Autonomous Data Transfer Operations for Missions

Max Repaci, Paul Baker, and Fred Brosi
Global Science and Technology, Inc.
Greenbelt, MD, USA

Automating the data transfer operation can significantly reduce the cost of moving data from a spacecraft to a location on Earth. Automated data transfer methods have been developed for the terrestrial Internet. However, they often do not apply to the space environment, since in general they are based on assumptions about connectivity that are true on the Internet but not on space links. Automated file transfer protocols have been developed for use over space links that transfer data via store-and-forward of files or segments of files. This paper investigates some of the operational concepts made possible by these protocols.

The current trend in space missions is away from a few large expensive missions and toward many small low-cost missions. If the overall budget is not increased, this trend necessarily implies a lower cost per mission. We argue that a portion of the necessary cost reductions can be realized through increased standardization and automation. Standardization of communications protocols and operations can allow the development, maintenance, and even operational costs of infrastructure and software to be shared among many missions. Automation can eliminate a significant component of the operations budget.

In the past, the mission budgets were larger and there were fewer missions. Those mission components whose costs would be significantly reduced by these methods were a relatively smaller fraction of the total mission budget, and there were fewer missions over which to share costs. Therefore missions made heavy use of custom protocols and software, and manual operations were common. Now that the cost of the other mission components has been reduced, for example, by flying smaller spacecraft, missions have a strong incentive to apply these methods.

We will investigate operational scenarios possible with an automated data transfer method. We will consider operations to support routine data transfers, since those most readily lend themselves to automation. Examples of these transfers include that of command loads to the spacecraft or of stored telemetry data from the spacecraft. Automating routine transfers can reduce operations costs; development costs can be reduced as well by standardizing the software, interfaces, and protocols used for data transfer.

I. Methods for Reducing Mission costs.

Automation of operations and standardization of communications protocols and operational methods can allow significant savings in both the development and operational phases of a mission. Standards allow the reuse of program code and procedures, both directly and indirectly by using protocol layering. Protocol layering refers to the process of dividing the protocols into layers that handle certain functionality. Standard interfaces are provided between the layers, so that layers which are optimized for a particular environment can be substituted while still reusing the other parts. Standards allow interoperability between missions, which allow not only sharing of development costs, but also in operations, as infrastructure can be shared among missions.

Due to the availability of random-access storage on the spacecraft, it is now possible to automatically retransmit data lost on the link, thus guaranteeing the delivery of telemetry data in a complete and uncorrupted form. The use of a reliable transfer protocol, that is, a protocol that delivers contiguous, error-free data, can in itself provide cost savings by allowing more efficient use of the link time. In the past, the error rates on the link were monitored until the link quality reached a specified level, and then the data was downloaded. If the data contained more than the acceptable number of errors, then the entire transfer had to be repeated. With a reliable protocol, the transfer can proceed automatically as soon as the connection is established, even if the error rate is relatively high, and there are guaranteed to be no errors in the transferred data. Thus, data can be transferred under link conditions that would not have been possible if a reliable protocol were not being used. Also, allowing uncorrected errors to remain in the transferred data can impede the automation of the analysis of the data, and may require potentially expensive corrective measures to be taken by the end user. We will discuss how standardized data transfer protocols based on automated file transfers can be used to reduce costs of mission operations.

II. Mission operations via file transfers.

There are certain requirements implicit in operating missions using file transfer operations. Naturally, the spacecraft must have at least a rudimentary file system on board. The status quo of operations, for example, downloading data, and uploading of data or commands, can be supported by operations with file transfers. However, certain extensions to current operational modes will not be possible. For example, heavily interactive operations cannot be supported efficiently due to overhead considerations.

It may not be possible to establish a reliable connection all the way from one end of the data transfer path to the other. The contact with a spacecraft is very often intermittent, and communications may be established to the spacecraft via different groundstations on different contacts. Therefore, the data transfer methods considered here are based on a store and forward model, a similar concept to sending email on the Internet. The store and forward method allows automation of routine data transfers to proceed without concern for the contact schedule of the spacecraft.

III. Operations with Internet protocols.

Certainly the reliable, automatic transfer of data occurs all the time on the terrestrial Internet. However, solutions developed for the Internet may not be easy to apply to the space environment. Internet file transfer applications generally require an underlying end-to-end reliable transport layer protocol, and therefore cannot be directly applied to data transfer over space links, since in general a reliable end-to-end transport layer protocol will not be available. Assuming that the spacecraft can be accessed via an internet[1], we will now describe how a mission operations concept can be built on the Internet protocols. Most of the application layer Internet protocols are based on an interactive mode of operation, but in order to achieve lower operation costs the routine operations must be automated. Furthermore, the applications designed for data transfer on the Internet generally assume that a TCP connection can be established directly between the end points, which will not be the case much of the time for most missions. A solution is to design an application that transfers data automatically and is built on the existing lower layers of the Internet protocols. A "messaging layer" can be built in to allow sharing of store and forward code. We will describe one such application based on the SAFE protocol[2].

SAFE is an application layer protocol that is designed to transfer data automatically to or from a spacecraft that can only be contacted intermittently. SAFE was targeted to automatically handle the routine transfer of data to or from a spacecraft. SAFE has a client/server architecture: The configuration for a one-way data transfer consists of a data server running on the node containing the data source and a client running on the node to which the data is to be sent. There must also be proxy servers and clients on all nodes that perform store and forward, but they are transparent to the endpoints. The transfer operation is performed via requests for data that are sent from the client. These requests contain a prearranged identifier for the file, the offset of the start of the request in the file, and the length of data requested. When the server receives a request, it responds with the data requested, or an error condition if the data is not available. When the client receives the data, it makes it available to user applications. If the data does not arrive in a reasonable time or is not complete, the client requests the missing data again. Thus, reliability is maintained through a data transfer loop closed at the application layer. The transport layer is assumed to only provide a stream service, i.e., ordering of the data, and notification of lost data. SAFE does not require that the transport layer provide reliable delivery in order to achieve end to end reliability, since reliability is managed at the application layer. This method of operation can also be converted into a 'push' mode by having the client initially request a very large amount of data. The server will then break the replies up into smaller pieces of a specified length, and transfer data as it becomes available.

One possible operations scenario would use a SAFE application to transfer stored telemetry data that is being continuously written into files on the spacecraft. Since the core procedures of SAFE do not include a file discovery method, a mapping would have to be provided between the names of the files and the file handle that SAFE uses to identify them. Then the client would be directed to transfer the files cyclically from the prearranged set of file handles. The server and client would use the mapping to name the files given the file handle. Alternatively, since SAFE is extensible, a method could be added to associate the file handle with a file in a standard way. SAFE could also aid in the managing of the file store on the spacecraft: The client could be directed to acknowledge receipt of data, and files that are known to have been successfully copied could be deleted.

SAFE can also be used to provide a reliable store and forward stream service using a standard extension to SAFE called the Recyclable File System (RFS). An application on the node with the data source maintains the RFS, which is a series of rotating buffers into which data can be written. The data is then automatically transferred to an equivalent set of buffers on the destination node. The application also keeps track of which data has been transferred, so that applications can access the data as soon as it arrives, even if it hasn't yet been contiguously transferred. An application that will use the data can read it from the remote RFS, either as a stream or by random access. The reliable stream service could be used, for example, to upload command loads to the spacecraft.

IV. Operations with CFDP.

CFDP (CCSDS File Delivery Protocol) is being standardized by the CCSDS (Consultative Committee for Space Data Systems), an international standards organization for space communications protocols. More information can be found on the CFDP protocol in the relevant CCSDS documents, referenced at the end of this document. We will give a brief summary of the architecture of CFDP and describe how it could be used as part of an automated data transfer mechanism^3.

CFDP is a store and forward messaging application designed for transferring files over space links and maintaining a remote file system on a spacecraft. The messages can contain file data, directives for remote management of the file store, user message data, or any combination thereof. Messages are passed between CFDP entities, one on the node containing each file store, which handle the processing of file management directives and writing of the file passed with the message, if there was one. File management directives include such things as file deletion, file or directory creation, or renaming files. The behavior the entities and the format of the messages are specified by CFDP. Each entity is controlled by a user application that directs it to send messages. The process of sending a message in the CFDP nomenclature is called a 'transaction'. The user application can direct a CFDP entity to start a transaction (the put directive), cancel, suspend, or resume a transaction, or report on the status of a transaction in process. For example, to transfer a file from the spacecraft to the ground, the user on the spacecraft would start a 'put' transaction, that is, send a message. The message would include the file to be transferred. The remote entity would receive the message and automatically write the file to the local file system. Transactions may also include user messages. The user messages can be used to add functionality to the core procedures of CFDP. The format of the user messages is not specified, except for that of two reserved messages. These reserved user messages represent standard extensions to CFDP. One of them allows a remote user to get directory listings of the local file store and the other initiates proxy transactions. Proxy transactions allow a user to direct a remote CFDP entity to initiate a transaction, for example, to send a file. Without the proxy user message, CFDP would not have any standard way to get a file from a remote entity.

CFDP can easily be configured to perform automated data transfer. When an instrument writes a data file on the source node file system, the local user application directs the resident CFDP entity to perform a Put operation to transfer the file to the ground entity. When the transaction is finished, the entity at the destination notifies the spacecraft entity that the file has been delivered. The spacecraft entity then notifies its user, which can then delete the file to free resources. The reverse procedure could be used to transfer command loads to the spacecraft. This scenario requires two components that are not part of the CFDP standard: user applications on the spacecraft and on the ground node. These applications would have to be developed by the implementers of this automated file transfer operation. The user application on the spacecraft directs its local CFDP entity to transfer any new files and handles the delivery of any user messages. It may also delete the data after it has been transferred successfully, or its peer application on the ground could issue the delete remotely. The user application on the ground uploads commands and other data, handles user messages, and provides an interface for remote management of the data store on the spacecraft.

V. Discussion

Although there is considerable overlap in functionality between CFDP and the Internet operations model presented above, the functionality offered by CFDP is a small subset of that possible through use of the Internet protocols. Therefore, the choice of which protocol to use depends on the requirements of the mission. The design of CFDP is focused on file delivery and file management for missions configured similarly to the way they are commonly done now. If the requirements of a mission can be satisfied by that and possibly some slight extension in functionality, then CFDP will be adequate. CFDP is fixed on supporting the status quo of mission operàtions, and perhaps short-term extensions to it, albeit with increased automation of operations. This focus should allow it to perform well under its targeted operations scenarios. Conversely, however, it makes it very difficult to extend CFDP to support a significantly different scenario. Therefore, if substantial additional functionality were required, the Internet operations model would probably provide it at a lower cost.

VI. References

CCSDS File Delivery Protocol (CFDP). Draft Recommendation for Space Data System Standards, CCSDS 720.1-G-0.5. Green Book. Issue 0.5. Washington, D.C.: CCSDS, July 1999.

CCSDS File Delivery Protocol (CFDP). Draft Recommendation for Space Data System Standards, CCSDS 727.0-R-3. Red Book. Issue 3. Washington, D.C.: CCSDS, July 1999.

[1] The Internet protocols are optimized for use under the conditions typically found on the terrestrial Internet. The space environment has many characteristics that can cause the Internet protocols to operate in a non-optimal way or to fail outright. Much effort has been made to adapt them to the space environment, including extensions and modifications to the Internet protocols standardized by the CCSDS. More information can be found on the SCPS homepage at <http://www.scps.org>.

[2] More information about SAFE can be found on the SAFE website: <http://safe.gst.com>.

[3] CFDP is not entirely consistent with the Internet operations model, since it includes its own transport layer protocol, but it can be operated over UDP.