSN) consists of 3 sites with 13 operational antennas. Five additional 34-m beam-wave-guide (BWG) antennas are planned between now and 2025 to address the growing communications and tracking needs for current and future deep space missions. The current DSN architecture and evolution plan is depicted in Figure 1[7].

Deep space missions are traditionally robotic missions. Except for the spacecraft uplink command and health and safety telemetry, the bulk of the deep space robotic mission data consists of data types that can tolerate reasonably high latency[8]. This allows the terrestial network to implement a data-buffering scheme that "smooths" the instantaneous bandwidth of the ground links, thus reducing the bandwidth requirements and thus the cost of the terrestial network.

The network tracking data – Doppler, ranging, and Delta Differential One-Way Ranging (DDOR) data – are deep space signal measurements at the DSN antenna for generating navigation solutions for spacecraft, and are typically mission time-critical.

Circa 2035, in addition to deep space robotic missions, it is envisioned that the DSN will provide communication and navigation services for human exploration missions to distant celestial objects like the Moon, asteroids, and Mars. Data delivery, and data latency requirements of human missions can be very different compared to the robotic deep space missions. This in turn drives the bandwidth requirements of the next-generation DSN terrestrial network to meet future mission needs.

In this paper, we describe a top-down and latency requirement-driven analysis and simulation approach to size the bandwidth and storage requirements of a store-and-forward[9] terrestrial network for the DSN, mission traffic scenario, and set of data types with different latency requirements. The focus of this paper is on downlink traffic only, as this is the key driver of the ground bandwidth and storage requirements.

---

[2] Coresponding author, email: Kar-Ming.Cheung@jpl.nasa.gov, Phone: 1-818-393-0662, Group Supervisor and Principal Engineer, Telecommunications Architectures Group, MS 238-420.
[3] System Engineer, Archtiecture, Strategic Planning & System Engineering Office, MS 301-450
[4] Ph.D Candidates, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.
[5] Academic Part-Time, Telecommunications Architectures Group, MS 238-420.
[6] Engineering Applications Software Engineer, Telecommunications Architectures Group, MS 238-420.

[7] The DSN antenna upgrade/retirement schedule is dependent on the planned budget.
[8] In this paper, latency refers to the time taken for a packet received by a DSN antenna to transmit across the ground network to its destined mission operation center.
[9] It is expected that advanced network protocols like Delay Tolerant Network (DTN) will provide efficient store-and-forward mechanisms to regulate the end-to-end data flow between the flight, ground, launch, and mission operation systems during the human Mars Era [12].

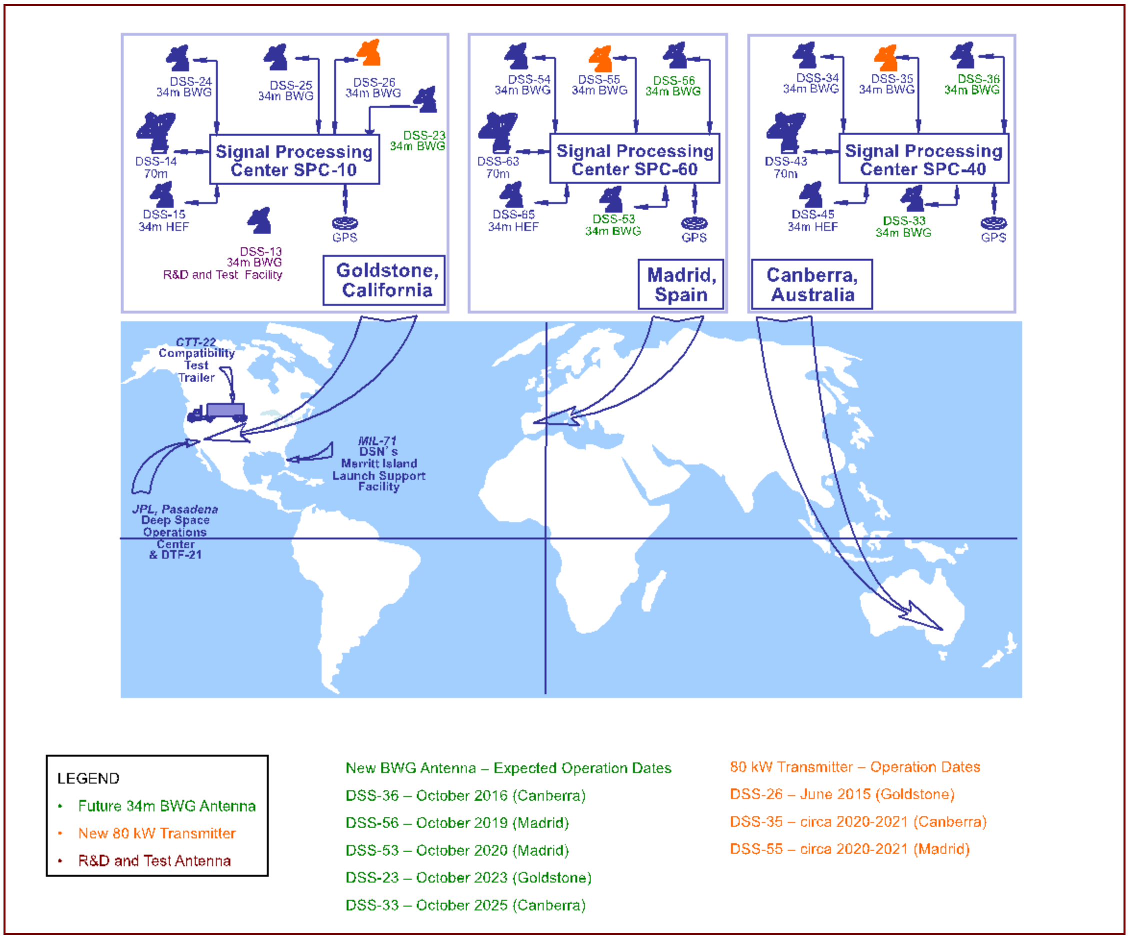

Figure 1. DSN Architecture and Evolution Plan as of October 26, 2015

The uniqueness of this approach lies in the modeling of the store-and-forward mechanism of each network node using a 2-state Markov scheme as discussed in [1] and [2]. The term store-and-forward refers to the data traffic regulation technique by which data are sent to a network node, temporarily stored, and later sent to the destination node or to another intermediate node. For the DSN terrestrial network, the store-and-forward mechanism is used to regulate the network data flow and terrestrial link resource utilization such that the user data types can be delivered to their destination nodes without violating their respective latency requirements. The Communication Service Office (CSO)[10] that provides wide area network communication services within the National Aeronautics and Space Administration (NASA) is transitioning the DSN terrestrial network from dedicated landlines to commercial cloud-based Internet services. After the upgrade, the bandwidth will be increased to 40 Mbps from each DSN site to JPL Central[11]. In this paper, we assume that the routing delay is negligible.

The general methodology involves the following steps:

1. In light of the new DSN cloud-based terrestrial network infrastructure, we consider the functional flow (instead of physical flow) of data from the DSN sites to JPL Central, and from JPL Central to the destination mission operations centers (MOCs). This translates into a star-like terrestrial network topology.
2. We generate the deep space mission traffic for the 30-day period within July/August 2039 derived from the Space Communications Mission Model (SCMM). We also specify the different data types and their corresponding data generation statistics, and end-to-end latency requirements.

[10] Formerly known as NASA Integrated Services Network (NISN).
[11] JPL Central refers to the data processing center at the Jet Propulsion Laboratory.

3. We exercise the SCMM-derived mission traffic model to simulate mission data that flow into the network regulated via the store-and-forward techniques. We then estimate the required bandwidth of each individual network segment, and the required storage of each network node that would meet the mission data latency requirements within the course of the user traffic simulation.

The rest of the paper is organized as follows: Section 2 describes the modeling of the SCMM-derived mission traffic in detail. Section 3 discusses the DSN terrestrial network modeling tool and the modeling of mission traffic data types. Section 4 provides the terrestial network bandwidth and buffer statistics, and discusses the corresponding derived requirements. Section 5 provides the concluding remarks and future work.

II. Modeling of the Mission Traffic

NASA's Space Communications and Navigation (SCaN) Office maintains a Space Communications Mission Model (SCMM) that enumerates current and anticipated future missions potentially requiring communications support from any of SCaN's three communications networks: the Space Network, the Near-Earth Network, and the Deep Space Network. In this study, we have focused on the missions potentially supported by the Deep Space Network (DSN) in the 2039 timeframe. Since many of the missions in this timeframe either lack substantive definition or are merely placeholders for competitively-bid mission lines, the latest NASA roadmaps and National Research Council Decadal Surveys were used to infer representative candidate missions. As part of a larger effort to understand future deep space mission trends and implications, a multi-step process was then initiated to translate this timeline of candidate missions into a set of communications-related requirements as a function of time that, for this study, could be used to generate an aggregate simulated tracking schedule for the timeframe of interest.

The first step in this process was to locate and "mine" concept design studies for each of the candidate missions. The "mining" involved digging out the appropriate communications and mission design data needed to ultimately develop and specify ground antenna uplink and downlink capability, frequency band, tracking pass frequency, and pass duration requirements as a function of time. These data were then stored in our "Mission Set Analysis Tool (MSAT)" database.

Since many of the concept designs were based on trajectories utilizing dates not quite in sync with the latest roadmaps, a second step involved updating those trajectories using an "Orbital Trajectory Inference Engine (OTIE)" for the Keplerian trajectories and, for the non-Keplerian ones, using specialized trajectory support from JPL's Mission Design & Navigation Section.

This updated trajectory information was then used within MSAT to associate each mission's specific operational segments with the specific dates at which they must begin and end – an association that then allowed each mission's tracking requirements to be properly specified as a function of time. MSAT also embodies a SCaN-standard link budget tool that, in conjunction with the trajectory-derived range information from OTIE, was used to derive each spacecraft's ground station G/T and EIRP requirements as a function of time.

The next step involved using OTIE to process the trajectory information in a manner that allowed specification of the view periods for each mission's spacecraft relative to each of the DSN's ground antennas as a function of time. Such information is vital since the antennas can only track the spacecraft when they can see them.

With each mission's spacecraft communications and tracking requirements, ground station G/T and EIRP requirements, and view periods available as a function of time, we were then in a position to execute the final step for arriving at a simulated tracking schedule: simulating how all of the associated spacecraft load up on the DSN's antennas as a function of time. To do this, the MSAT frequency band, tracking, G/T, and EIRP requirements and OTIE's view periods, for each mission's operational segments, were loaded into a DSN Architecture Loading Analysis Tool (ALAT). This tool employs an algorithm to apportion each mission's tracking requirements, for each operational segment, across the available antennas according to the frequency band, G/T, EIRP, and view period constraints – relative to the tracking needs of all the other missions. The net result, available as one of several possible outputs from ALAT, was a simulated schedule for each mission's tracking on the DSN, by antenna, as a function of time. The interplay between all of the different tools to arrive at this product is summarized in Figure 2.

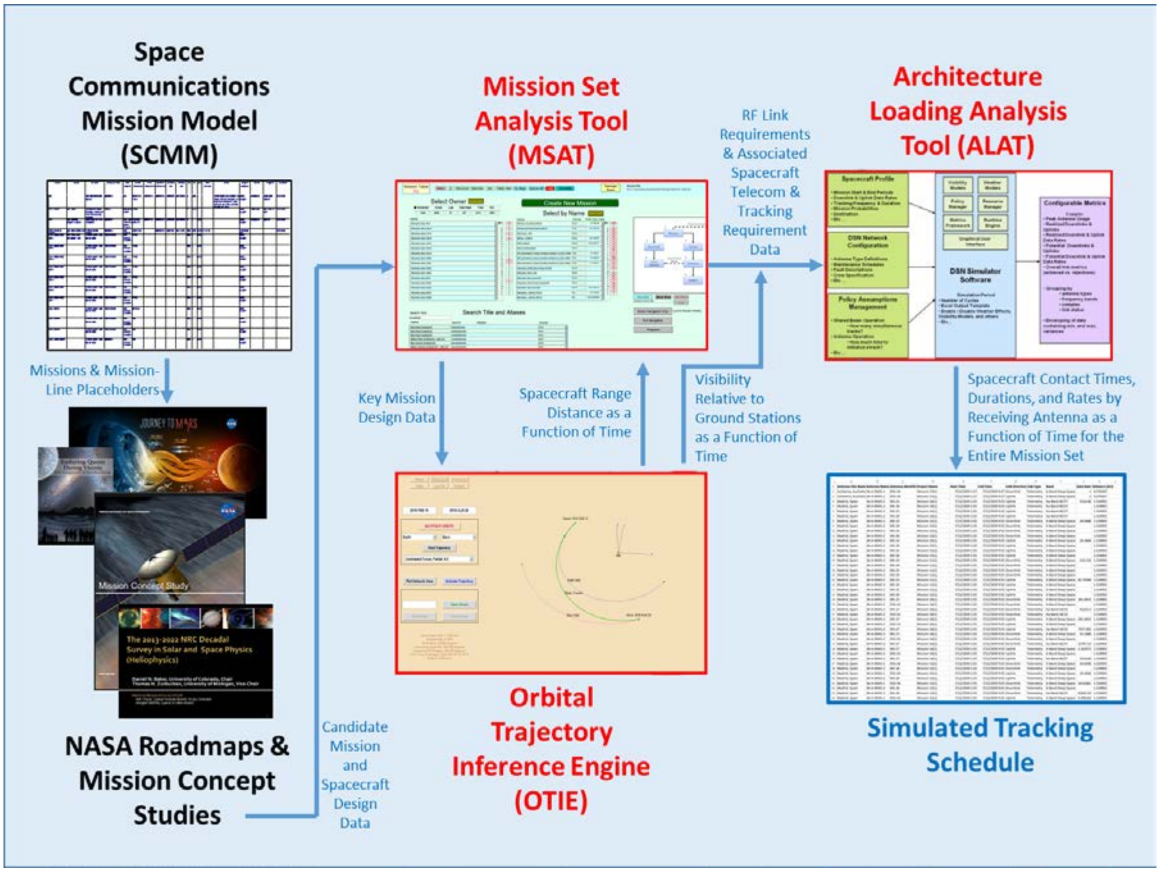

Figure 2. Process for Generating the Simulated Tracking Scheduled

The Simulated Tracking Schedule is an EXCEL output from ALAT that, as a function of time, enumerates each DSN antenna, corresponding antenna site, the name of the mission being tracked on the antenna, the mission ops segment during which the track is occurring, the start and end times of the contact, the link directions (i.e., downlink or uplink), the type of link (telemetry, command, etc.), the frequency band being used, the data rate, and the range distance. This information can then be used to derive the specific type of data being downlinked (e.g. engineering telemetry, video, science, etc.), the aggregate daily data volumes, and the instantaneous information bit rates into each antenna. By appending each track's information with the associated Mission Operations Center (MOC), it also becomes possible to estimate the volume of a particular type of data that needs to be transmitted to a particular MOC via the postulated terrestrial network topology[12].

The Simulated Tracking Schedule also contains navigation tracks in which no communication data are actually downlinked from the spacecraft, just one-way ranging tones. Known as delta Differential One-way Ranging (delta-DOR), this technique involves using two antennas located at different station sites to listen for tones from the spacecraft such that the receive times can be compared to yield a time delay. This time delay multiplied by the

[12] Because this study is looking at a 30-day period in 2039, the specific MOCs that will be used for many of the missions cannot be known a priori. Many of the missions (e.g., Discovery, New Frontiers, etc.) are competitively-bid. So, there is no way to know who will be awarded the mission and, hence, no way to know who will be operating it. Even in the human Mars exploration realm, there is no way to know what the international partner contributions might be and how that might influence associated MOC participation. So, a simplified set of MOCs is assumed, with each MOC assigned to one or more missions, based on the DSN's historical users and associated mission types. The MOCs included ESA, GSFC, JHU, JPL, and JSC. While other international and U.S. MOCs are anticipated in the future, this set is considered adequate for modeling the terrestrial network and observing the influence of its store-and-forward mechanisms on data types of different latencies.

speed of light yields a path-length difference between the two stations that can then be used to compute the angular location of the spacecraft. For calibration purposes, the DOR tone observations are compared with the stations' repeated observations of a quasar in the same portion of the sky as the spacecraft. Since the quasar's angular position is already well known, differencing these observations with the DOR observations cancels out various sources of signal delay, allowing an improved accuracy for the angular measurement of spacecraft position. All of the observations involve data-volume-intensive open-loop recordings on the ground that must be compared to one another. So, while no communication data are being downlinked from the spacecraft, these navigation tracks produce large quantities of data on the ground that have to be transmitted through the postulated terrestrial network topology.

For this study, all of the above track-related information was generated for a 30-day time period between July 12, 2039 and August 12, 2039 – a high-activity time period in which one postulated human mission is preparing to leave Mars and another is in route to Mars. The DSN architecture in this timeframe was assumed to contain all of the antennas planned for the period following the DSN Aperture Enhancement Project, as well as seven additional 34m antennas per complex[13] to enable sufficient antenna arraying to meet the high data rate demands of the human Mars missions when at maximum range. The next section will go into more detail regarding the modeling of the mission traffic data types and the DSN's terrestial network.

III. Interplanetary Network Modeling Tools and Mission Data Definitions

This section presents ArchNet, the coarse-grain discrete-event network simulator developed for estimating the required network capacity as a function of the amount of data transmitted through the system and the latency constraints of this data (see Figure 3). This presentation is structured as follows: First, a generic introduction on how to model a space ground system communication network is presented. Then, ArchNet's software architecture is discussed in detail along with the different software modules and components used to implement it. This is followed by a description on how traffic can be specified in the simulation environment. Finally, a brief description on how to interface with the tool and extract useful results is provided.

A. Modeling the Ground Segment of Space Communication Networks

The Networks that provide communication and navigation services to space exploration missions are typically composed of multiple geographically dispersed ground stations interconnected through a ground infrastructure [1]. Notionally, the system resembles a wide-area network where nodes can either be ground sites (with one or multiple antennas connected to them), network operation centers, mission operation centers and data processing centers (e.g. NASA's Distributed Active Archive Centers [3])

Next we outline the set of underlying assumptions that were used while developing ArchNet. Based on the findings from reference [1], we assume that all signal processing functionality (e.g. waveform sampling and digitalization, framing or decoding) are performed at the DSN sites and are transparent to the simulation. On the other hand, all nodes in the network (including the DSN sites) perform two canonical sets of high-level networking functionality: routing and leveling. Routing refers to the selection of the best path towards destination and is implemented by minimizing the hop count between a given node and the data's terminal user. In that sense, data is always sent first to JPL[14] and then, if necessary, forwarded to the appropriate MOC.

Alternatively, leveling refers to the ability of certain nodes to store data and route it at a later time (i.e. it is analogous to store and forwarding) while ensuring that spacecraft and scientific data latency requirements are not violated. Multiple approaches to modeling store-and-forward and flow control on networks have been proposed in the literature (e.g. [4], [5]). Of them, ArchNet adapts and efficiently implements the two-state Markov leveling scheme presented in reference [2], as it was originally conceived and validated for networks that support space-exploration applications such as the ones considered in this paper. Finally, all results generated with ArchNet and presented in this paper are only concerned with the terrestrial return lines from the three DSN sites to the mission operations centers.

[13] This is postulated based on a data return requirement of 250 Mbps from Areostationary relays operating at 37-37.5 GHz in the Human Mars Era.

[14] To perform data conditioning, and data merging from multiple sources, e.g. during station-handover.

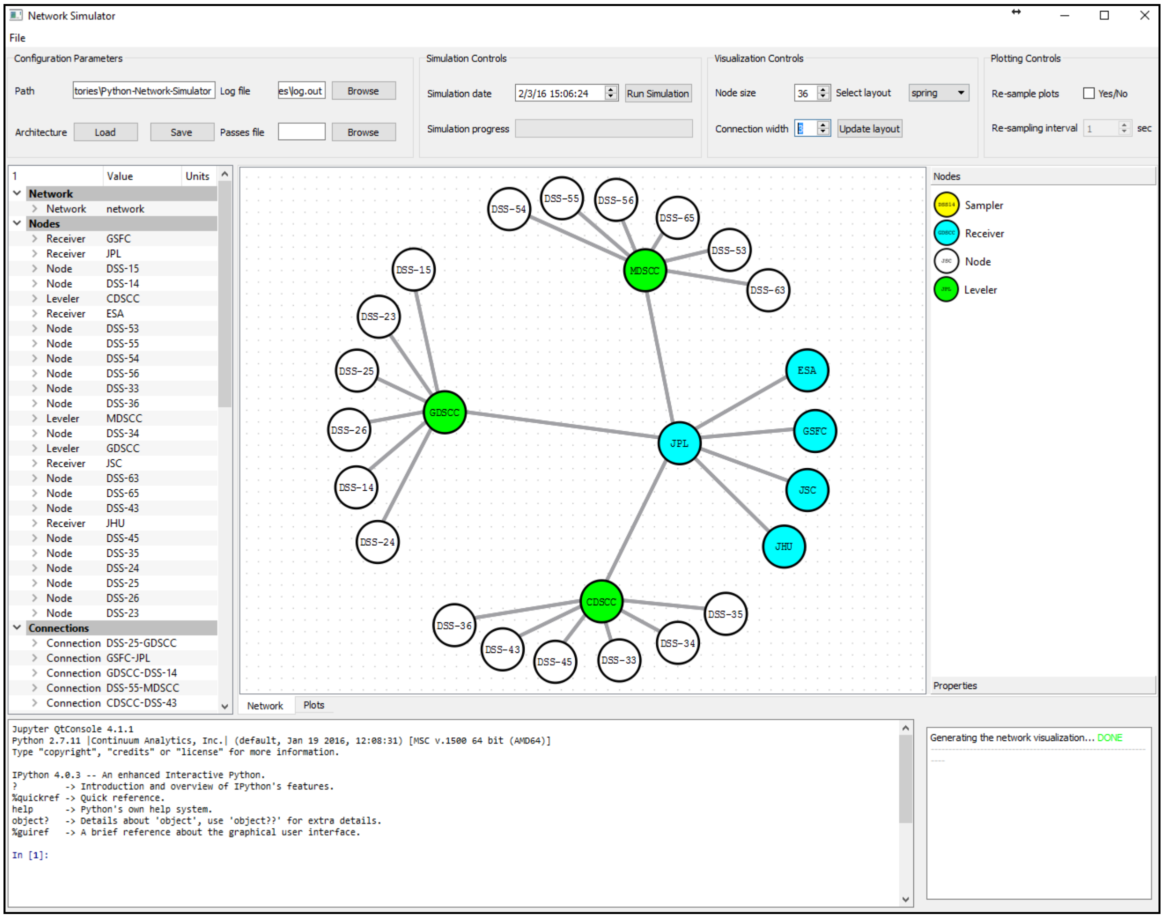

Figure 3. ArchNet Simulation Environment

B. ArchNet Software Architecture

ArchNet's software architecture has been developed in accordance to the following overarching goals: First and foremost, provide a flexible and computationally efficient simulation environment that can operate under a wide variety of network architectures and traffic profiles. Second, facilitate the user interaction through a robust Graphical User Interface (GUI) that allows rapid visualization of the network architecture and the simulation results. Third, ensure that simulation inputs and outputs can be provided and obtained through a wide variety of standard interfaces. And fourth, develop a software environment that can be cheaply and easily maintained and upgraded.

Based on these goals, ArchNet has been fully developed and implemented using the well-known open-source programming language Python [7]. For network simulation purposes, SimPy [8] has been utilized as the fundamental discrete-event engine that handles synchronization of messages and nodes across the network. On the other hand, PyQt [9] functionality has been extended in order to create an integrated simulation and visualization environment. This choice has also enabled the use of Matplotlib [10] as an efficient plotting system for visualizing and analyzing the simulation results. Finally, IPython [11] has been utilized in order to embed a simulation-synchronized Python console into ArchNet's GUI and consequently retain Python's scripting capabilities for advanced simulation data statistical analysis.

ArchNet has been implemented based on a multi-layered Object-Oriented (OO) software architecture. Except for supporting GUI and simulation-related classes, the majority of the software system encapsulates functionality using a highly modular approach: At the lowest layer, classes only implement simulation functionality. These original set of classes is then extended into the second layer in order to provide network simulation and visualization capabilities. Finally, a third layer provides the interfacing functionalities that allow the user to easily specify inputs and extract simulation outputs. Figure 4 provides a simplified view of the first two layers of functionality. Note that the simulation environment can be easily extended to include other types of nodes by simply sub-classing S*imNode* or one of its children. Similarly, different types of connections could potentially be defined (e.g. satellite link, terrestrial fiber optic) by sub-classing *SimConnection.*

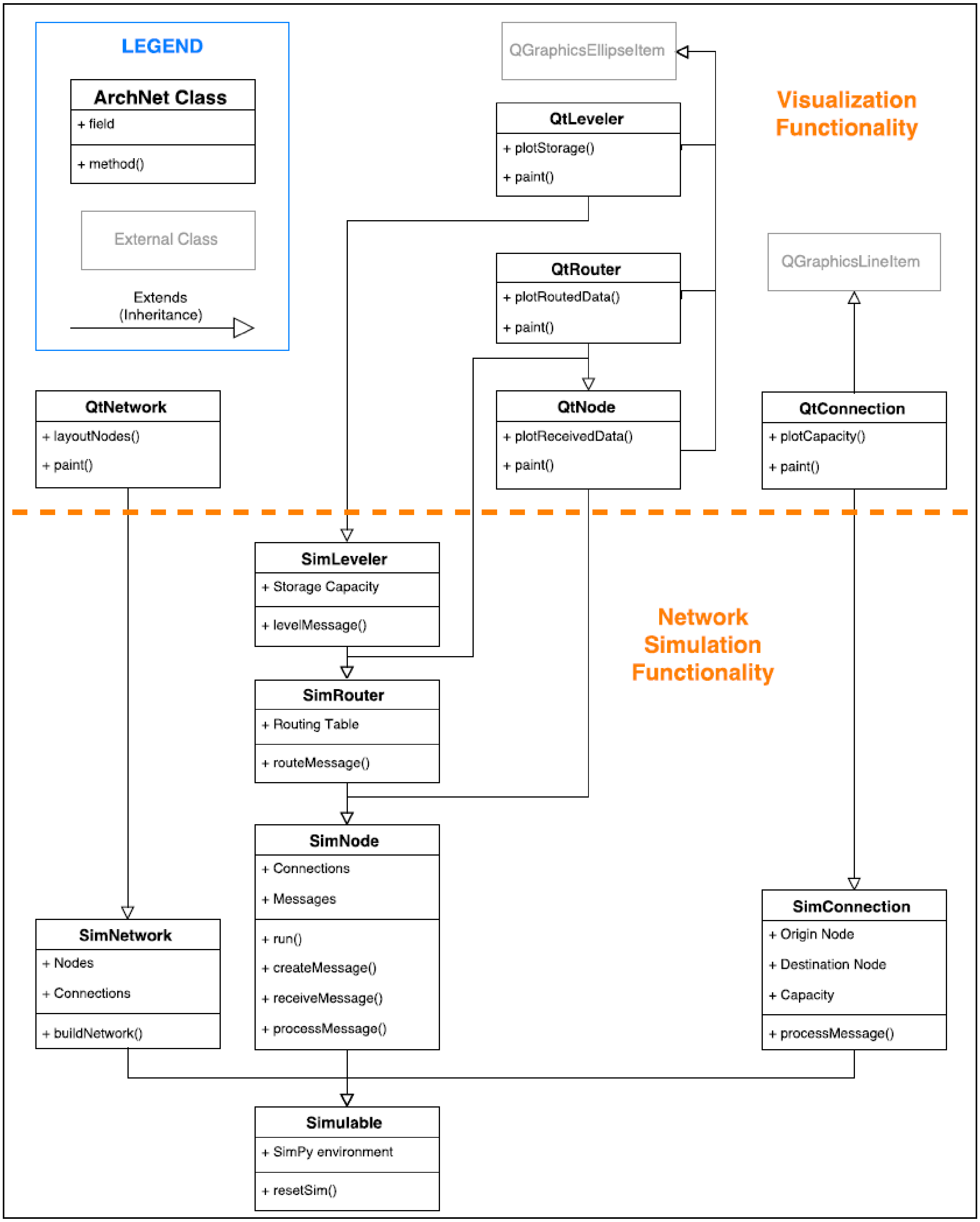

Figure 4. ArchNet Software Architecture

On the other hand, section A indicated that ArchNet is a high-level coarse-grain simulator. Consequently, information sent through the network is modeled at a high abstraction level – a message – rather than commonly used bits, frames or packets. A message is a generic information encapsulation construct that has six main properties: Identifier, origin, destination, start time, duration and data types, where each data type represents an independent of stream information that is subject to different communication characteristics and requirements (see section C for a detailed description). For routing purposes, messages are processed as a single unit since all information they contain has the same origin and destination. In contrast, levelers process messages by processing each data type individually and reshaping the data streams according to their respective latency requirements. Therefore, a message that initially contained two data types, one with no latency margin and another one with large latency allowance, can be split into two separate single data type messages that will be transmitted at different instants in time, have different duration and bandwidth requirements, but still share the same identifier and origin-destination pair.

C. Message and Data Type Definition

Having defined ArchNet's software architecture, this section describes how traffic can be specified in the simulation environment. Most network simulators define traffic through a set of source nodes that generate packets according to given stochastic or deterministic process. However, in space communication networks such as the DSN traffic is not random but given by a pre-negotiated schedule that clearly stipulates which mission will have support time from the network at each instant in time. In that sense, ArchNet utilizes the simulated tracking schedule (see section II) as an input in order to clearly define the start and end contact times between a remote mission and a DSN asset (also referred to as *pass*). This tracking schedule also specifies the expected pass data rate, as well as type of data being transmitted.

To specify the latter, a canonical classification of passes for the network customers has been defined based on current flight project operations and later extended to include unprecedented Mar exploration missions at Mars. This classification provides a simplified yet representative mapping between different types of missions, the passes they request and the data types in each of these passes. Table 1 provides an overview of the canonical classification assuming five data types are available[15]:

- Telemetry: Real-time health information from the spacecraft and its instruments.
- Quick-look science: Instrument data with limited latency requirement (30 minutes to 1 hour).
- Bulk science: Instrument data that can be returned over multiple hours (e.g. 8 to 12 hours).
- Voice and video: Transfer of voice and video, either full duplex real-time or buffered.
- Navigation: Radiometric measurements to be processed for obtaining a navigation solution.

Note that the obtained mapping and data volume fractions specify values for passes in which different data types are multiplexed into one single frequency band contact. If that is not the case (e.g. a contact is scheduled both at X and S-band), it is always assumed that the lower data rate data type is channeled through the lowest frequency band. Note also that passes for robotic missions are subdivided according to different mission phases, namely cruise, prime and extended science. Finally, navigation passes have been separated from all the others and represent only DOR measurements. As previously stated, this is the only form of navigation data that will generate a significant enough amount of data to be considered in this study.

D. ArchNet Utilization, Interfaces and Result Extraction

This section briefly describes how to utilize and interface with ArchNet, both from the perspective of defining inputs to the simulation and extracting meaningful results. As explained in Sections A and C, two types of inputs are required: The network architecture and the network simulated tracking schedule. The network architecture defines the set of nodes that compose the network, the functionality they implement and their respective connectivity with other parts of the system. In that sense, an XML file that contains a list of nodes and connections is sequentially parsed by ArchNet in order to fully define and visualize the system under consideration (see Figure 7 for an example). On the other hand, the network's simulated tracking scheduled can be specified with either an XML or an Excel file that lists the contacts between missions and the network, the time and duration at which they occur, the data destination node and the pass type they represent (specified in the format of *mission_type/mission_phase/pass_type*). Based on this information, ArchNet automatically utilizes Table 1 to define latency requirements for the different data streams being sent over the network and estimate their effect on the required bandwidth capacity.

Once the simulation has finished, the produced outputs can be analyzed using three complimentary approaches: First, the ArchNet's GUI provides a limited set of automated plotting capabilities that allow the user to request time

[15] Recall at this point that only return data from the spacecraft to the mission operation center is being considered here.

series plots, histograms and experimental cumulative probability distribution plots for all nodes and connections in the network. This information is also available through the embedded simulation-synchronized Python console, which can be used to save and process the simulation outputs. Finally, all plots from connections and nodes can be exported into comma separated and Excel files to be processed with external scientific software.

Mission Type	**Mission Phase**	**Pass type**	**Data Type**	**Data Volume Fraction**	**Latency Requirement[16]**
Robotic	Cruise	Telemetry	Telemetry	100%	5 seconds
	Prime	Telemetry	Telemetry	100%	5 seconds
		Science	Telemetry	10%	5 seconds
			Quick-look science	5%	1 hour
			Bulk science	85%	8 hours
	Extended	Telemetry	Telemetry	100%	5 seconds
		Science	Telemetry	10%	5 seconds
			Bulk science	90%	8 hours
Human	Prime	Exploration	Telemetry	10%	5 seconds
			Quick-look science	10%	1 hour
			Bulk science	80%	8 hours
		Astronaut	Telemetry	10%	5 seconds
			Audio/Video	90%	2 seconds
		Telemetry	Telemetry	100%	5 seconds
		Audio/Video	Audio/Video	100%	2 seconds
		Science	Quick-look science	10%	1 hour
			Bulk science	90%	8 hours
Any	Any	Navigation	DDOR	100%	6 hours

Table 1. Pass to Data Type Mapping

IV. Terrestrial Network Bandwidth and Buffer Statistics

The DSN terrestrial network topology that includes the DSN sites (and their antennas), the JPL Central, and the destination MOCs is shown in Figure 5. The green nodes[17] denote the DSN sites where store-and-forward operations are applied to the mission downlink data to reduce the terrestrial network bandwidth requirements. The white nodes[18] correspond to the antennas at each DSN site. Finally, the blue nodes[19] represent the MOCs, i.e. the destinations for the network data flows in the 30-day period of July 2039.

As described earlier in Sections I, II, and III, the end-to-end traffic flow simulation is executed as follows. The SCMM-derived mission downlink traffic is first generated. Data type modeling is applied to the mission data. Network tracking data required to support spacecraft navigation are also generated as part of the network data flow simulation. The network simulator ingests the SCMM-derived mission downlink traffic, and the network tracking data traffic, and models the store-and-forward mechanism that regulates the terrestrial network bandwidths without violating the latency requirements of different data types. The "leveled" mission data are sent to JPL before they are distributed to the respective MOCs.

[16] This requirement does not include the light time delay between the spacecraft and the DSN site. It refers only to the time it takes for data received at a DSN site to be delivered at the corresponding MOC.
[17] MDSCC - Madrid Site, GDSCC - Goldstone Site, CDSCC – Canberra Site.
[18] For ease of presentation, only a subset of the assumed number of antennas are shown in the figure.
[19] JHU – Johns Hopkins University, ESA – European Space Agency, GSFC – Goddard Space Flight Center, JSC – Johnson Space Center, JPL – Jet Propulsion Laboratory

The bandwidth time profiles of the ground links leaving the three DSN sites are shown in Figure 6. The storage time profiles of the three DSN sites are shown in Figure 7.

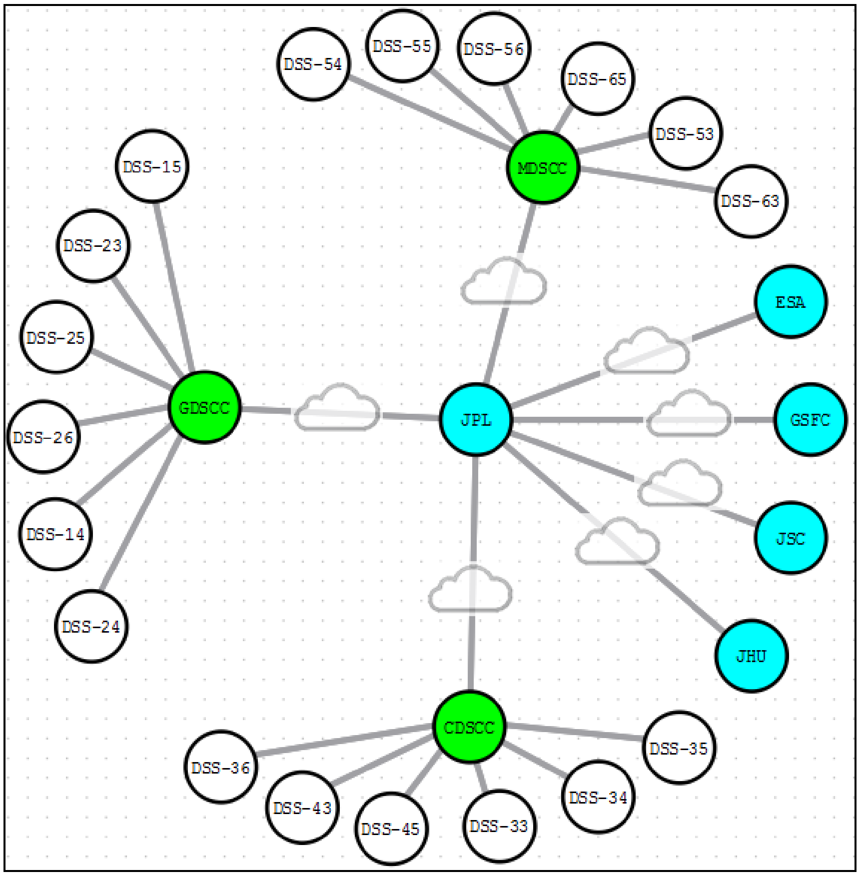

Figure 5. DSN Terrestial Network Data Flow Topology

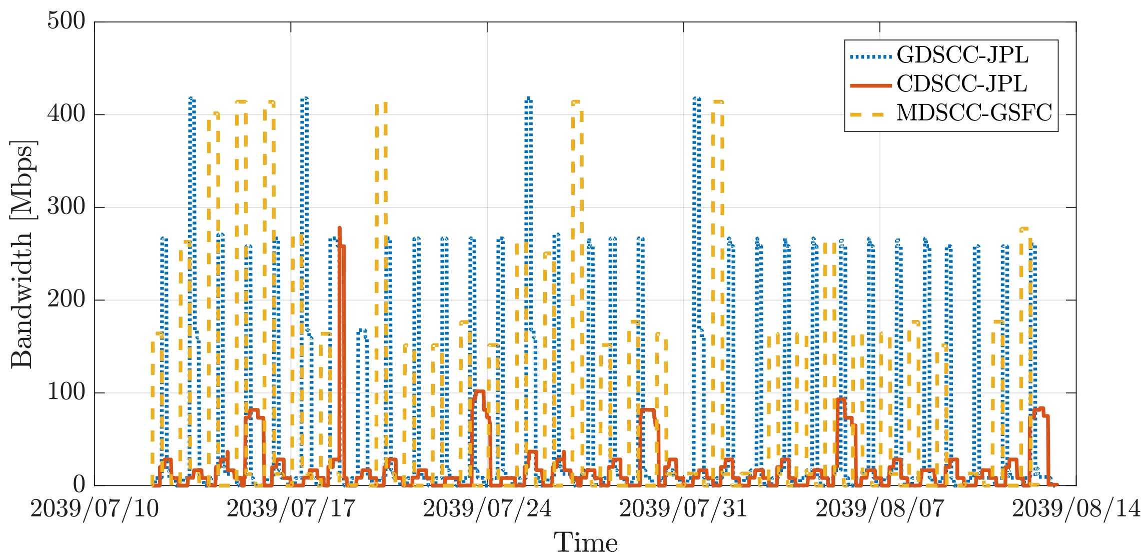

Figure 6. Bandwidth Profiles of DSN-JPL Terrestrial Links

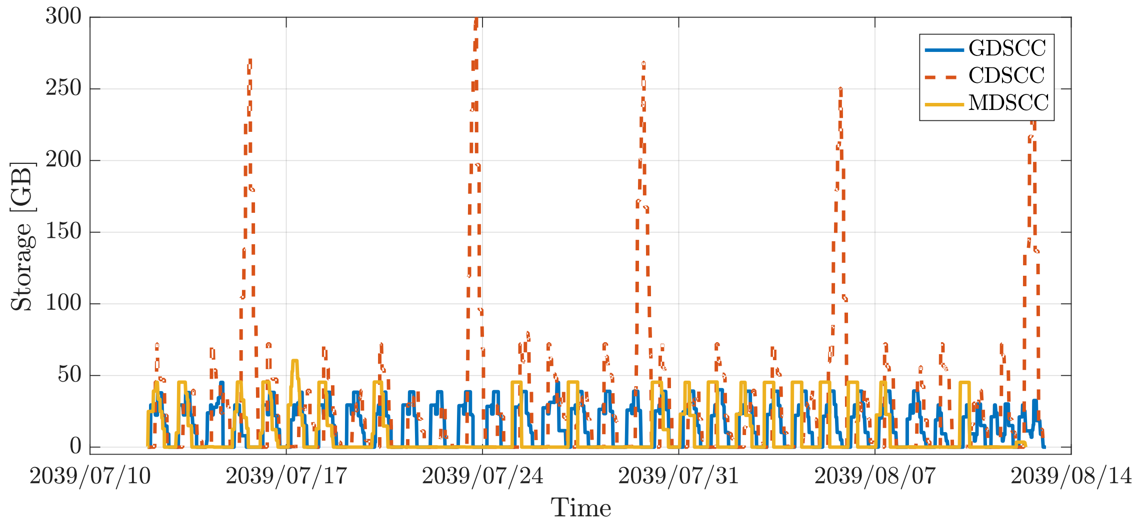

Figure 7. Storage Profiles of DSN Sites

Table 1 summarizes the percentage bandwidth statistics for terrestrial links leaving the DSN sites to JPL, and those for ground links leaving JPL to the MOCs. Table 2 summarizes the buffer statistics for the DSN sites that serve as the store-and-forward nodes.

Percentage (%) Bandwidth (Mbps)	**85%**	**90%**	**95%**	**99%**
GDSCC-JPL	258	260	267	418
CDSCC-JPL	28	65	82	102
MDSCC-JPL	177	263	414	414
JPL-JSC	244	244	388	395
JPL-GSFC	28.4	30.9	36.6	39.1
JPL-JHU	2.63	2.71	2.73	2.84
JPL-ESA	8.5	8.7	11	11.2

Table 1. Bandwidth requirements between DSN sites and JPL

Percent (%) Storage (GB)	**85%**	**90%**	**95%**	**99%**
GDSCC	30.8	38.4	39.0	39.2
CDSCC	55.3	72.0	138.2	250.4
MDSCC	45.2	45.3	45.4	45.4

Table 2. Storage requirements at DSN sites

Figure 8 shows the mission pass type statistics on number of contacts and contact time of the three DSN sites. Figure 9 shows the aggregate pass type percentages of the DSN.

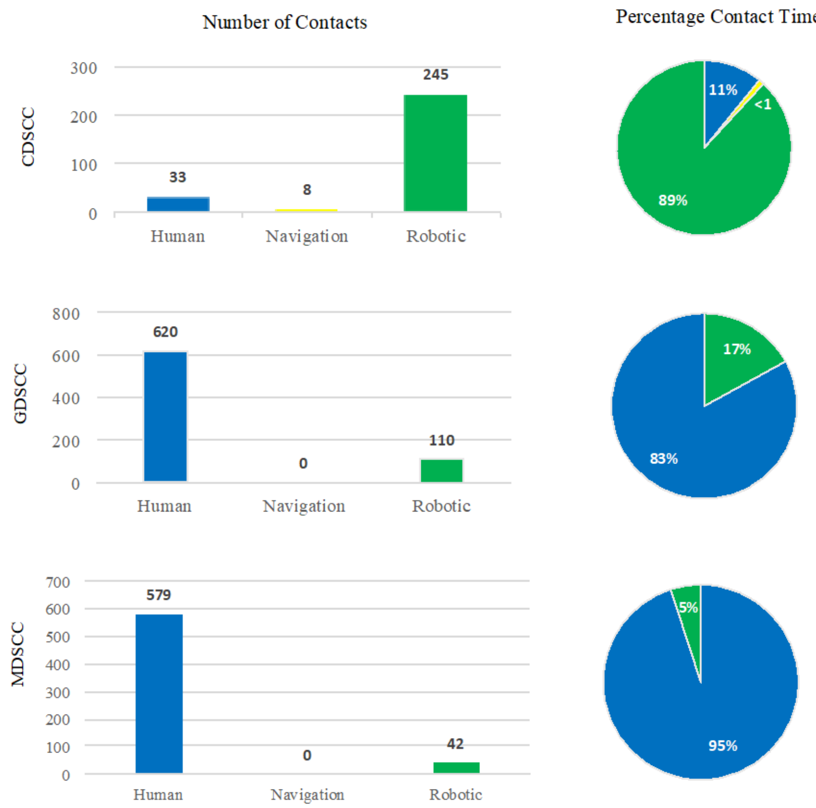

Figure 8. Mission Pass Type Statistics for Each DSN Site

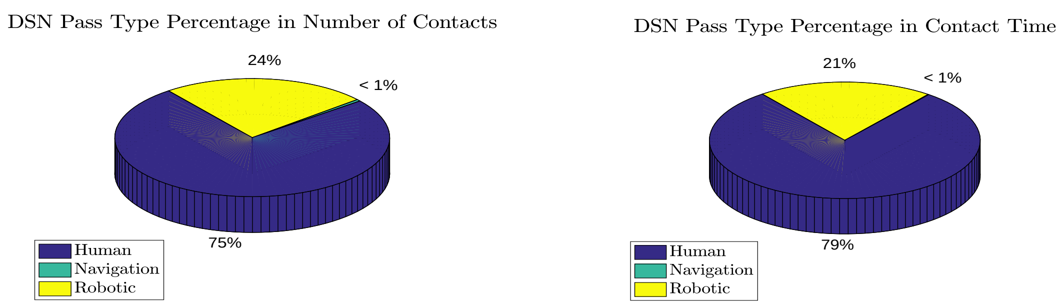

Figure 9. Aggregated Pass Type Percentage for DSN

The MOCs' statistics on number of contacts and contact time are shown in Figure 10.

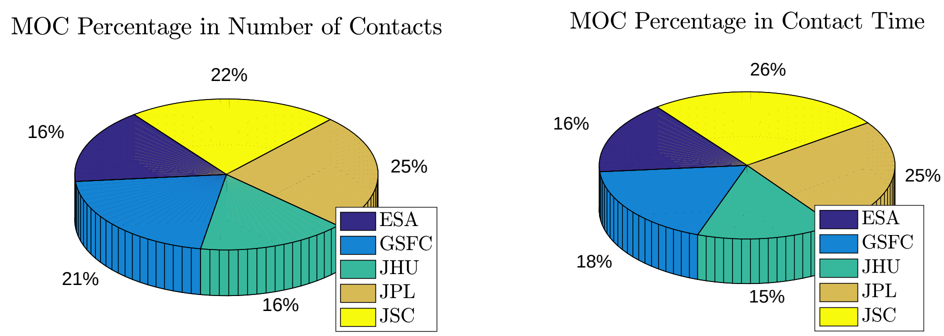

Figure 10. MOCs' Statistics on number of Contacts and Contact Time

Based on the above statistics generated from the traffic simulation, we have the following observations and interpretations:

1. The network bandwidth and storage simulations are derived from the SCMM-derived mission traffic for the 30-day period within July/August. The simulation time is probably too short to provide representative statistics for the purpose of terrestrial network design and planning. Nevertheless, this 30-day period is considered to be a busy month and, hence, provides a stressing case for the terrestrial network requirements for the DSN in the Human Mars Era.
2. In this 30-day simulation, Mars visibility is northern hemisphere biased. Therefore, the human Mars missions are mostly tracked by the Goldstone and Madrid sites, leaving the Canberra site to track the robotic deep space missions. As a result, the store-and-forward mechanism applied at Canberra can largely reduce the bandwidth requirements from that site to JPL, as compared to the other two DSN sites to JPL (see Figure 6 and Table 1). In turn, the storage requirement at Canberra is more than five times larger at Canberra than Goldstone and Madrid (see Figure 7 and Table 2). This, of course, can change in other time periods as the declination bias changes.
3. The human Mars missions' downlinks are dominated by high-rate low-latency audio/video data. As a result:
    a. The required ground bandwidths from Goldstone and Madrid to JPL are nearly four times as that of Canberra (Figure 6, Table 1).
    b. The required storage capacity of the Canberra site is much higher than those of the Goldstone and Madrid sites (Figure 7, Table 2).
    c. The ground link from the Madrid site to JPL exhibits a large percentage of unused time (Figure 6).
4. Over 75% of the DSN contacts and contact time are used for human missions (Figure 9).
5. The current plan to upgrade DSN terrestrial network to 40 Mbps bandwidth will not be sufficient for this 30-day scenario of Human Exploration Era, which require a bandwidth of 100 – 400 Mbps at different complexes.

V. Concluding Remarks and Future Work

In this paper, we describe the analysis and simulation effort to model the end-to-end traffic flow and buffering mechanism of the Deep Space Network (DSN) as a large-scale store-and-forward network in the Human Exploration Era. During this time DSN will provide communication and navigation services for human missions to distant celestial objects like the Moon, asteroids, and Mars. We leverage on the SCMM-derived mission traffic for a 30-day period in 2039, and apply novel store-and-forward modeling techniques to estimate the bandwidth and storage statistics of the DSN.

Note that this paper addresses only the July 2039 scenario of Direct-to-Earth (DTE) downlinks, and considers RF communications only. A more detailed study is currently underway that includes optical links in addition to the RF communications. We also plan to generate higher-fidelity SCMM-derived mission traffic models that include lunar and Mars relay communications (RF and optical) between surface assets and the relay orbiters. We expect the relay orbiters would implement the onboard store-and-forward function, and the study would provide the bandwidth and storage statistics that help to formulate the 'relays' storage and bandwidth requirements. The results of the study will be published in a future report.

Acknowledgement

The author would like to thank Timothy Pham for helpful discussions on DSN terrestrial network transition and upgrade plan. The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and at the System Architecture Laboratory, Massachusetts Institute of Technology.

References

[1] K. Cheung, D. Abraham, "End-to-End Traffic Flow Modeling of the Integrated SCaN Network," IPN Progress Report, 42-189, May 2012.

[2] K. Cheung, E. Jennings, J. Segui, "Coarse-Grain Bandwidth Estimation Techniques for Large-Scale Network," IEEE Aerospace Conference 2013, Big Sky, Montana, March 2013.

[3] Kobler, B., Berbert, J., Caulk, P., & Hariharan, P. C. (1995, September). Architecture and design of storage and data management for the NASA Earth Observing System Data and Information System (EOSDIS). In Mass Storage Systems, 1995.'Storage-At the Forefront of Information Infrastructures', Proceedings of the Fourteenth IEEE Symposium on (pp. 65-76). IEEE.

[4] Cruz, R. L. (1991). A calculus for network delay. I. Network elements in isolation. Information Theory, IEEE Transactions on, 37(1), 114-131.

[5] Matni, N., Tang, A., & Doyle, J. C. A Case Study in Network Architecture Tradeoffs.

[6] Sanchez Net, M., Selva, D., Cameron, B., Crawley, E., Seas, A., & Seery, B. (2014, March). Results of the mit space communication and navigation architecture study. In Aerospace Conference, 2014 IEEE (pp. 1-14). IEEE.

[7] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[8] Matloff, N. (2008). Introduction to discrete-event simulation and the simpy language. Davis, CA. Department of Computer Science. University of California at Davis. Retrieved on August, 2, 2009.

[9] Summerfield, M. (2007). Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming. Pearson Education.

[10] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science and engineering, 9(3), 90-95.

[11] Pérez, F., & Granger, B. E. (2007). IPython: a system for interactive scientific computing. Computing in Science & Engineering, 9(3), 21-29.

[12] G. Kazz, S. Burleigh, K. Cheung, B. Shah, "Evolution of the Mars Relay Network End-to-End Information System in the Mars Human Era (2030 – 2040)," to be published in the SpaceOps 2016.