A Demand Access Protocol for Space Applications

Jay L. Gao, Dee Leang
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91109
Jay.L.Gao@jpl.nasa.gov, Dee.Leang@jpl.nasa.gov

Abstract—This paper describes a demand access protocol for space communications, which is a messaging procedure that facilitates the exchange of resource requests and grants between users and service providers. A minimal set of operational and environmental needs and constraints are assumed since the intent is to keep the protocol flexible and efficient for a wide-range of envisioned NASA robotic and human exploration missions. The protocol described in this document defines the message format and procedures used to ensure proper and correct functioning of a demand access communications system, which must operate under customized resource management policies applied by the users and service providers. This protocol also assumes a minimal set of capabilities from the underlying communications system so that no unique requirements are imposed on the communications sub-systems. [1] [2]

TABLE OF CONTENTS

1. INTRODUCTION

The next generation of robotic and human exploration missions will greatly benefit from a demand-driven networking paradigm, whereby the communication "turn-around" time – the response time between scientists and remote instruments or mission control on Earth and astronauts – is on the order of minutes and seconds instead of hours or days. The capability for rapid re-planning and re-tasking of communications systems based on real-time needs brings flexibility to missions, improved operability, and a higher degree of robustness and autonomy during un-planned, contingency situations.

A demand access protocol provides the mechanism whereby communication resources can be dynamically assigned based on users' requests. This includes both rapid response to immediate demands arising from contingencies or opportunistic science events, and advanced reservation of communications sessions that ensure long-term availability of resources for autonomous science and mission activity planning. Demand access technology improves the fundamental tradeoff between efficiency and flexibility by allowing multiple spacecrafts to request and release access to a common communication service provider on a dynamic, need-driven basis. Demand access eliminates the drawback of static allocations that are often overly conservative and tightly-coupled, where even small adjustments in resource assignments can be quite disruptive to other users. Demand access can also respond quickly to changing user needs under time-varying weather conditions and ground equipment outages so that as many missions as possible can be supported.

The current Deep Space Network (DSN) ground infrastructure is highly subscribed and will require automated scheduling capabilities in order to support the growing number of missions in the near future. While scheduling and planning methodology and tools have been developed [1], they still require a messaging protocol to automate the exchange of resource needs and availability in real-time in order to fully utilize the responsiveness and efficiency of a demand-driven system. For example, if a modified schedule generated on the ground still requires days or weeks of lead-time before it can be sent to a spacecraft and executed, then the benefit of demand-driven communication is lost.

The current state of practice in space mission operation is based on command and control, where the ground directs the execution of a well-planned schedule of activities, including communications scheduling, for the space segment. The basic premise is that the ground has far more resources and capabilities to analyze situations and make correct decisions. Here, the fundamental time constant in the command and control loop is the communications round-trip time plus the latency of the decision process on the ground, with the latter sometimes significantly higher. To minimize the impact of this slow control loop in a bandwidth-constrained environment, mission planners generate long series of commands (i.e. sequences of activities) that can be uplinked to a remote spacecraft and executed at predetermined time instances in the future. The efficiency of this system, however, reduces the flexibility for resource sharing, especially for unplanned events. The lack of flexibility means resources cannot be assigned as

1———————————

[2] IEEEAC paper #1313, Version 3, Updated December 22, 2006

needs arise; instead, a pre-determined estimation of the need is used, which is often generated many days or even weeks in advance. This experience has been gained from the successful deployment of relay orbiters around Mars and corresponding lessons learned [2], which have opened up the possibility of rethinking how improvements can be made in the responsiveness of the communications infrastructure around remote regions of exploration and on Earth.

Two elements are crucial in order to introduce flexibility into the communications infrastructure. Firstly, a more efficient and rapid scheduling process is required. In the case of the Mars Exploration Rovers and the 2001 Mars Odyssey orbiter, the scheduling process for relay service has been labor-intensive; it involves personnel from both missions to have regular meetings to discuss the configurations of supported passes in the coming days or weeks. In general, automation should be introduced wherever possible and policy-based resource management should be deployed. Depending on the time constant and application domain, resource management can occur on sub-second time scale. Many multiple access communications systems today, especially for short-range wireless environment, conduct frame-by-frame negotiation of channel access between users and service access points based on service-level agreements or policies. In the space domain, due to the long propagation delay and the nature of its physical environment, such a level of resource management is generally not necessary. However, having a resource management entity that is able to make rapid decisions based on resource needs and availability would be the first step in moving away from labor-intensive planning.

The second element is establishing a common method for exchanging requests and grants between users and service providers. This means that requests and grants are prepared, transmitted, received, and processed in a standardized, automated fashion. The standardization messing format and procedure will have significant impacts across the entire network. It means that any two spacecrafts can interoperate and issue resource requests and responses to each other with the flexibility of embedding these messages either within an on-going data stream or over a separate signaling channel. This can raise the level of collaboration between missions and support the gradual build up of a network from a fleet of spacecrafts managed by individual projects. The combination of dynamic resource management and standardized demand access messaging can improve the time-constant for dynamic resource assignment in space.

The earliest concept of demand-driven communication is demonstrated by the Deep Space 1 (DS-1) mission [3] via the beacon monitor operation. The idea is that when the spacecraft enters certain operational states and requires communication with the ground, it transmits one of four tones (carrier only) to a separate network of monitoring antennas on the ground. Each tone represents a distinct level of urgency and communication needs. The concept has successfully demonstrated how demand-driven communication can facilitate autonomous spacecraft operation. However, being restricted to a finite set of tones means that the number of request types is limited and no mechanism exists for two-way negotiations. Since DS-1, most missions has been able to operate outside a demand-driven paradigm until the Mars relay network concept lead to the conception of a dedicated relay orbiter for Mars. The Mars Telecommunications Orbiter [4] mission, later canceled due to budget constraints, began the process of re-examining the concept of demand-driven communication. The complexity of communications schedules required to support autonomous rover surface activities can be greatly reduced by having an automated, in-situ configuration of the rover-orbiter link. Even in the absence of competition for link access with other users, automated selection of data rate and connection time based on the rover's needs provides a whole new level of flexibility and labor savings in mission operation. When multiple rovers or landers are introduced, scheduling of requests can be resolved in-situ based on priority. These envisioned networking functionalities require two-way handshaking and exchange of information that cannot be provided by a beacon approach. The lunar surface network for human exploration will also greatly benefit by having this technology.

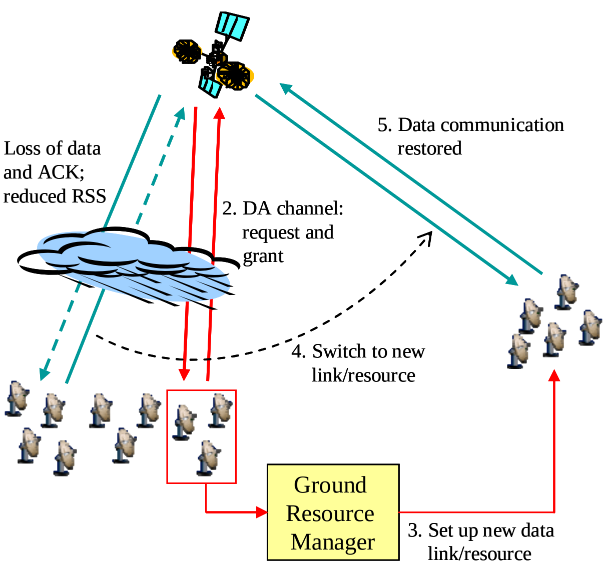


Figure 1: Demand access usage scenario supporting link recovery.

It is conceivable that in the near future, most missions will still operate in a deterministic fashion where most communication passes are pre-planned. However, demand access technology can play a crucial role in link layer recovery. A potential scenario could involve the deep space link between a remote spacecraft and the DSN. For example, a spacecraft could send telemetry back to Earth, receive low rate commands, and provide acknowledgements from the uplink via Ka-band. By monitoring the received signal strength (RSS) on the uplink, as well as analyzing the acknowledgement frames (or the lack of them), the spacecraft could decide that the link was down; this could be due to weather events, ground equipment failure, or other factors. The spacecraft can initiate the recovery process by first switching over to a more robust demand access signaling channel to issue a new service request message to the DSN. At this moment, depending on the priority of the data, weather conditions, and the availability of resources in the DSN, the spacecraft would desire a certain level of service and the DSN would be able to support a certain level of service. When the request message from the spacecraft is received by DSN, the "need" of the spacecraft and the "capability" DSN will be matched to see if a new link can be established. If so, the DSN's resource manager can respond on the demand access signal channel with all the necessary information for the spacecraft to switch to a different bandwidth, modulation, coding, data rate, etcetera, and even use a different ground station or a set of ground stations (in an array) if necessary. If the requested service cannot be met, then optional negotiations can be conducted to determine an acceptable resource allocation or cancel the current pass. The ability to conduct such spontaneous and dynamic requests, grants, and direct negotiations of resources between the spacecraft and DSN can be provided by a protocol supporting demand access.

This paper focuses on the description of a demand access protocol that facilitates the exchange of request and replies between network elements. It is assumed that decisions regarding initial requests and service grants are made by separate resource management entities.

2. Protocol Description

The proposed Demand Access Protocol (DAP) defines a messaging procedure that can be used to negotiate and reserve resources in advance. It is intended to operate between two entities, one of which is a requester and user of resources, while the other is a provider of resources. A minimal set of basic operational and environmental needs and constraints are assumed since the intent is to provide both flexibility and efficiency for envisioned National Aeronautics and Space Administration (NASA) missions.

Terminology

To simplify the description of DAP, several common terms must first be defined.

- A **User** is a DAP entity that requests and uses resources.

- A **Provider** is a DAP entity that provides resources.

- A **message** is a fundamental unit of information exchange between Users and Providers. Messages carry important information required to communicate requests, replies, commands, and acknowledgments.

- A **session** refers to the bounded period of time from when resource negotiation first takes place to when resource negotiation or usage terminates between a User and a Provider. Sessions are uniquely identified by special IDs so that a User and a Provider may have multiple simultaneous sessions with each other, as well as with other DAP entities.

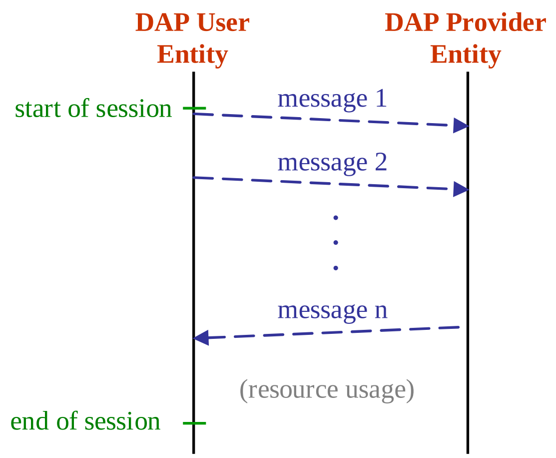


Figure 2: An example of a DAP session with multiple message exchanges between a User and a Provider. The session begins when the first message is sent and ends when the resource usage period has completed.

Message Types

DAP provides four basic message types that facilitate processes required for resource negotiation. All messages contain the basic elements required for unique identification, such as User and Provider IDs, session IDs, and message sequence numbers. Parameters specific to each message type are summarized below.

- **Request** messages identify the circumstances for which resource requests are made. Important fields indicate the priority of the request, the persistence time, and the type/values of resources being requested. These messages are created only by Users.

- **Reply** messages are created in response to request messages, and indicate whether a request has been accepted or rejected. If a request has been accepted, a reply confirms the circumstances for the reservation. If a request has been rejected, a reply provides reasoning for the decision and optionally indicates hints regarding what an acceptable request may have been. These messages are created only by Providers.

- **Command** messages signal that a previously accepted reservation request has been modified by the Provider. These messages include information on what the resource types/values of the existing reservation have been changed to, and are only created by Providers.

- **Acknowledgment** messages are special indicators used simply to inform the sender of any of the above messages that the message was received successfully. No other information is communicated.

Operational Modes

The quality of service offered by DAP is selectable on a message-by-message basis. As such, request, reply, and command messages may be transmitted via a one-way or two-way mode.

The one-way mode of transmission of a message means that the message is sent on a best-effort basis. It is hoped that the message arrives at the destination successfully with no errors, and no reply (if applicable) or acknowledgment of the message is required. This notion is very similar to the Internet's usage of the User Data Protocol (UDP).

The two-way mode of transmission of a message provides error recovery through retransmission, since messages sent in this mode must be acknowledged. Acknowledgment generally comes in the form of an acknowledgment message (although reply messages can imply acknowledgment as well). If a message is transmitted and acknowledgment of it is not received within an expected period of time, a retransmission scheme is responsible for retransmitting the message and ensuring correctness.

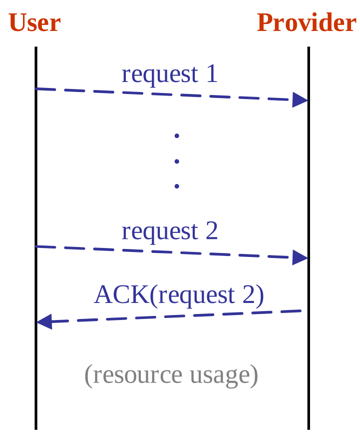


Figure 3: Request 1 is transmitted by the User in the one-way mode. Request 2, however, is transmitted in the two-way mode and requires an acknowledgment confirmation.

Architectural Assumptions

DAP is expected to exist within a well-defined architectural environment. The details, designs, and implementations regarding such an environment are outside the scope of DAP's definition; however, the required environment is important to discuss.

- An application must be present at the User entity that is capable of defining and initiating resource reservation requests. Without an application, DAP would have no driver.

- Each Provider entity must have a resource manager that can communicate with DAP. The resource manager is expected to manage all resources available at the Provider entity, and thus supply feedback to DAP about the acceptance or rejection of requests.

- An underlying communications system must provide the means with which DAP messages are transmitted. No assumptions are made concerning the communications system; however, DAP is not intended to compensate for systems where high levels of errors and data loss are common.

3. PROTOCOL DESIGN

DAP contains two layers as shown in Figure 4. The translation layer is responsible for converting between entity-specific resource definitions and DAP resource definitions. It serves the essential function of allowing different applications and resource managers to adapt to a common messaging system. The second layer within DAP is the messaging layer, which is responsible for creating, transmitting, and receiving DAP messages.

DAP's intent is to specify an overall messaging system used to negotiate resource reservations. It does not specify or restrict what resources may be negotiated or how resources are locally defined. These matters are left entirely to the application and resource manager, and are communicated to DAP via the adaptable translation layer.

Additional DAP Details

A typical DAP session begins with a User sending a resource request to a Provider. The priority level specified in the request can be used by the Provider as a measure of importance. The included persistence time informs the Provider of the duration with which the request is valid. For example, if a request is received, but is not accepted by the Provider and communicated to the User before the persistence time, it becomes invalid and no further action is necessary.

Upon receiving a resource request, a Provider may accept or reject it. This decision is based entirely upon the workings of the resource manager.

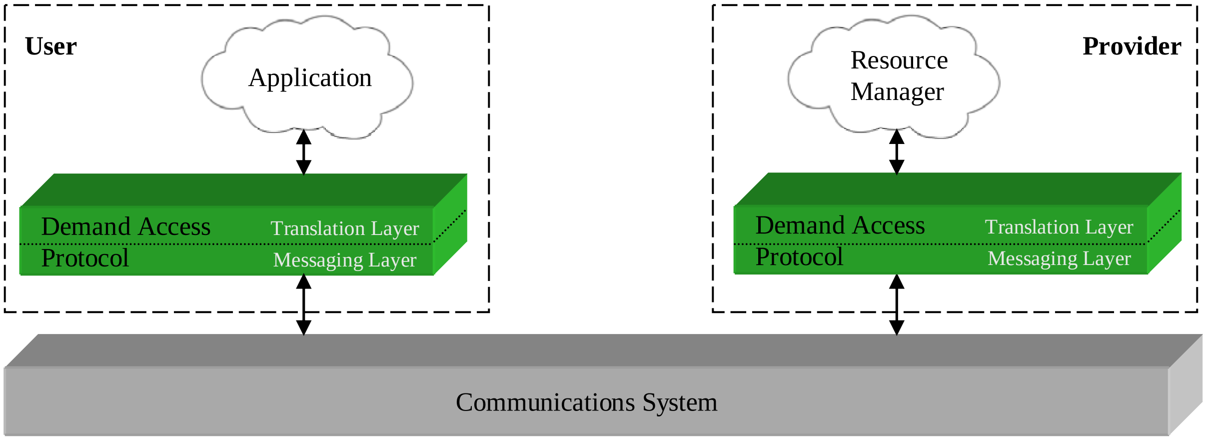


Figure 4: To support DAP, an application, resource manager, and communications system must be present.

The Provider may modify the parameters of any existing reservation at any time. If an existing reservation is modified, then a command is sent to the affected User to notify it of the new change. This is one way in which DAP provides for a demand-driven network paradigm.

In addition to commands that are issued by Providers to modify reservations, Users may also modify their own existing reservations. This is performed by sending special requests that indicate an override. Override requests give Users the capability to dynamically change the resource parameters of their own existing reservations as they see fit.

As mentioned earlier, messages can be sent in one-way or two-way modes. If a link is expected to contain errors, a message sent in two-way mode will guarantee its eventual error-free arrival. If a message must be sent urgently and no time is available to wait for an acknowledgment, or if the link is known to be error-free, then the one-way mode can be used. Both modes are available to accommodate unpredictable situations and offer another level flexibility.

Example Usage of DAP

An operational DAP scenario can be seen in Figure 5, where two independent rovers (i.e. Users A and B) on the surface of Mars wish to negotiate resources with a single orbiter (i.e. Provider). A corresponding ladder diagram is shown in Figure 6 that illustrates how both rovers may use DAP to interact with the orbiter.

In an example scenario, Rover A begins session A1 by sending a new request to Orbiter to set up a future communication link at 6Mbps. The request is sent reliably, so Rover A expects some form of acknowledgment. Orbiter is not yet ready to accept or reject the request, so a simple acknowledgment message is returned. Since only an acknowledgment message, and not a reply message, is received by Rover A, the request remains outstanding up until the persistence time. At the persistence time, Rover A disregards its initial request, as does Obiter.

In the meantime, Rover B issues its own request to Orbiter to reserve 20GB of disk space for a future pass in session B1. Orbiter's disk capacity is only 10GB, so it replies to Rover B with a rejection message.

Rover A decides to begin a new session and requests a future communication link at 4Mbps. It detects that the message was lost since it did not receive an acknowledgment or reply message, so it retransmits the request. Orbiter accepts the new reservation and replies with an acceptance.

At a later time, Rover B begins session B2 by sending a request message to Orbiter. . The current transmission link is known to be error-free, so the rover is confident that that Orbiter will receive its message, which is the reason why it was sent in one-way mode. The message also has high priority and indicates a reservation for a communication link at 4Mbps and 5GB of disk space. In order for Orbiter to satisfy and accept Rover B's request, it modifies Rover A's existing reservation and downgrades its level of service to only 1Mbps. This modification is communicated to Rover A via a command message sent in two -way mode. Rover A has no choice but to comply with the command and acknowledge it.

Due to unexpected hardware failures, Rover B loses much of the data that it initially wanted to downlink. Instead of having 5GB of data to transmit back to Earth, it now only has 1GB of data. As a result, it overrides its own reservation and requests only 1GB of disk space from Orbiter, which is immediately accepted.

After all of the negotiation processes, Rover A has a reservation for a 1Mbps link, while Rover B has a reservation for a 4Mbps link and 1GB of disk space. When the resource usage periods begin for Rover A and Rover B, Orbiter is prepared to honor the reservations. If either of the rovers chose to do so, additional requests could be made to further modify the existing reservations or make new ones. In the case of Figure 6, the sessions A2 and B2 end when their respective resource usage periods terminate.

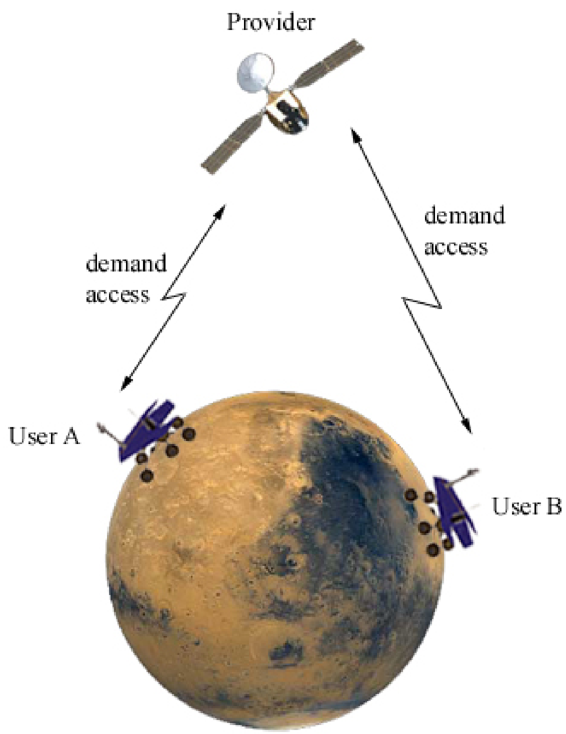


Figure 5: An example DAP scenario depicting two rovers (i.e. Users) competing for resources from one orbiter (i.e. Provider) around Mars.

4. Conclusions

This paper describes the motivation and design of a demand access protocol. The protocol provides both flexibility and resource efficiency for future space-based networks whereby autonomous and dynamic event-driven operation is the new paradigm. DAP provides the messaging formats and procedures for negotiating resource allocations for rapid response to demands arising from contingency, as well as advanced reservation of resources for future task planning.

To make the vision of a demand-driven communication paradigm a reality, each mission must believe in the need for having such flexibility in rescheduling communications. Depending on the science and exploration objectives and the baseline concept of operation, demand-driven communication may first appear at best an “enhancing” rather than an “enabling” technology. But the recent focus on lunar exploration has introduced the possibility of deploying a network in space that is so complex that, in order to support long term human presence, there is no choice but to have such level of flexibility. Ongoing efforts are underway to provide as much bandwidth as possible to human missions due to public interest in video footage, as well as the need for more spontaneous and interactive communications services. But as the bandwidth need approaches link capacity, demand-driven communication will play a key role in further optimizing the efficiency and the responsiveness of the network. While missions may not yet see a need for it, this technology will inevitably bring new options for operating future missions.

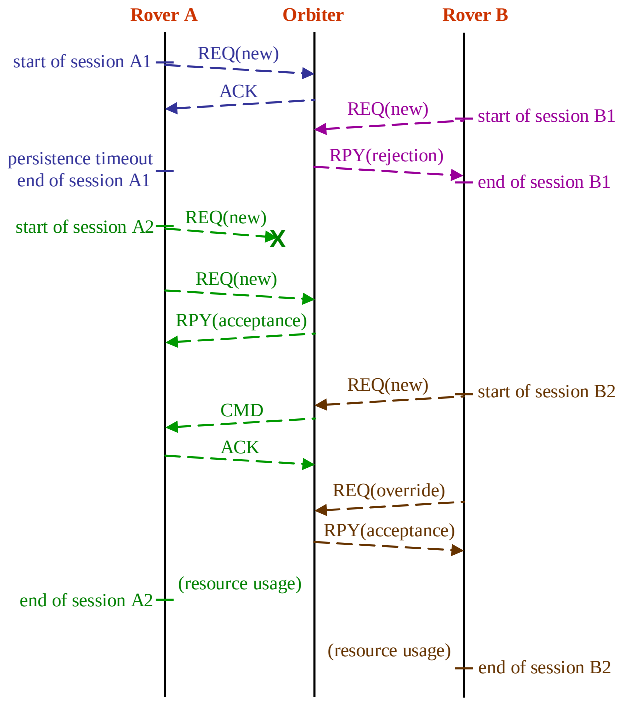


Figure 6: A ladder diagram showing the usage of DAP to negotiate resources between two rovers and an orbiter. Refer to the section entitled *Example Usage of DAP* for more details.

Additional SPIN [5] and simulation work is being conducted to model, verify, and test the DAP design.

References

[1] D. S. Bagri, J. I. Statman, and M. S. Gatti. *Operation's Concept for Array-based Deep Space Network.* IEEE Aerospace Conference, pp. 1532–1540, March 2005.

[2] R. Gladden, P. Hwang, B. Waggoner, B. McLaughlin, P. Fieseler, R. Thomas, M. Bigwood, and P. Herrera. *Mars Relay Coordination Lessons Learned.* IEEE Aerospace Conference, pp. 177–190, March 2005.

[3] M. Rayman, P. Varghese, D. Lehman, L. Livesay. *Results from the Deep Space 1 Technology Validation Mission.* 50th International Astronautical Congress, Amsterdam, The Netherlands, October 1999, IAA-99-IAA.11.2.01.

[4] S. Franklina, J. Slonski, S. Kerridgea, G. Noree, J. Riedela, D. Stosica, C. Racho, B. Edwardsb R. Austina, and D. Borosonc. *The 2009 Mars Telecom Orbiter Mission.* IEEE Aerospace Conference, pp.465–480, March 2004.

[5] Holzmann, G. J. *The SPIN Model Checker: Primer and Reference Manual.* Addison-Wesley, 2004. ISBN 0-321-22862-6.

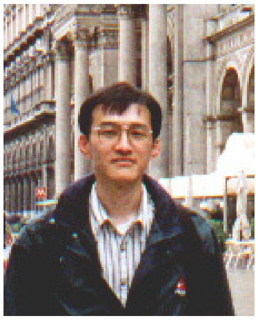
[6] L. C. Wolf and R. Steinmetz. *Concepts for Resource Reservation in Advance.* Special Issue of Journal of Multimedia Tools and Applications, The State of the Art in Multimedia Computing, May 1997.

[7] D. G. A Mobach, B. J. Overeinder and F. M. T. Brazier. *A Resource Negotiation Infrastructure for Self-Managing Applications.* Second International Conference on Autonomic Computing, pp. 381–382, June 2005.

CONTRACTUAL ACKNOWLEDGEMENT

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.

BIOGRAPHIES



Jay L. Gao is a research staff member of the Communication Architecture and Research Section at the Jet Propulsion Laboratory in Pasadena, CA. He is currently working on protocol development and performance evaluation for in-situ surface-to-surface and surface-to-orbit communications in planetary missions, and sensor network research and modeling. He received his Ph.D. in Electrical Engineering from the University of California at Los Angeles in 2000.

Dee Leang is a software engineer within the Flight Software and Data Systems Section at the Jet Propulsion Laboratory in Pasadena, CA. He assisted with the development, implementation, and testing of the CCSDS File Delivery Protocol (CFDP), and is currently a member of the Multi-Mission System Architecture Platform project developing flight software. He received his M.S. in Information Networking from Carnegie Mellon University in 2004.