A Demand Access Paradigm for NASA's Deep Space Network

Nihal Dhamani, Mark D. Johnston, Girly Lucena
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena CA 91109
nihal.n.dhamani, mark.d.johnston, girly.d.lucena @ jpl.nasa.gov

Abstract

NASA's Deep Space Network (DSN) is the primary resource for communications and navigation for interplanetary space missions, for both NASA and partner agencies. As part of an investigation into improved efficiency and responsiveness, we have been exploring and prototyping the infusion of a ”demand access” model into the DSN scheduling process. Today, DSN is fully pre-scheduled in advance, and many users rely on a stable schedule to plan their own spacecraft activities, weeks in advance of execution. However, a new class of missions is emerging that may not be scheduled as far in advance, and may be event-driven in coming across science targets at unpredictable times. These users could take advantage of an on-demand mechanism to download data. Simulations have shown that such a mechanism could improve latency (time from data collection to download) by 2x, as well as more efficiently utilize the available DSN antennas. In this paper, we describe a prototype of a demand access process and how it addresses the challenges of co-existing with a statically-scheduled body of missions, while providing the benefits of lower latency science data return.

Introduction

NASA's Deep Space Network (DSN) consists of three communications complexes, located in Goldstone, California; Madrid, Spain; and Canberra, Australia. Each complex contains one 70-meter antenna and three or four 34-meter antennas. These ground antennas are responsible for communications and navigation support for a wide range of scientific space missions, from those in highly elliptical earth orbits, to some beyond the solar system. In future years, DSN will also support human missions to the moon and beyond. The placement of the three DSN complexes allows at least one of them to be in view of any distant spacecraft at all times

The current DSN scheduling process (Johnston and Lad 2018) starts roughly six months ahead of execution when missions enter detailed requirements for their tracking. These requirements are submitted into the Service Scheduling Software (S3)(Johnston et al. 2014), where a variety of scheduling strategies are run to integrate the requirements into a single initial schedule. Next, a human scheduler, called the Builder of Proposal (BOP) makes further changes to the schedule, often eliminating hundreds of conflicts through a time consuming and labor intensive process lasting about a week. This is followed by a negotiation phase, also about a week in duration, where the schedule is released to all mission representatives who must collaboratively negotiate any changes or updates to the schedule. Any further changes to the schedule need to be mutually agreed upon by all of the mission representatives affected by the change.

Following the deconfliction and negotiation phases, the schedule is baselined and usually remains stable for about 22 weeks (4-5 months) prior to execution. This allows missions to use the schedule from the DSN as an input to their internal spacecraft planning and sequencing processes, which can be labor intensive and time consuming. For deep space missions with long light-travel times, sequences are pre-loaded on-board weeks ahead of time, and late changes can be difficult. Adding onto the already complicated DSN scheduling problem is the fact that in recent years, the increased number of missions and the increased data return from missions has led to high oversubscription of DSN resources, leading to a very limited availability of DSN antenna time.

Some newer mission concepts and proposals would benefit from a more dynamic scheduling process. Smallsats are lower cost missions that may have a smaller operations team, be power-limited, and have some constraints that may not be accurately modeled months in advance. Additionally, some science missions are proposed that would respond to ephemeral phenomena such as outbursts on comets or asteroids, or astrophysical transients such as flares or bursts. For these kinds of missions, predicting the times when a future download would be needed is not possible, and yet their allocation of DSN time would not be large, likely only a few tracks per week. Time would be wasted if the mission had nothing to report, and conversely, if there was something to report but the next scheduled track was a week away, there would be a significantly delay introduced in getting data to the ground so that follow-up observations could be scheduled. As such, the traditional static pre-scheduling method described above is not favorable for smallsats and event-driven missions as it lacks adaptability in response to chang-

---

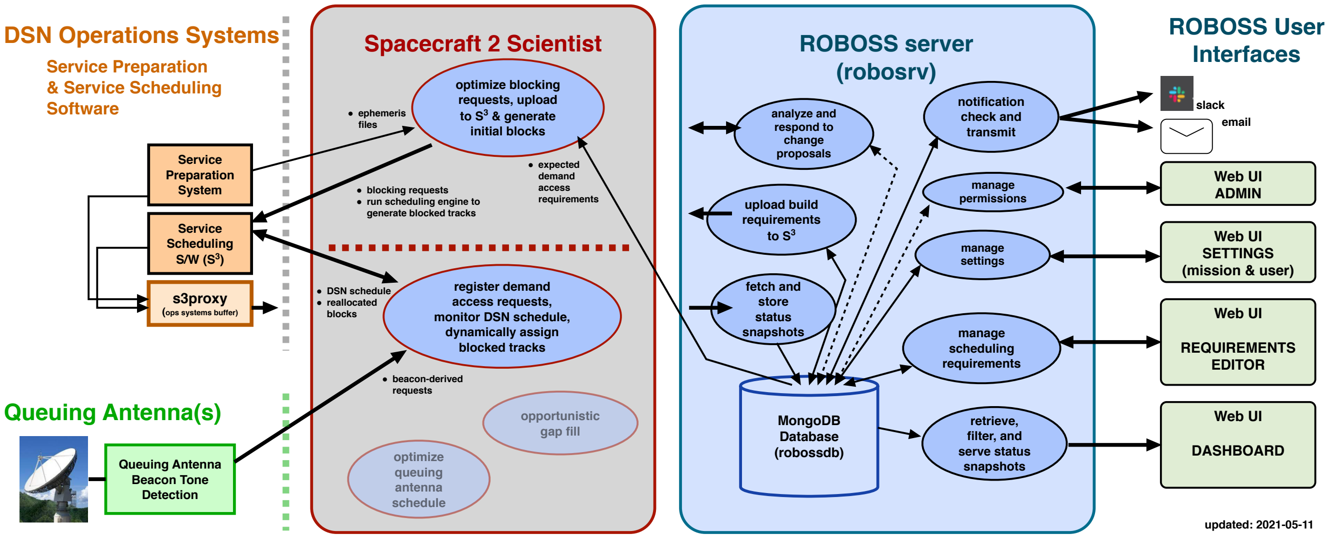


Figure 1: Architecture of the DSN demand access prototype as an extension of the Robo Scheduling Software (ROBOSS) framework.

ing events.

The proposed solution would be to incorporate an on-demand communications approach, where spacecraft could request tracking time and be scheduled on an as-needed basis, referred to as "demand access" (Johnston and Wyatt 2017). The concept of demand access and responding to changing events is not novel and many systems have been developed to accomplish these tasks. For instance, the Demand Access Network Scheduler (DANS) was designed to be capable of automatically rescheduling antenna and subsystems in the event of changing track requests or equipment outages (Chien, Lam, and Vu 1997). Additionally, NASA's Space Network also defines its own Demand Access Service (DAS) for some usage of the Tracking and Data Relay Satellite System (TDRSS), where spacecraft orbiting near the Earth can request tracking time through a handshake protocol (Gitlin, Kearns, and Horne 2002). Similarly, the Space Mobile Network also utilizes the concepts of user-initiated services (UIS) to request links to communications resources (Israel et al. 2018). These services are non-preemptive and provide for late allocation of available capacity.

While there has been previous work done in the domain surrounding demand access scheduling, the increasing oversubscription of DSN resources and human-driven scheduling process of the DSN present unique challenges. Ultimately, any demand access paradigm for the DSN must be able fit within the semi-manual scheduling framework and needs to coexist with stable long-term allocations that many missions require in order to develop their on-board command sequences. As such, the demand access work described in this paper extends upon the traditional demand access models by introducing a way to roughly model anticipated demand, generate optimized tracks to accommodate the anticipated demand, and reallocate holding time in the schedule to meet an anticipated demand when a request comes in, all while existing within the regular DSN scheduling process. For the DSN, the cost of preemption is high in terms of disruption to deep space mission operations, and so pre-allocation for anticipated demand provides a buffer against late schedule changes. A simulation study of this demand access scheduling approach for the DSN (Hackett et al. 2021) was carried out on a set of 10 missions which showed promising improvement across key indicators. Mainly, the study found that for some mission sets, the mean duration between the time of data collection to the start of the downlink improved by a factor of 2 while also decreasing the percentage of dropped science data to 0. These preliminary simulation results give credence that this demand access model is worth exploring.

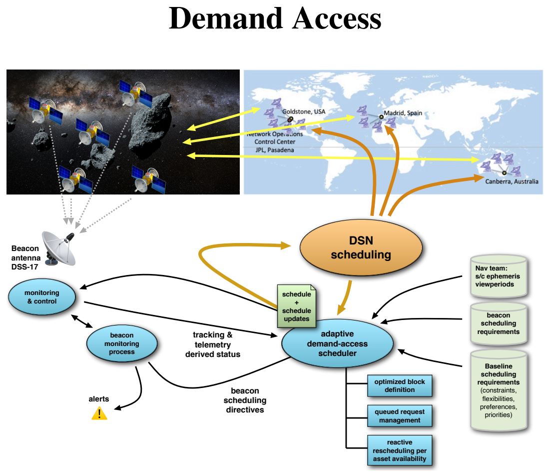


Figure 2: An overview of the demand access concept for DSN. A fleet of autonomous spacecraft, for example exploring the asteroids, sends beacon tones requesting contacts if needed. Pre-allocated optimized tracks are in the DSN schedule and allocated to a specific mission just in time.

While the notion of "demand access" for space communications networks has been available for some time for Earth orbiters and geosynchronous relay satellites as discussed in the introduction, incorporating the demand access model into the DSN presents some unique challenges that must be met to make such an approach work, most notably:

- existing users who participate in the static DSN scheduling process rely on a stable schedule, and so the infusion of demand access must not disrupt these users
- must be able to exist within the context of highly oversubscribed DSN resources
- a key objective of demand access is to utilize DSN resources more efficiently, and to service DSN users in a more timely manner – so approaches that do not accomplish these objectives will not be viable

Since DSN antenna time is a scarce resource, the concept for demand access (Figure 2) incorporates a secondary antenna at another site with a smaller antenna diameter and shorter integration time that can detect a "beacon" tone from a spacecraft (Wyatt et al. 1998). These tones can have a small range of values, usually 4 to 8, that can be used to indicate that the spacecraft is healthy or not, and that it has data to download and with what level of urgency. The tone is a one-way signal – it is not feasible to return an acknowledgement. Beacon tones are already operational, most notably on the New Horizon's spacecraft which utilized beacon tones on the way to Pluto (Kusnierkiewicz et al. 2005). Once a tone is received that indicates a need for contact, the request is put in the queue for allocation from a set of pre-specified antenna blocks of time in the schedule. Thus, this smaller antenna is called the "queuing antenna". For this study, the Morehead State University 21-meter antenna is considered the queuing antenna since it has been outfitted with DSN-compatible S- and X-band equipment.

Incorporating the beacon-tone demand access model into the DSN consists of two separate processes, as described in the following sections.

Inputting Mission Requirements

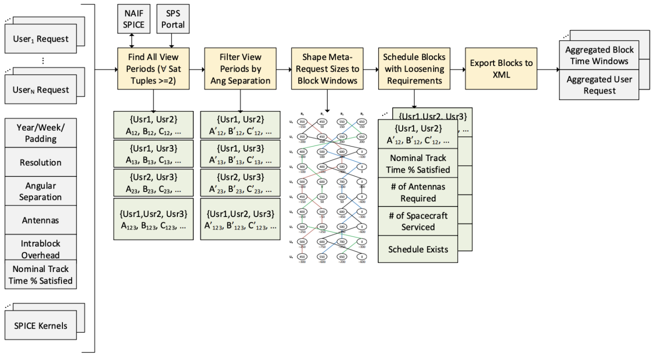


Figure 3: An overview of the blocking algorithm (Hackett, Bilén, and Johnston 2018)

.

At the start of the regular DSN scheduling process, missions and science users enter their scheduling requirements into S3 for integration, deconfliction, and negotiation. Similarly, for the demand access scheduling process, users enter their mission requirements into a scheduling tool called ROBOSS (Robo Scheduling Services) that was developed to support smallsat scheduling in the context of DSN processes, including requirements submission and monitoring. In order for spacecraft resources to be reserved for demand access users for near real-time allocation as described below, they must be *blocked* out, aggregated together based on requirements into a larger "pseudo-spacecraft" which contends for track time just like any other mission in the DSN scheduling process.

To accomplish this, we have extended ROBOSS (Figure 1) with the capability of recording, for each demand access mission, the *expected* utilization (antenna resources, time durations, number of tracks per week, separation, etc.) that is anticipated. These requirements are then sent to a "blocking" algorithm that was developed to combine the visibility times for multiple spacecraft and find times that can be shared among them. The blocking algorithm (Figure 3) comprises of the following main steps: extracting spacecraft pointing positions, finding combinations of spacecraft within the specific blocking angle, aggregating spacecraft requests based on the blocking time windows, and using a backtrack scheduling algorithm to schedule the aggregated requests (Hackett, Bilén, and Johnston 2018). The resulting output of the blocking process is an XML file defining the blocked requirements of the aggregated "pseudo-spacecraft" which are easily importable into SSS and used in the DSN scheduling process as normal.

To summarize, the resulting process is fully automatic – given the set of missions and their expected demand access requirements, it generates and uploads DSN scheduling requirements to reserve blocks of time that are optimized for sharing among demand access users, while accommodating their overall expected tracking needs. These requirements would be included in the normal deconfliction and negotiation process and would be present in the schedule as generic, i.e. not allocated to a specific mission until the second process described below.

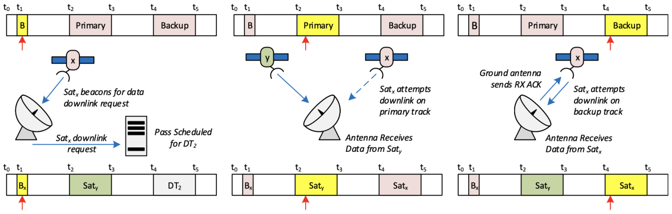
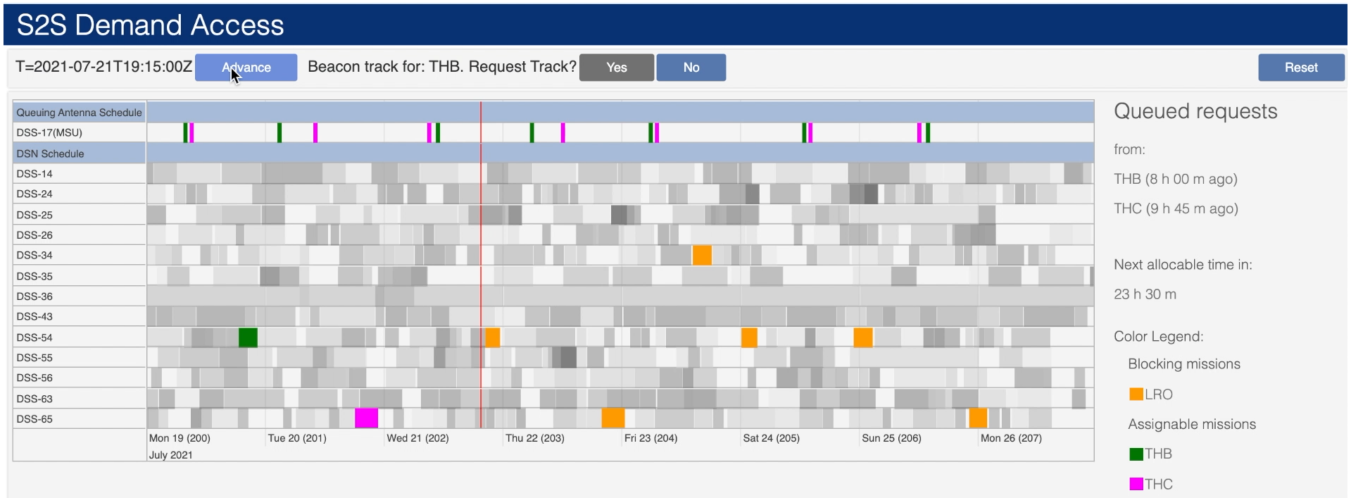


Figure 4: An overview of the near real-time reallocation of generic tracks to specific spacecraft based on the request queue (Hackett, Bilén, and Johnston 2019)

.



T=2021-07-21T19:15:00Z

Beacon track for: THB. Request Track?

Queued requests

from:

THB (8 h 00 m ago)

THC (9 h 45 m ago)

Next allocable time in:

23 h 30 m

Color Legend:

Blocking missions

🟧 LRO

Assignable missions

🟩 THB

🟪 THC

Figure 5: Demonstration GUI for simulating queuing antenna tone receipt and triggering dynamic track allocation.

Near Real-Time Dynamic Allocation

The second process operates closer to real-time and is responsible for allocating generic tracks to specific spacecraft depending on the requests received via the queuing antenna. Each spacecraft has on-board a schedule of future potential tracks that could be allocated to it, as well as a queuing antenna schedule for sending the beacon tone signals. The dynamic allocation process keeps track of tone requests, as well as of unallocated blocks in the schedule. At some trigger time ahead of the start of each block (expected to be a few hours), the dynamic allocation process makes a decision as to which spacecraft gets the block, then automatically converts the block in the schedule from a generic track to a specific spacecraft track (Figure 4). The queue is updated and the process repeats. Should the block not be needed (no queue entries) there are various options for what to do (release the time, allocate to engineering tasks, etc), and this is an area still being explored. On the other hand, should there be more demand than available blocks, the spacecraft can try again to request an allocation at the next opportunity.

To interface the various components in this workflow, a demand access API was developed in Python3 using Flask (Grinberg 2018). It enables the simulation of the beacon tone request from spacecraft, maintains and the updates the spacecraft request queue, assigns the generic track to a specific spacecraft, and handles the committing and background interactions with S3 to make reallocation as seamless as possible. The API maintains state and context throughout the demand access workflow while having easily configurable options such as setting the workspace environment and the trigger time before the start of each block. Currently, the request queue operates on a FIFO ordering where the earliest spacecraft request takes priority over all others when multiple spacecraft are competing for the same generic track. However, since this work is an initial prototype, there will possibly be future changes to the specific workings of the API such as deciding which spacecraft request takes priority. As such, the API is designed to be extensible and flexible to future changes as development and research continue.

Additionally, for demonstration purposes, we have implemented a web-based graphical user interface (GUI, Figure 5) that interacts with the demand access API in the backend. The GUI and API are configured with a very simple example consisting of two spacecrafts, THEMIS-B (THB) and THEMIS-C (THC), that have already gone through the blocking process to generate the generic unallocated blocks which are shown as the highlighted orange tracks in the schedule. The GUI also showcases the schedule of the queuing antenna which would receive the one-way beacon tones from the spacecraft and also shoes the current set of requests in the request queue. Additionally, the GUI contains controls to simulate the queuing antenna receipt of beacon tone requests for the next available track and allows the user to step through a week-long simulation of dynamic block allocation. As users advance through the week and choose to simulate spacecraft beacon tone requests, the near-time reallocation of the generic tracks is witnessed. One thing to note is that for simplicity's sake, we are assuming that both THB and THC can be assigned to any unallocated track; however, for more complicated scenarios this would not always be the case and the generic tracks would contain metadata specifying which spacecraft it can be allocated to.

Results and Future Work

This approach showcases an initial prototype of how dynamically allocated tracks can co-exist with static pre-scheduled tracks. We have implemented the essential aspects of the two main processes of demand access as described above, where tone requests are simulated via a GUI and API for demonstration purposes. We have been able to demonstrate:

- entry of expected mission requirements, and generation of blocked requests for the DSN scheduling system – expressed as XML and automatically uploaded into the DSN scheduling software system, S3
- a series of dynamic block allocations based on a user-

specified response to a set of simulated beacon tone requests

Since this work describes an initial prototype, there remains much work to be done along with answering a few key questions brought up by the demand access paradigm. One key area of focus would be determining the criteria for deciding which spacecraft to pick when multiple are competing for the same generic track. Our work can be extended by attaching different priorities to different types of beacon tones, and perhaps even attaching priorities to missions themselves. Figuring out the best criteria will involve running a myriad of simulations on different prioritization schemes and selecting key metrics to rank the outputs. Additionally, as discussed earlier, another area of focus is figuring out how best to utilize the tracking time in blocks that remain unused as a result of no spacecraft requesting the time.

In parallel with some of the future work described above, a next step in development of the demand access model is to work on the integration of an actual queuing antenna – the 21-meter dish at Morehead State University in Kentucky. Along with the possibility of an actual test with a spacecraft, this will allow for further refinement of the operations concept, and will also help with identifying gaps in the approach.

Another important discussion to have is how the DSN demand access model would affect operations and ground data systems (GDS). Current operations and GDS are manual and complex. On-demand DSN capabilities would both enable and drive the need for more automated GDS and operations systems. Such systems are already being explored, including an on-demand cloud-based GDS where downlinked data can flow to the cloud, enabling mission operators to process science and telemetry data directly on the cloud. An on-demand DSN system combined with an on-demand GDS system would enable missions to operate with far higher levels of automation, enable better utilization and scaling of DSN resources, and improve efficiency of operations on the ground.

Acknowledgements: This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Chien, S.; Lam, R.; and Vu, Q. 1997. Resource Scheduling for a Network of Communications Antennas.

Gitlin, T.; Kearns, W.; and Horne, W. D. 2002. The NASA Space Network Demand Access System (DAS). In *SpaceOps 2002*.

Grinberg, M. 2018. *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.".

Hackett, T. M.; Bilén, S. G.; Johnston, M. D.; and Wyatt, E. J. 2021. Demand Access for NASA's Deep Space Network: A New Paradigm for Operations. In *SpaceOps 2021*.

Hackett, T. M.; Bilén, S. G.; and Johnston, M. D. 2018. Spacecraft Block Scheduling for NASA's Deep Space Network. In *SpaceOps 2018*.

Hackett, T.; Bilén, S.; and Johnston, M. D. 2019. Investigating a Demand Access Scheduling Paradigm for NASA's Deep Space Network. In *11th International Workshop on Planning and Scheduling for Space (IWPSS)*, 51–60.

Israel, D.; Roberts, C. J.; Morgenstern, R.; Gao, J. L.; and Tai, W. 2018. Space mobile network concepts for missions beyond low earth orbit.

Johnston, M. D., and Lad, J. 2018. Integrated Planning and Scheduling for NASA's Deep Space Network – from Forecasting to Real-time. In *SpaceOps*.

Johnston, M. D., and Wyatt, E. J. 2017. AI and Autonomy Initiatives for NASA's Deep Space Network (DSN). In *IJCAI AI in Space Workshop*.

Johnston, M.; Tran, D.; Arroyo, B.; Sorensen, S.; Tay, P.; Carruth, J.; Coffman, A.; and Wallace, M. 2014. Automated Scheduling for NASA's Deep Space Network. *AI Magazine* 35:7–25.

Kusnierkiewicz, D. Y.; Hersman, C. B.; Guo, Y.; Kubota, S.; and McDevitt, J. 2005. A description of the Pluto-bound New Horizons spacecraft. *Acta Astronautica* 57(2):135–144.

Wyatt, E. J.; Hotz, H.; Sherwood, R.; Szijarto, J.; and Sue, M. 1998. Beacon monitor operations on the deep space one mission. Technical Report 98-0712, JPL.