

SmartSSR DTN Router

David Edell
Matthew McKay

APL

The Johns Hopkins University
APPLIED PHYSICS LABORATORY

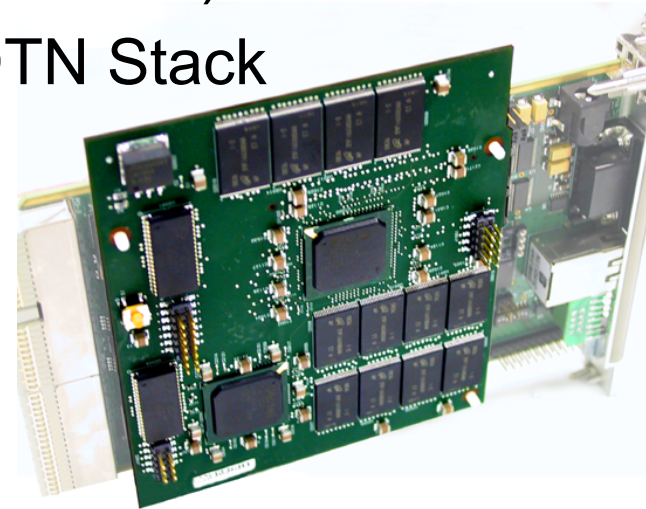


Overview

- **What is the SmartSSR?**
- **Overview of Delay Tolerant Networking (DTN)**
- **DTN on RTEMS**
- **APL Flash File System**
- **RTEMS Integration Issues**
- **SmartSSR DTN Router**

SmartSSR

- The SmartSSR DTN Router is an integrated device:
 - UT699 Leon3 Processor
 - NAND Flash Daughter Card
 - APL Flash Filesystem (an extension of YAFFS)
 - Interplanetary Overlay Network (ION) DTN Stack
 - RTEMS Operating System
 - SpaceWire Communication Bus
- **What's New?**
 - DTN
 - RTEMS: Previous prototyping used VxWorks 6.5
 - CFDP will operate over DTN



What is DTN?

- **DTN is a set of protocols that collectively enable delay or disruption tolerant networking.**
 - Bundle Protocol (BP)
 - Linklayer Transport Protocol (LTP)
 - Datagram Retransmission (DGR)
- **Standards-based protocol stack**
 - RFC 4838 defines the core protocols
 - CCSDS Green Book, 734.0-G-1 defines recommended architecture for space deployments.
- **JPL's Interplanetary Overlay Network (ION)**
 - DTN Reference Implementation
 - Open Source application suite available through OhioU (<http://ion.osp.ohiou.edu>)

Current spacecraft communications

Existing Standard is Point to Point communication

- **Traditional method of communicating data is using point to point links**
 - An orbiter to ground station
- **Disadvantages:**
 - One-way trip light times.
 - Earth to Moon: 1.7 seconds
 - Earth to Mars: order of minutes to tens of minutes.
 - High bit error rate due to solar radiation
 - Celestial bodies are in constant motion
 - Blocks line-of-sight
 - Results in intermittent connection
- **Complicated Planning and execution**

Delay Tolerant Network

A new model for spacecraft communication

- **Goal: Automate many of the logistical challenges**

- *Store-and-forward* techniques can be used to compensate for intermittent link connectivity and high latency.

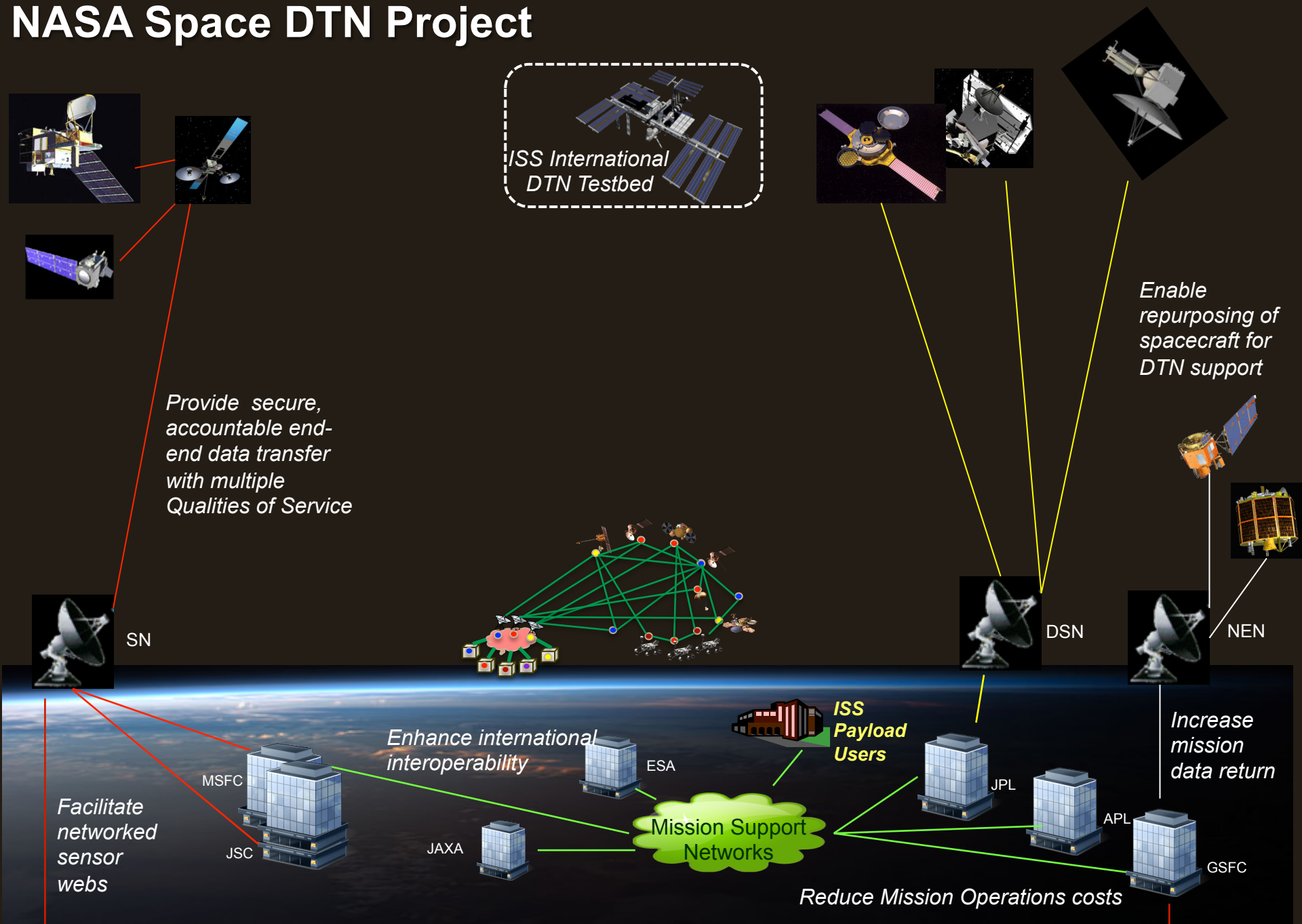
- **Idea based on real world successes**

- Mars Rovers and Mars Odyssey
 - Data from Mars Rovers relayed through Odyssey
 - Faster to upload data to the Orbiter then back to earth rather than directly

- **NASA DTN Project**

- An inter-agency project to advance DTN-related technologies
- Network testing between NASA Centers, APL, and ISS
- Funded integration of DTN into the SmartSSR concept

NASA Space DTN Project



Interplanetary Overlay Network (ION)

- **Reference implementation of DTN**

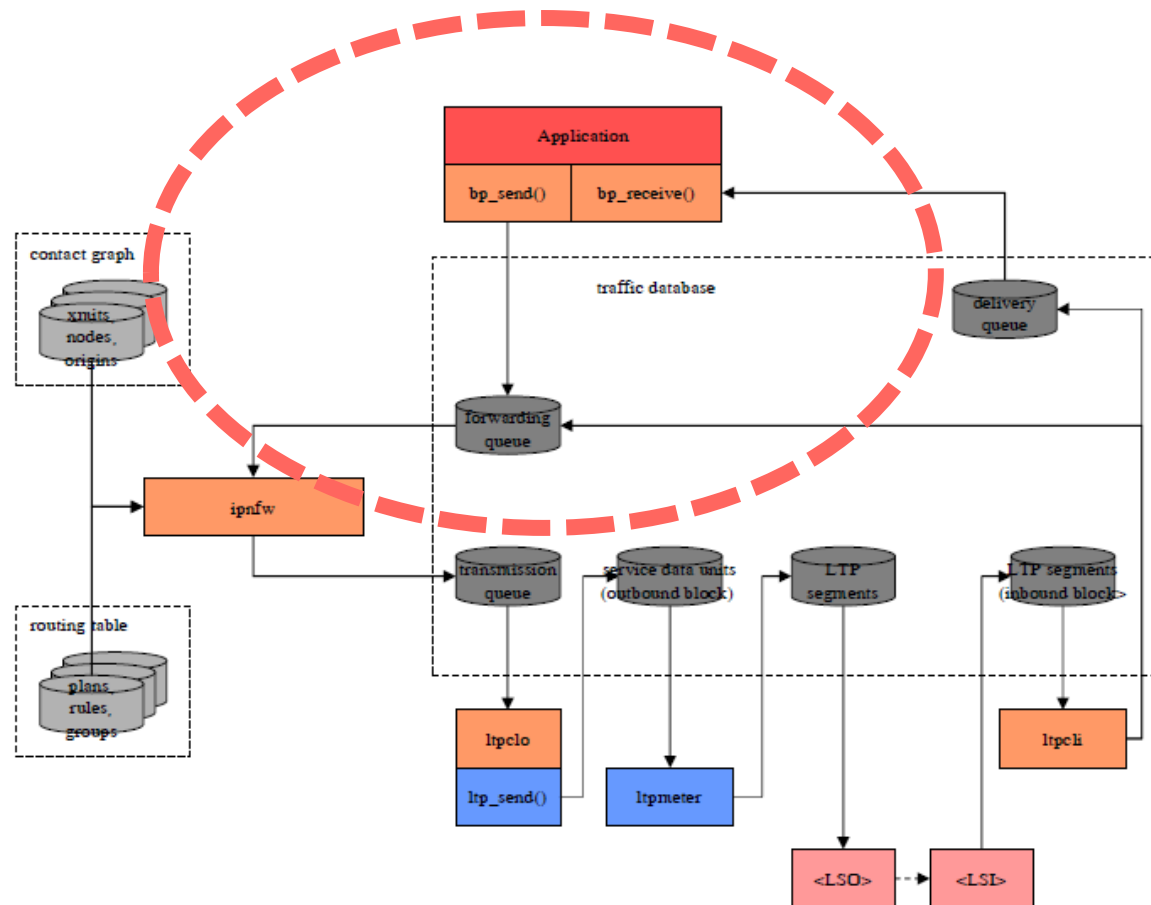
- Created by JPL
- Open source version available at <http://ion-dtn.sourceforge.net/>
- NASA-DTN funding continued development

- **ION Includes**

- Implementation of the DTN Network Stack
- Test applications for sending and receiving data
 - Includes CFDP over DTN
- A Spacecraft Data Recorder (SDR) Library for temporarily storing packets.

ION SDR

- SDR provides for automatic storage and retrieval of packets in support of store-and-forward functionality.



SmartSSR DTN Router

- **SmartSSR handles all DTN traffic**

- DTN packets addressed for the spacecraft can be saved as a file, or forwarded to the main processor for handling
- The SDR will temporarily store other packets on the SSR until the next opportunity to forward to its destination.

- **SDR Limitations**

- Optimized for traditional disks or ram disks
 - Allocates a large file and treats it as raw memory
 - This implies all SDR operations require editing an existing file.
- Creation and deletion of separate files is preferred for a Flash file system.
 - Upgrade of SDR is an open task

APL Flash File System

■ **An Extension of YAFFS**

- An open source file system designed for NAND Flash Devices
 - Write-Once, Read-Many, Erase by Block
 - Optimum performance in writing or appending files
 - File edits supported, but inefficient
- Used in the Android operating system
- Logged File System

■ **Existing APL Extensions (formerly YAFFS3)**

- Eliminated dynamic memory allocation
- VxWorks Integration
- Enhanced support for large disks by allowing multiple pages per chunk

APL Flash File System on RTEMS

- **Why RTEMS?**

- Target platform is Aeroflex Gaisler LEON3
- RTEMS is given priority over VxWorks by vendor

- **YAFFS File System is OS-independent**

- A Direct interface is available for standalone-usage
- Source is provided for integration into a Linux kernel
- Calls to the direct interface are inserted into the appropriate file system driver template for the target operating system.

- **VxWorks vs RTEMS Integration**

- OS Integration requires the definition of several data structures and/or kernel registration functions on startup
 - Definitions vary between OS, but the concepts are equivalent.

RTEMS Port Lessons Learned

▪ Build Environment

- RTEMS development started with Eclipse on a Windows system
- Kernel Configuration Issues appeared on transition to Linux
 - A kernel configuration option related to networking changed between RTEMS releases
 - Version differences were not clear between the Gaisler Windows Installer package and the Linux RTEMS image.

▪ APL SSR Driver Portability

- A memcpy is used to copy data into a hardware FIFO
- On VxWorks, memcpy moves bytes in from end to start
- RTEMS copies from start to end
- To allow cross-OS testing on the recorder, VxWorks version was modified to use a non-optimized for-loop assignment.

RTEMS Port Lessons Learned: DTN

▪ ION Build Settings for RTEMS

- Includes RTEMS Build Settings were non-functional
- Existing compiler directive used to set component entry points as `main()` or `app_name()` function
 - The latter is used for RTEMS and VxWorks
 - These directives were not applied to all libraries and test functions.
 - Remaining applications explicitly updated with this change, or excluded from the RTEMS build

▪ LTP Compilation Issue

- Default build environment enabled `-O2` optimizations flag
- Removing `-O2` flag eliminated LTP compilation error.



Questions?