TELEMETRY AND COMMAND FRAME ROUTING IN A MULTI-MISSION ENVIRONMENT

Manfred Bester
Bester Tracking Systems, Inc.
Suisun City, CA 94585, USA
bester@bester.com

Brett Stroozas
Stroozas Flight Ops
Walnut Creek, CA 94598, USA
bretts@stroozas.com

ABSTRACT

In a modern ground control network for space communications, secure peer-to-peer TCP/IP network socket connections are typically used to transfer real-time telemetry and command frames between satellite operations centers and remote ground stations. Reliable and timely reconfiguration of data paths for upcoming pass supports becomes rather complex when many spacecraft and ground stations are involved. This paper describes a routing software application that was developed to facilitate switching of telemetry and command data paths between multiple ground stations and spacecraft command and control systems, and to forward telemetry streams to multiple client applications in parallel. Fully automated configuration and monitoring of the data flows is accomplished via a remote control interface that is tied into a pass scheduling system. The software is part of the SatTrack Suite and currently supports multi-mission flight operations, including those of the recently launched THEMIS constellation mission at Space Sciences Laboratory, University of California at Berkeley.

KEY WORDS

Satellite Control Networks, Telemetry and Command Data Flows, Peer-to-Peer TCP/IP Network Socket Connections, Frame Routing, Control Center Automation

INTRODUCTION

The Mission Operations Center (MOC) at Space Sciences Laboratory (SSL), University of California at Berkeley (UCB) was established in 1999 to support the NASA funded FAST and RHESSI Small Explorer Missions. A third mission, CHIPS, was added in 2003.[1] While CHIPS employs the relatively new Internet-in-space technology for two-way communications, both

FAST and RHESSI utilize CCSDS standard packetization with Channel Access Data Units (CADUs) on the telemetry baseband channel and Command Link Transmission Units (CLTUs) with embedded Telecommand (TC) codeblocks on the command channel.[2] Connectivity for transfer of telemetry and command data between the satellite command and control workstations at the MOC and the frontend processors at the remote ground stations is typically established via secure peer-to-peer Transmission Control Protocol / Internet Protocol (TCP/IP) network socket connections. One side acts as a network client initiating the connections, while the other side acts as a network server, receiving and accepting the connections. Once the connections for telemetry and command transfer are established, data flows are activated and two-way communications between the MOC and the spacecraft take place for the duration of a pass support. Several layers of firewalls are often used to control traffic between secure network segments to prevent that ground or space assets are compromised.

When THEMIS – a NASA Medium Explorer constellation mission consisting of five small satellites to study magnetospheric events leading to the aurora – was funded in 2003, the MOC at UCB/SSL needed to be expanded to accommodate the additional infrastructure required for satellite command and control operations.[3] It was obvious that a new architecture needed to be implemented to manage network connectivity and data flows with a high degree of flexibility, reliability and systems automation. The core of the new network topology consists of two software applications, FrameRouter and FrameRelay, providing the enhanced capabilities. Both programs are remote controlled via a central real-time scheduling system, the Gateway Server. FrameRouter, FrameRelay and Gateway Server are tools in the SatTrack Suite.[4]

FRAME ROUTING

FrameRouter can be visualized as the TCP/IP equivalent of a baseband matrix switch, routing data from one source to multiple destinations, and is realized as a multi-threaded software application. A main thread performs all remote interface and data flow control functions. In addition, so-called routing threads are started or terminated upon remote request by the Gateway Server. There can be up to 100 routing threads running simultaneously. When a routing thread is started, it activates two server (listen) sockets, each associated with a *route point* (port). One route point is designated as the data *source* and the other as the data *destination*. The connection between two route points is called a *route*. Only one client can connect to the source port, but up to 100 clients can simultaneously connect to the destination port. Data received on the source port are forwarded to any client connected to the destination port. Status information on each active routing thread, such as Internet Protocol (IP) addresses of connected clients, numbers of bytes received and forwarded, and error messages are sent to the Gateway Server for remote monitoring, logging and operator paging, in case an error condition occurs.

FrameRelay is a multi-threaded application also and works similar to FrameRouter, except that it provides network connectivity with the opposite gender, acting as a network client instead of a network server. Upon remote request from the Gateway Server, FrameRelay starts a relay thread that opens two client connections – a source and a destination – to two servers on specified ports. Any data received from the source connection are relayed to the destination. There can be up to 100 relay threads running simultaneously. FrameRelay will automatically attempt to reconnect if one of its client connections drops out. One of the purposes of FrameRelay is to link different instances of FrameRouter, thereby allowing creation of a larger data distribution network. This capability is used for relaying telemetry data from the secure, isolated operations network inside the MOC to the less secure SSL Local Area Network (LAN), as described further below.

SYSTEM CONFIGURATION

All FrameRouter and FrameRelay configuration is performed via a plain text file that is read by the Gateway Server. In this configuration file, each route point is associated with a port number for a particular instance of FrameRouter, and is aliased to a textual mnemonic that can subsequently be used to submit *make* or *break* route requests. In addition, an individual IP address or range of IP addresses can be specified to restrict access to any route point. Routes or relays can only be established between defined route points. The selection of port numbers is completely user configurable. Certain blocks of port numbers may be reserved for the remote ground stations to connect to, whereas other blocks may be assigned to spacecraft control systems in the MOC. An example of the assignment of blocks of router port numbers for different types of data flows is shown in Table 1. Additional blocks of port numbers may be defined as necessary.

Table 1: Typical Example of Assignment of Router Port Numbers

Route Point Type – Ground Station Side	Connect Ports	Data Flow Direction	Connect Ports	Route Point Type – Spacecraft Control System Side
Telemetry Data Sources	11xxx	–>	21xxx	Telemetry Data Destinations
Command Data Destinations	12xxx	<–	22xxx	Command Data Sources
Control Data Destinations	13xxx	<–	23xxx	Control Data Sources
Tracking Data Sources	14xxx	–>	24xxx	Tracking Data Destinations
Station Status Block Sources	15xxx	–>	25xxx	Station Status Block Destinations
Auxiliary Data Sources	16xxx	–>	26xxx	Auxiliary Data Destinations

Figure 1 illustrates how data flows between various elements of the satellite control network are established. Remote ground stations and other ground systems elements external to the MOC connect from the left side of the figure, while spacecraft command and control workstations plus other data processing clients connect from the right side. For each remote ground station a pair of telemetry and command ports is dedicated to each supported mission, as indicated by the small squares along the left side of the FrameRouter box in the block diagram. For example, the primary frontend processor in the Berkeley Ground Station (BGS) connects to the sockets labeled TLM_BGS_THEMIS_PTP1 and CMD_BGS_THEMIS_PTP1 for a pass support of one of the five THEMIS spacecraft, while other pairs of sockets are used for data flows with RHESSI or FAST.[1,5,6] Shown on the right side of the block diagram are the spacecraft command and control workstations for THEMIS, RHESSI and FAST. All of these run the Integrated Test and Operations System (ITOS).[7] Four pairs of telemetry and command routes are shown in an activated state, supporting simultaneous passes of two of the five THEMIS spacecraft (THEMIS A and THEMIS C) , RHESSI and FAST via the Berkeley (BGS), Wallops (WGS) and McMurdo Ground Station (MGS), respectively. The routing threads are started during the pre-pass set-up for a given pass support, allowing the remote ground stations and the command and control systems to establish their connections several minutes prior to the scheduled time of spacecraft acquisition.

To support on-orbit operations of eight spacecraft, the Berkeley MOC uses a total of 18 ground station antennas located at 10 different locations, including those of two commercial ground networks.[8,9] In addition, the THEMIS missions also utilizes NASA's Tracking and Data Relay Satellite System (TDRSS) to support special operations such as orbit maneuvers. For the latter, connectivity for telemetry and command data flows is established via the White Sands Complex (WSC).

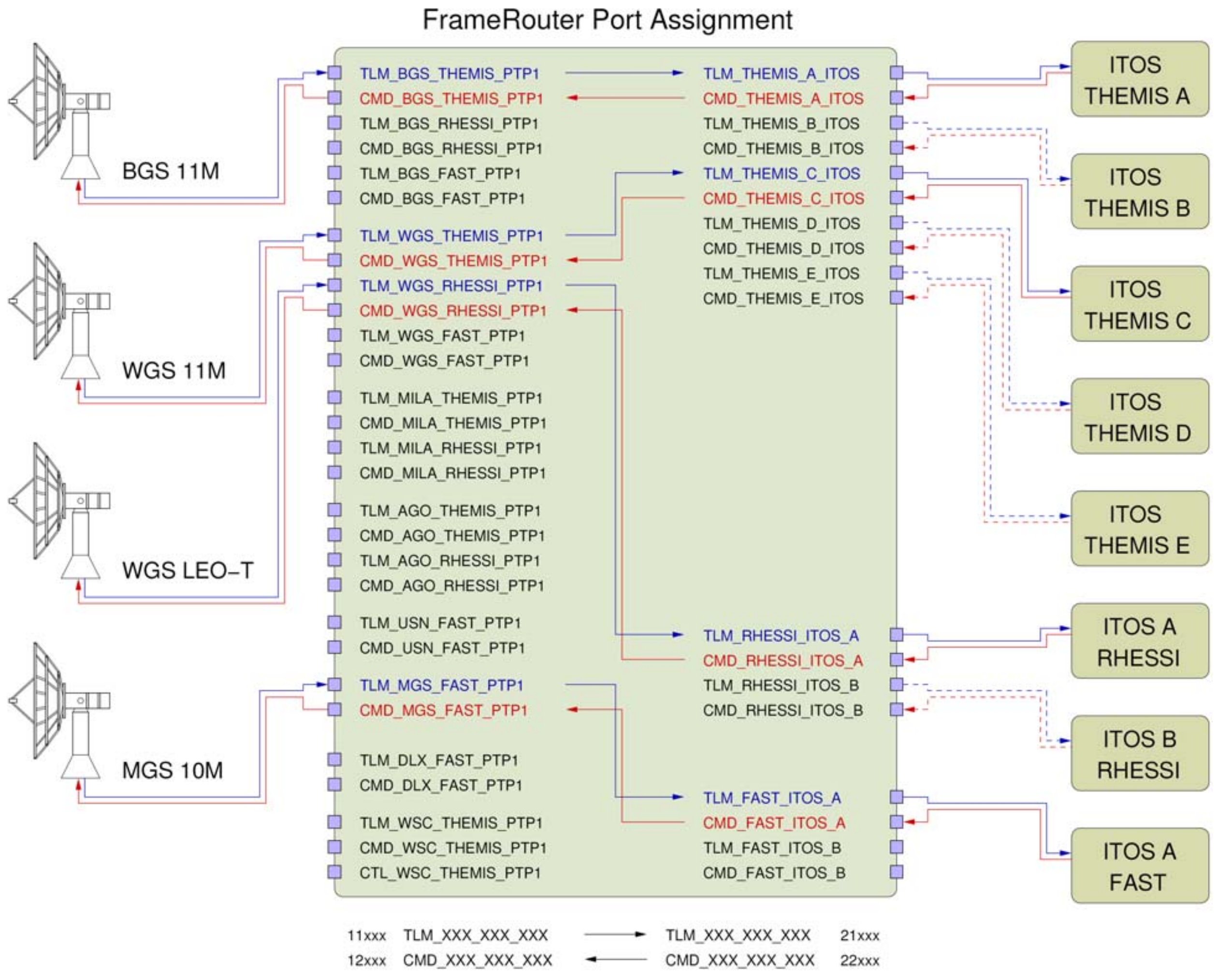


Figure 1: Assignment of route point mnemonics associated with ports numbers, and illustration of telemetry and command data routing.

RELAYING DATA STREAMS

If telemetry streams need to distributed to a larger number of client applications, the network topology can be easily expanded by configuring multiple instances of FrameRouter that are linked by instances of FrameRelay to relay data streams between various network segments, as is shown in Figure 2. Note that FrameRouter *MOC* is located on the secure operations network, while FrameRouter *SSL* runs on the open SSL LAN. Also note that only the primary ITOS system for each spacecraft has command authority. Any additional instances of ITOS on either the secure MOC network or on the open SSL LAN are only allowed to receive telemetry data.

Another application that connects to FrameRouter SSL to receive and process real-time telemetry data streams from the five THEMIS spacecraft is the Berkeley Trending and Plotting System (BTAPS).[10] BTAPS decommutates all spacecraft engineering data and stores these in a MySQL database.[11] Data can subsequently be retrieved for real-time or post-event trending and plotting analyses. These tools were invaluable during mission integration and environmental testing.

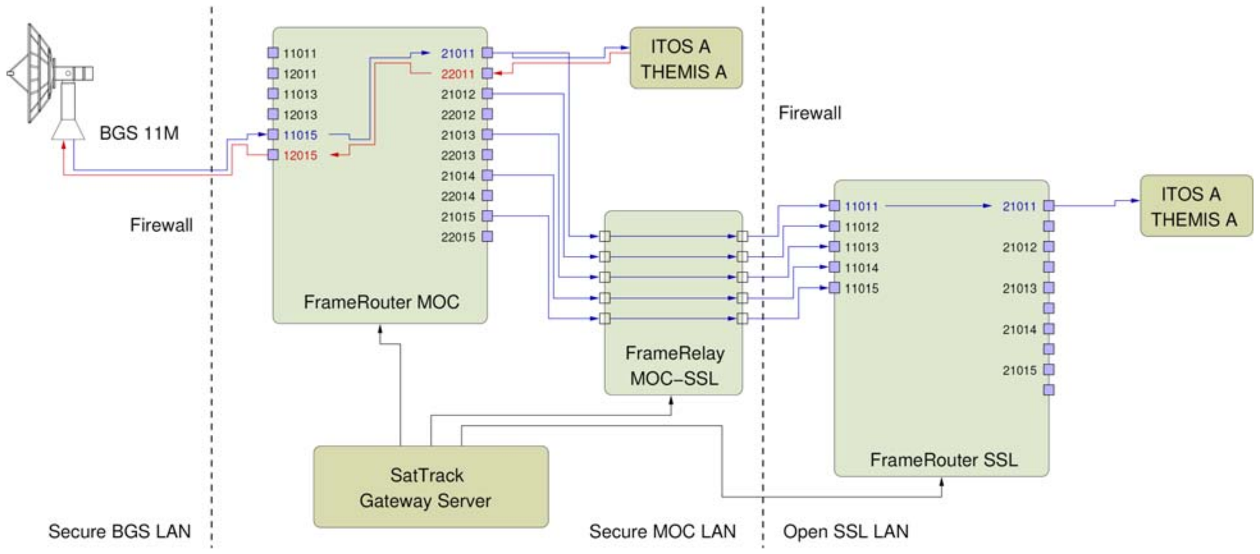


Figure 2: Linking of FrameRouter *MOC* and FrameRouter *SSL* by FrameRelay *MOC-SSL*. All instances of FrameRouter and FrameRelay are controlled remotely and configured for pass supports by the Gateway Server.

COMMAND LANGUAGE

All instances of FrameRouter and FrameRelay are remote controlled via the Gateway Server using a simple command language. To control either application, the user needs to sign on to the Gateway Server as a *scheduler* or a *config* client. Once connected, the following requests can be submitted for controlling routes or polling status from each FrameRouter or FrameRelay:

```
sgs> select              router <router name>
sgs> select              relay <relay name>
sgs> makeroute           <srcPort | srcMnemonic> [–>] <dstPort | dstMnemonic>
sgs> breakroute          <srcPort | srcMnemonic> [–>] <dstPort | dstMnemonic>
sgs> breakroute          <route ID>
sgs> makeroutes          facility <facility name> object <object name>
sgs> breakroutes         facility <facility name> object <object name>
sgs> breakallroutes      [router <router name>]
sgs> showroutes          router <router name>
sgs> showallroutes
sgs> showroutepoints     router <router name>
sgs> makerelay           <srcRouterName> <srcPort> [–>] <dstRouterName> <dstPort>
sgs> breakrelay          <srcRouterName> <srcPort> [–>] <dstRouterName> <dstPort>
sgs> breakrelay          <relay ID>
sgs> makerelays          facility <facility name> object <object name>
sgs> breakrelays         facility <facility name> object <object name>
sgs> breakallrelays      [relay <relay name>]
sgs> showrelays          relay <relay name>
sgs> showallrelays
sgs> setaccess           <mnemonic | port> <IP address mask>
```

Make and break requests need to be preceded by at least one select request. Furthermore, certain rules need to be followed to successfully make or break routes and relays. For instance, a route point must not be part of an already existing route or relay when a new route or relay involving the same route point is to be established. The *setaccess* request can be submitted to the Gateway Server to dynamically change the access permissions for a given port on a selected FrameRouter, for example to fail-over command authority from one ITOS system to another. A typical sequence of commands would be:

```
myhost%  telnet xxx.xxx.xxx.199 9000
sgs>  scheduler
sgs>  select router moc
sgs>  setaccess 22021 xxx.xxx.xxx.246
sgs>  quit
myhost%
```

AUTOMATION SUPPORT

There are basically two methods to implement automated operation. The first method is to configure the Gateway Server such that routes are automatically made before the scheduled times of Acquisition of Signal (AOS) and broken after Loss of Signal (LOS). The second method is to incorporate remote commands in ITOS Spacecraft Test and Operations Language (STOL) scripts, taking advantage of the FrameRouter and FrameRelay control language described above.

In the former case the following sample entries need to be placed into the configuration file of the Gateway Server:

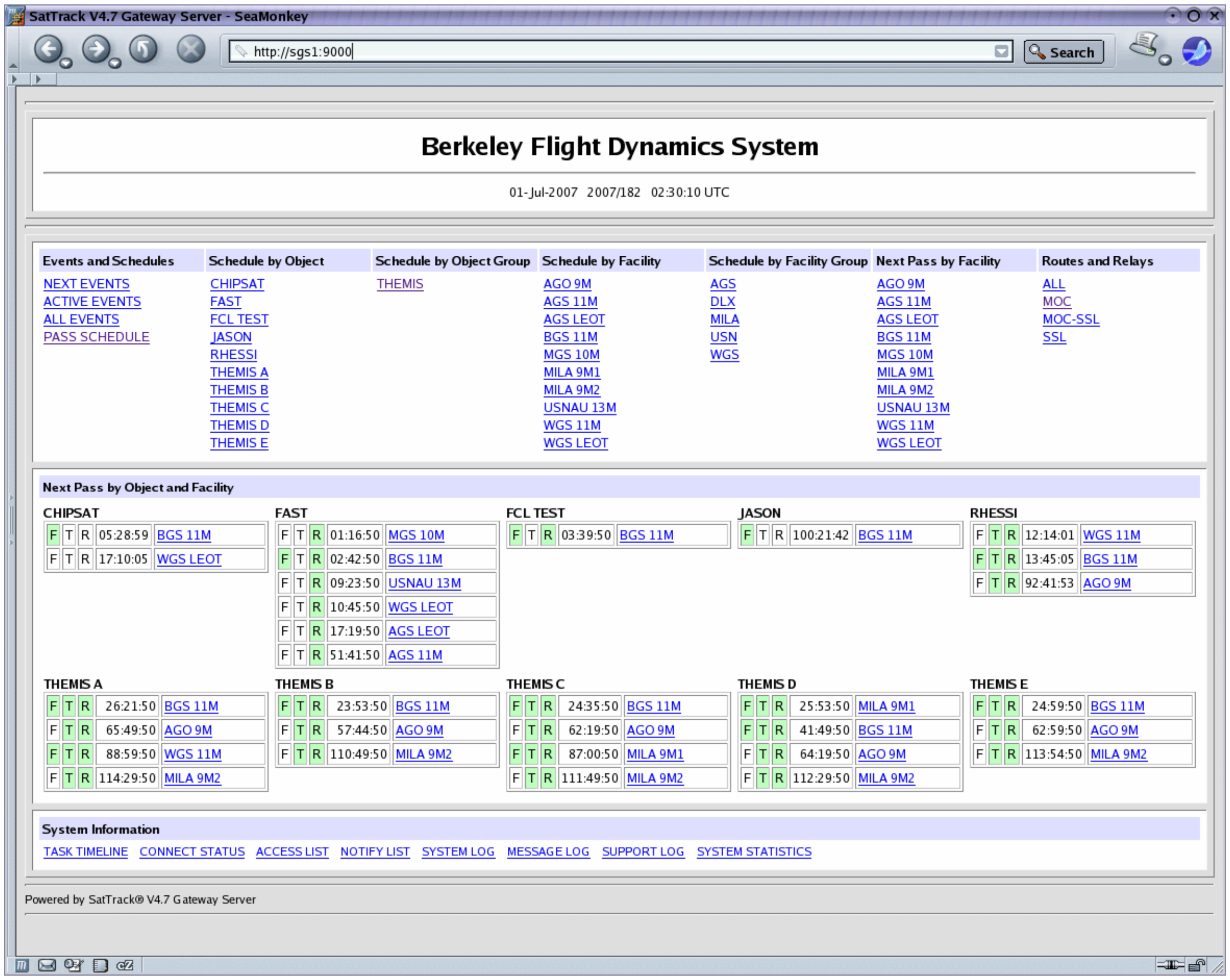
```
Facility:              BGS 11M
Object:                RHESSI
Data route:            MOC TLM_BGS_RHESSI_PTP1 –> TLM_RHESSI_ITOS_A
Data route:            MOC CMD_RHESSI_ITOS_A –> CMD_BGS_RHESSI_PTP1
Data route:            SSL TLM_RHESSI_RELAY_A –> TLM_RHESSI_ITOS_A
Data relay:            MOC-SSL MOC TLM_RHESSI_ITOS_A –> SSL TLM_RHESSI_RELAY_A
Make routes:           AOS−180
Break routes:          LOS+60
Make relays:           AOS−120
Break relays:          LOS+30
Enable route control:  yes
```

In this example, there are two routes specified for router *MOC* – one each for telemetry and command. Also, one route is specified for router *SSL*. The two routers – *MOC* and *SSL* – are linked by one relay named *MOC-SSL*. The times for making and breaking routes and relays are specified in seconds relative to the scheduled times of AOS and LOS. If route control is enabled, then these routes and relays are controlled by the Gateway Server. With the second method, all routes and relays can be made or broken by submitting the following request sequence to the Gateway Server from within the secure operations network environment – either interactively from a terminal window or out of an ITOS STOL script:

```
myhost%  telnet xxx.xxx.xxx.199 9000
sgs>  scheduler
sgs>  makeroutes facility bgs 11m object rhessi
...
sgs>  breakroutes facility bgs 11m object rhessi
sgs>  quit
myhost%
```

WEB INTERFACE

A web interface shown in Figure 3 allows viewing the status of the scheduling system, and in particular the connect status of any running FrameRouter or FrameRelay application. The *Main Menu* page features hyperlinks to view schedules for individual spacecraft called *objects* or ground stations called *facilities*. The *Route Status* page shown in Figure 4 below can be viewed by following the corresponding links under *Routes and Relays* towards the right side of the Main Menu. For each FrameRouter and FrameRelay connected to the Gateway Server, any active routing or relay threads are listed with their corresponding route points in mnemonic form and with their port number. Once a TCP/IP connection on a particular route point is established, the IP addresses of the connected hosts will be different from 0.0.0.0 and is shown on a green background. Note that there can be only one connection on a source route point, but multiple on a destination route point. The number of bytes received and forwarded will be shown also. This page is refreshed automatically once every 30 seconds.


SatTrack V4.7 Gateway Server - SeaMonkey

http://sgs1:9000

Berkeley Flight Dynamics System

01-Jul-2007 2007/182 02:30:10 UTC

Events and Schedules	Schedule by Object	Schedule by Object Group	Schedule by Facility	Schedule by Facility Group	Next Pass by Facility	Routes and Relays
NEXT EVENTS	CHIPSAT	THEMIS	AGO 9M	AGS	AGO 9M	ALL
ACTIVE EVENTS	FAST		AGS 11M	DLX	AGS 11M	MOC
ALL EVENTS	FCL TEST		AGS LEOT	MILA	AGS LEOT	MOC-SSL
PASS SCHEDULE	JASON		BGS 11M	USN	BGS 11M	SSL
	RHESSI		MGS 10M	WGS	MGS 10M	
	THEMIS A		MILA 9M1		MILA 9M1	
	THEMIS B		MILA 9M2		MILA 9M2	
	THEMIS C		USNAU 13M		USNAU 13M	
	THEMIS D		WGS 11M		WGS 11M	
	THEMIS E		WGS LEOT		WGS LEOT	

Next Pass by Object and Facility

CHIPSAT

F	T	R	05:28:59	BGS 11M
F	T	R	17:10:05	WGS LEOT

FAST

F	T	R	01:16:50	MGS 10M
F	T	R	02:42:50	BGS 11M
F	T	R	09:23:50	USNAU 13M
F	T	R	10:45:50	WGS LEOT
F	T	R	17:19:50	AGS LEOT
F	T	R	51:41:50	AGS 11M

FCL TEST

F	T	R	03:39:50	BGS 11M

JASON

F	T	R	100:21:42	BGS 11M

RHESSI

F	T	R	12:14:01	WGS 11M
F	T	R	13:45:05	BGS 11M
F	T	R	92:41:53	AGO 9M

THEMIS A

F	T	R	26:21:50	BGS 11M
F	T	R	65:49:50	AGO 9M
F	T	R	88:59:50	WGS 11M
F	T	R	114:29:50	MILA 9M2

THEMIS B

F	T	R	23:53:50	BGS 11M
F	T	R	57:44:50	AGO 9M
F	T	R	110:49:50	MILA 9M2

THEMIS C

F	T	R	24:35:50	BGS 11M
F	T	R	62:19:50	AGO 9M
F	T	R	87:00:50	MILA 9M1
F	T	R	111:49:50	MILA 9M2

THEMIS D

F	T	R	25:53:50	MILA 9M1
F	T	R	41:49:50	BGS 11M
F	T	R	64:19:50	AGO 9M
F	T	R	112:29:50	MILA 9M2

THEMIS E

F	T	R	24:59:50	BGS 11M
F	T	R	62:59:50	AGO 9M
F	T	R	113:54:50	MILA 9M2

System Information

TASK TIMELINE CONNECT STATUS ACCESS LIST NOTIFY LIST SYSTEM LOG MESSAGE LOG SUPPORT LOG SYSTEM STATISTICS

Powered by SatTrack® V4.7 Gateway Server

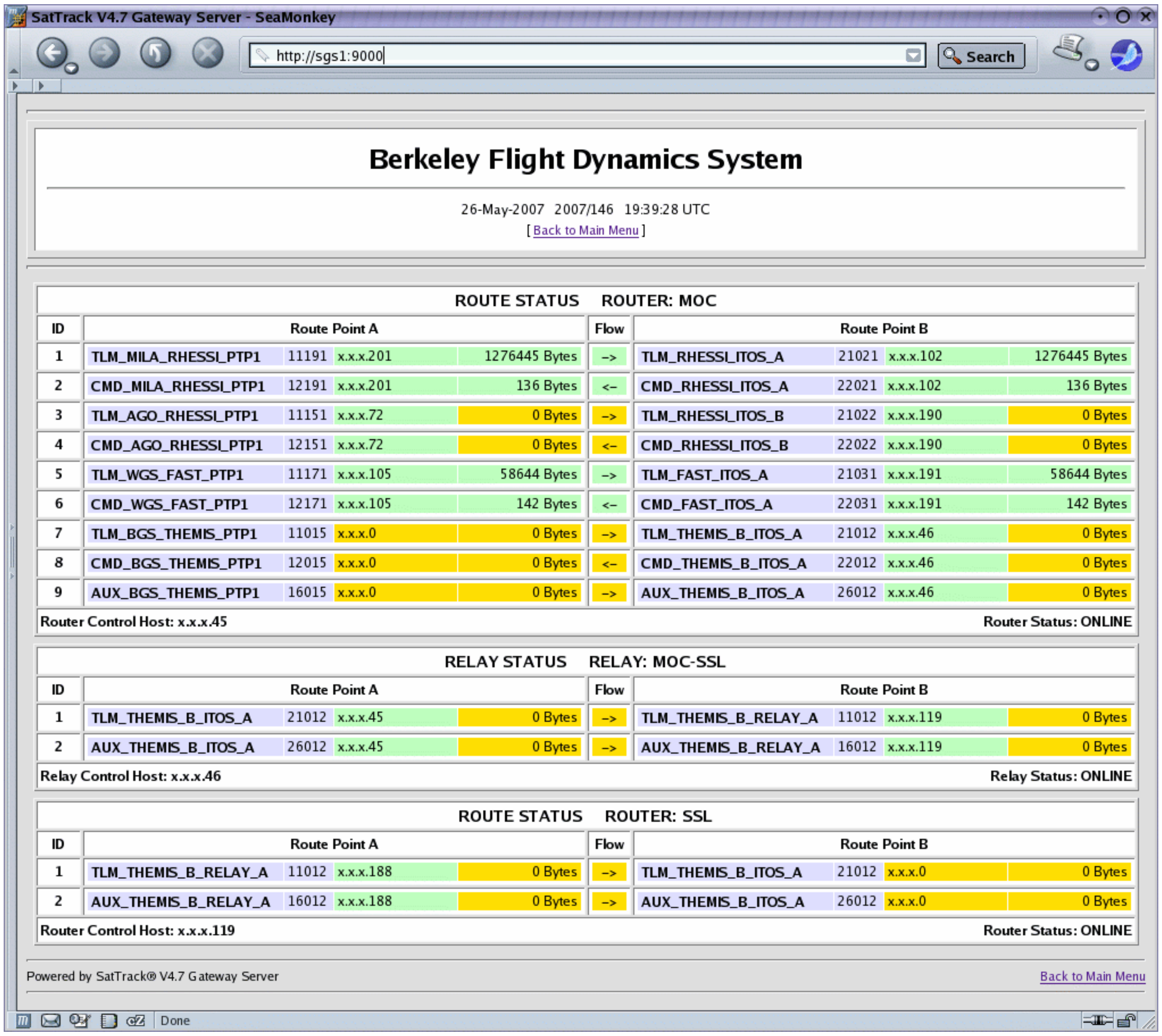
Figure 3: Sample web page showing current satellite pass support schedule.

SYSTEM REQUIREMENTS

As part of the SatTrack® Suite, FrameRouter, FrameRelay and Gateway Server were written in the C programming language.[4] Typical hardware requirements are relatively inexpensive Intel i686 compatible systems with 2-3 GHz clock speeds and 512 Mbytes of memory running the Red Hat Enterprise Linux operating system, or SPARC based platforms by Sun Microsystems with SunOS 5.9 or higher.[12-15] A software port to Apple Mac OS X is currently underway.[16]

Regarding overall data transfer performance, higher clock speeds will typically provide better throughput, although limitations will likely be imposed by the available bandwidth on the network segments to the remote ground stations. Within the 100BASE-T local area network at the Berkeley MOC, telemetry frames were successfully distributed in real-time at 4 Mbps to multiple telemetry processing systems across a firewall.

To provide operational redundancy for a mission critical FrameRouter application, it is advisable to install a hot back-up system that can be configured to the same IP address as the primary system. If the primary system fails, the back-up system can take over with only minimal loss of service time and without requiring any other network reconfiguration.



Berkeley Flight Dynamics System

26-May-2007 2007/146 19:39:28 UTC
[Back to Main Menu]

ROUTE STATUS ROUTER: MOC

ID	Route Point A				Flow	Route Point B			
1	TLM_MILA_RHESSI_PTP1	11191	x.x.x.201	1276445 Bytes	-->	TLM_RHESSI_ITOS_A	21021	x.x.x.102	1276445 Bytes
2	CMD_MILA_RHESSI_PTP1	12191	x.x.x.201	136 Bytes	<--	CMD_RHESSI_ITOS_A	22021	x.x.x.102	136 Bytes
3	TLM_AGO_RHESSI_PTP1	11151	x.x.x.72	0 Bytes	-->	TLM_RHESSI_ITOS_B	21022	x.x.x.190	0 Bytes
4	CMD_AGO_RHESSI_PTP1	12151	x.x.x.72	0 Bytes	<--	CMD_RHESSI_ITOS_B	22022	x.x.x.190	0 Bytes
5	TLM_WGS_FAST_PTP1	11171	x.x.x.105	58644 Bytes	-->	TLM_FAST_ITOS_A	21031	x.x.x.191	58644 Bytes
6	CMD_WGS_FAST_PTP1	12171	x.x.x.105	142 Bytes	<--	CMD_FAST_ITOS_A	22031	x.x.x.191	142 Bytes
7	TLM_BGS_THEMIS_PTP1	11015	x.x.x.0	0 Bytes	-->	TLM_THEMIS_B_ITOS_A	21012	x.x.x.46	0 Bytes
8	CMD_BGS_THEMIS_PTP1	12015	x.x.x.0	0 Bytes	<--	CMD_THEMIS_B_ITOS_A	22012	x.x.x.46	0 Bytes
9	AUX_BGS_THEMIS_PTP1	16015	x.x.x.0	0 Bytes	-->	AUX_THEMIS_B_ITOS_A	26012	x.x.x.46	0 Bytes

Router Control Host: x.x.x.45 Router Status: ONLINE

RELAY STATUS RELAY: MOC-SSL

ID	Route Point A				Flow	Route Point B			
1	TLM_THEMIS_B_ITOS_A	21012	x.x.x.45	0 Bytes	-->	TLM_THEMIS_B_RELAY_A	11012	x.x.x.119	0 Bytes
2	AUX_THEMIS_B_ITOS_A	26012	x.x.x.45	0 Bytes	-->	AUX_THEMIS_B_RELAY_A	16012	x.x.x.119	0 Bytes

Relay Control Host: x.x.x.46 Relay Status: ONLINE

ROUTE STATUS ROUTER: SSL

ID	Route Point A				Flow	Route Point B			
1	TLM_THEMIS_B_RELAY_A	11012	x.x.x.188	0 Bytes	-->	TLM_THEMIS_B_ITOS_A	21012	x.x.x.0	0 Bytes
2	AUX_THEMIS_B_RELAY_A	16012	x.x.x.188	0 Bytes	-->	AUX_THEMIS_B_ITOS_A	26012	x.x.x.0	0 Bytes

Router Control Host: x.x.x.119 Router Status: ONLINE

Powered by SatTrack® V4.7 Gateway Server Back to Main Menu

Done

Figure 4: Sample web page showing current route and relay status information. IP addresses and port numbers have been altered to maintain network security of operational systems.

CONCLUSIONS

The frame routing software tools described above have been fully operational at the MOC at UCB/SSL since early 2006 and supported approximately 1500 passes for the THEMIS mission since its launch on February 17, 2007. Furthermore, the system has been running continuously for longer than 180 days without any software restarts, supporting up to 50 passes daily for all missions combined. The software demonstrated its advantages and strengths in performance particularly during the THEMIS launch and early orbit campaign when real-time telemetry data needed to be distributed to a much larger number of workstations to allow systems and subsystems engineers to monitor on-orbit checkout, maneuver operations and science instrument deployment of the new satellite constellation.

ACKOWLEDGEMENTS

The THEMIS mission is operated at Space Sciences Laboratory, University of California at Berkeley, under NASA contract NAS5-02099.

REFERENCES

1. Bester, M., Lewis, M., Quinn, T. and Rauch-Leiba, J., "Automations of Operations and Ground Systems at U.C. Berkeley," Proceedings of the 5th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations (RCSGSO), Pasadena, CA, July 2003.
2. Recommended Standards (Blue Books), Consultative Committee for Space Data Systems (CCSDS), Reston, VA, www.ccsds.org.
3. Angelopoulos, V. et al., "Time History of Events and Macroscale Interactions during Substorms (THEMIS) – A NASA Medium-class Explorer Mission Phase A Concept Study Report," University of California, Berkeley, CA, October 2002.
4. SatTrack® Suite V4.7, Bester Tracking Systems Inc., Suisun City, CA, www.bester.com.
5. Programmable Telemetry Processor (PTP), Avtec Systems Inc., Chantilly, VA, www.avtec.com.
6. Enertec 3801-20 Multimission Baseband Processor, Zodiac Data Systems Inc., Alpharetta, GA, www.enertecgroup.com
7. Integrated Test and Operations System (ITOS), the Hammers Company, Greenbelt, MD, www.hammers.com.
8. Universal Space Network (USN), Horsham, PA, www.uspacenet.com.
9. DataLynx™ Satellite Control Network, Honeywell Aerospace, Columbia, MD, www.honeywell.com.
10. Cruce, P., Roberts, B., Quinn, T. and Bester, M., "A Database Centered Approach to Satellite Engineering Data Storage, Access, and Display", Proceeding of the 21st Annual AIAA/USU Conference on Small Satellites, Logan, UT, August 2007.
11. MySQL Open Source Database, MySQL A, Cupertino, CA, www.mysql.com.
12. Rackmount Servers, Silicon Mechanics Inc., Bothell, WA, www.siliconmechanics.com.
13. Shuttle XPC, Shuttle Computer Group Inc., City of Industry, CA, www.shuttle.com.
14. Enterprise Linux, Red Hat Inc., Raleigh, NC, www.redhat.com.
15. SunOS 5.9 and 5.10 Operating Systems, Sun Microsystems, Inc., www.sun.com.
16. Mac OS X Operating System, Apple Inc., Cupertino, CA, www.apple.com.