machine. The component provides all service interfaces via a Representational State Transfer (REST) protocol over "https" to external clients. There are two general interaction modes with the service: upload and download of data. For data upload, the service must execute logic specific to the upload data type and trigger any applicable calculations including pass delivery latencies and overflight conflicts. For data download, the software must retrieve and correlate requested information and deliver to the requesting client.

The provision of this service enables several key advancements over legacy processes and systems. For one, this service represents the first time that end-to-end relay information is correlated into a single shared repository. The software also provides the first multimission latency calculator; previous latency calculations had been performed on a mission-by-mission basis.

This work was done by Daniel A. Allard, Roy E. Gladden, Jesse J. Wright, Franklin H. Hy, Gregg R. Rabideau, and Michael N. Wallick of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47454.

Interplanetary Overlay Network Bundle Protocol Implementation

NASA's Jet Propulsion Laboratory, Pasadena, California

The Interplanetary Overlay Network (ION) system's BP package, an implementation of the Delay-Tolerant Networking (DTN) Bundle Protocol (BP) and supporting services, has been specifically designed to be suitable for use on deep-space robotic vehicles. Although the ION BP implementation is unique in its use of zero-copy objects for high performance, and in its use of resource-sensitive rate control, it is fully interoperable with other implementations of the BP specification (Internet RFC 5050).

The ION BP implementation is built using the same software infrastructure that underlies the implementation of the CCSDS (Consultative Committee for Space Data Systems) File Delivery Protocol (CFDP) built into the flight software of Deep Impact. It is designed to minimize resource consumption, while maximizing operational robustness. For example, no dynamic allocation of system memory is required. Like all the other ION packages, ION's BP implementation is designed to port readily between Linux and Solaris (for easy development and for ground system operations) and VxWorks (for flight systems operations). The exact same source code is exercised in both environments.

Initially included in the ION BP implementations are the following: libraries of functions used in constructing bundle forwarders and convergence-layer (CL) input and output adapters; a simple prototype bundle forwarder and associated CL adapters designed to run over an IP-based local area network; administrative tools for managing a simple DTN infrastructure built from these components; a background daemon process that silently destroys bundles whose time-to-live intervals have expired; a library of functions exposed to applications, enabling them to issue and receive data encapsulated in DTN bundles; and some simple applications that can be used for system checkout and benchmarking.

This work was done by Scott C. Burleigh of ARCO for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-41628.

STRS SpaceWire FPGA Module

NASA's Jet Propulsion Laboratory, Pasadena, California

An FPGA module leverages the previous work from Goddard Space Flight Center (GSFC) relating to NASA's Space Telecommunications Radio System (STRS) project. The STRS SpaceWire FPGA Module is written in the Verilog Register Transfer Level (RTL) language, and it encapsulates an unmodified GSFC core (which is written in VHDL). The module has the necessary inputs/outputs (I/Os) and parameters to integrate seamlessly with the SPARC I/O FPGA Interface module (also developed for the STRS operating environment, OE).

Software running on the SPARC processor can access the configuration and status registers within the SpaceWire module. This allows software to control and monitor the SpaceWire functions, but it is also used to give software direct access to what is transmitted and received through the link. SpaceWire data characters can be sent/received through the software interface, as well as through the dedicated interface on the GSFC core. Similarly, SpaceWire time codes can be sent/received through the software interface or through a dedicated interface on the core.

This innovation is designed for plug-and-play integration in the STRS OE. The SpaceWire module simplifies the interfaces to the GSFC core, and synchronizes all I/O to a single clock. An interrupt output (with optional masking) identifies time-sensitive events within the module. Test modes were added to allow internal loopback of the SpaceWire link and internal loopback of the client-side data interface.

This work was done by James P. Lux, Gregory H. Taylor, Minh Lang, and Ryan A. Stern of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47434.