GLAST Large Area Telescope

Instrument Flight Software
Flight Unit Peer Review
16 September 2004

Inter-task Communications

Presenter: Sergio Maldonado
Author: Anthony Waite
Stanford Linear Accelerator Center

apw@slac.stanford.edu
650-926-2075

Inter-task Communications: Requirements

- **FSW General Requirements:**
  - **EPU-SIU Interface (5.2.1.1)**
    - **The EPU FSW shall communicate with the SIU via a custom CPU-to-CPU serial message protocol.**
  - **SC Command Processing by SIU (5.3.3.1)**
    - **The SIU FSW shall receive and process LAT commands from the SC via the CTDB in the order that they are received.**
  - **Command Execution Notification (5.3.3.2)**
    - **All LAT commands from the SC shall generate time-tagged telemetry notification of their dispatch for execution in the order they are received.**
  - **Command Completion Status (5.3.3.3)**
    - **The FSW shall generate time-tagged telemetry notification about command completion status (success or general cause of failure) at the time such information becomes available.**
  - **Command Dispatch (5.3.3.4)**
    - **Commands from the SC shall be placed into the FSW command dispatch queue immediately upon receipt.**
  - **Command Format (5.3.3.5)**
    - **The FSW shall be able to process commands of the format described in the LAT FSW Telecommand and Telemetry Formats document.**

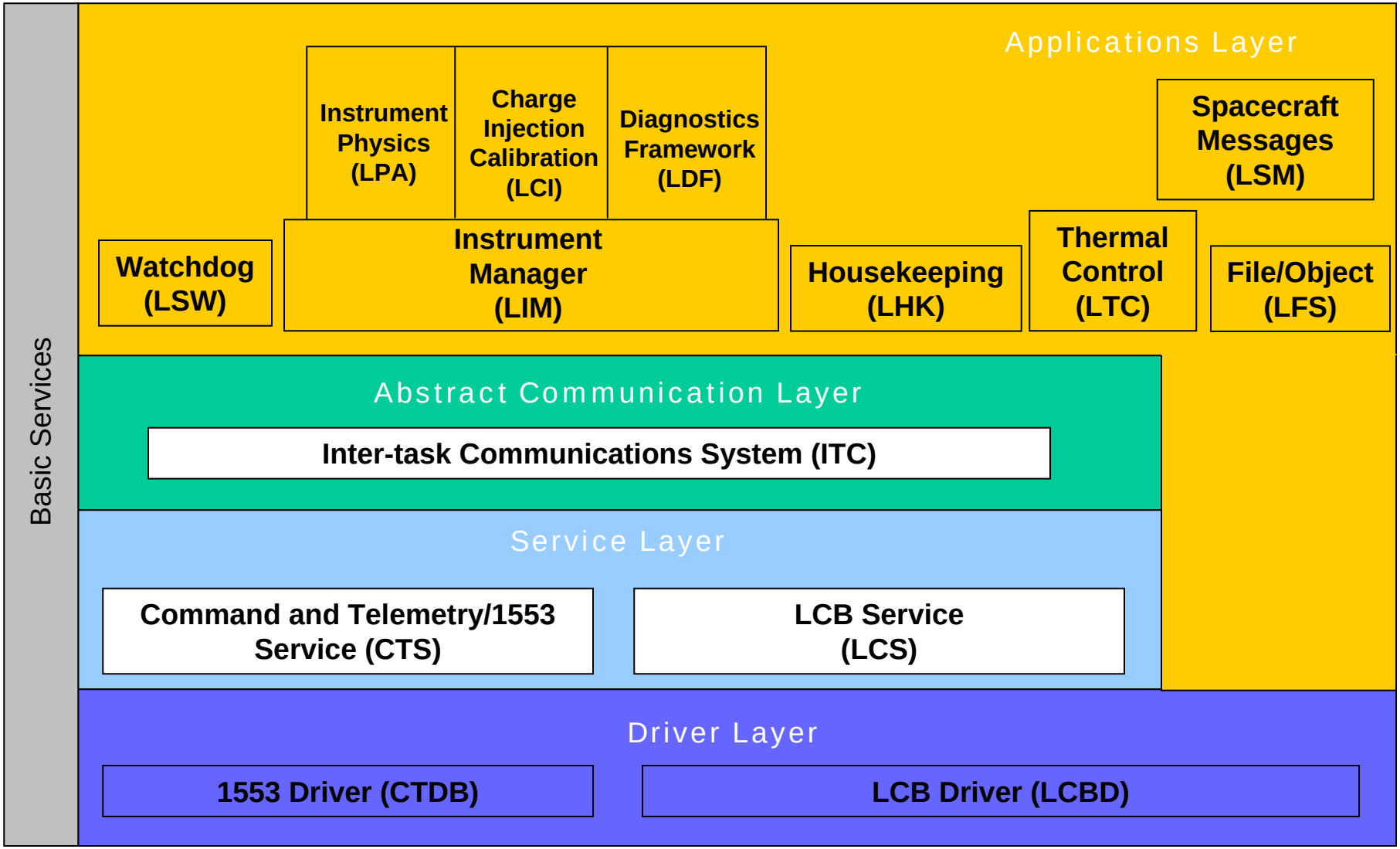
Inter-task Communications: Requirements (cont'd)

- **FSW General Requirements (cont'd):**
  - **Command Execution (5.3.3.6)**
    - **The SIU FSW shall be able to reconfigure and direct the operation of the instrument through a combination of direct execution of commands from the SC, or by preprogrammed operational sequences triggered by commands from the SC.**
  - **Command Execution Sequence (5.3.3.7)**
    - **Immediate (single) commands from different sources shall be capable of interleaving with other immediate (single) commands and with block commands for uploads and dumps.**
  - **Multiple Block Commands (5.3.3.8)**
    - **The FSW shall execute block commands for uploads or dumps serially (i.e., not simultaneously, not interleaved with other commands).**
  - **Data Integrity Errors (5.3.3.9.1)**
    - **The SIU FSW shall validate commands prior to execution in case of an error during transmission.**
  - **Parameter Validation (5.3.3.9.2)**
    - **The SIU FSW shall validate command parameters prior to execution.**

Components of Inter-task Communications

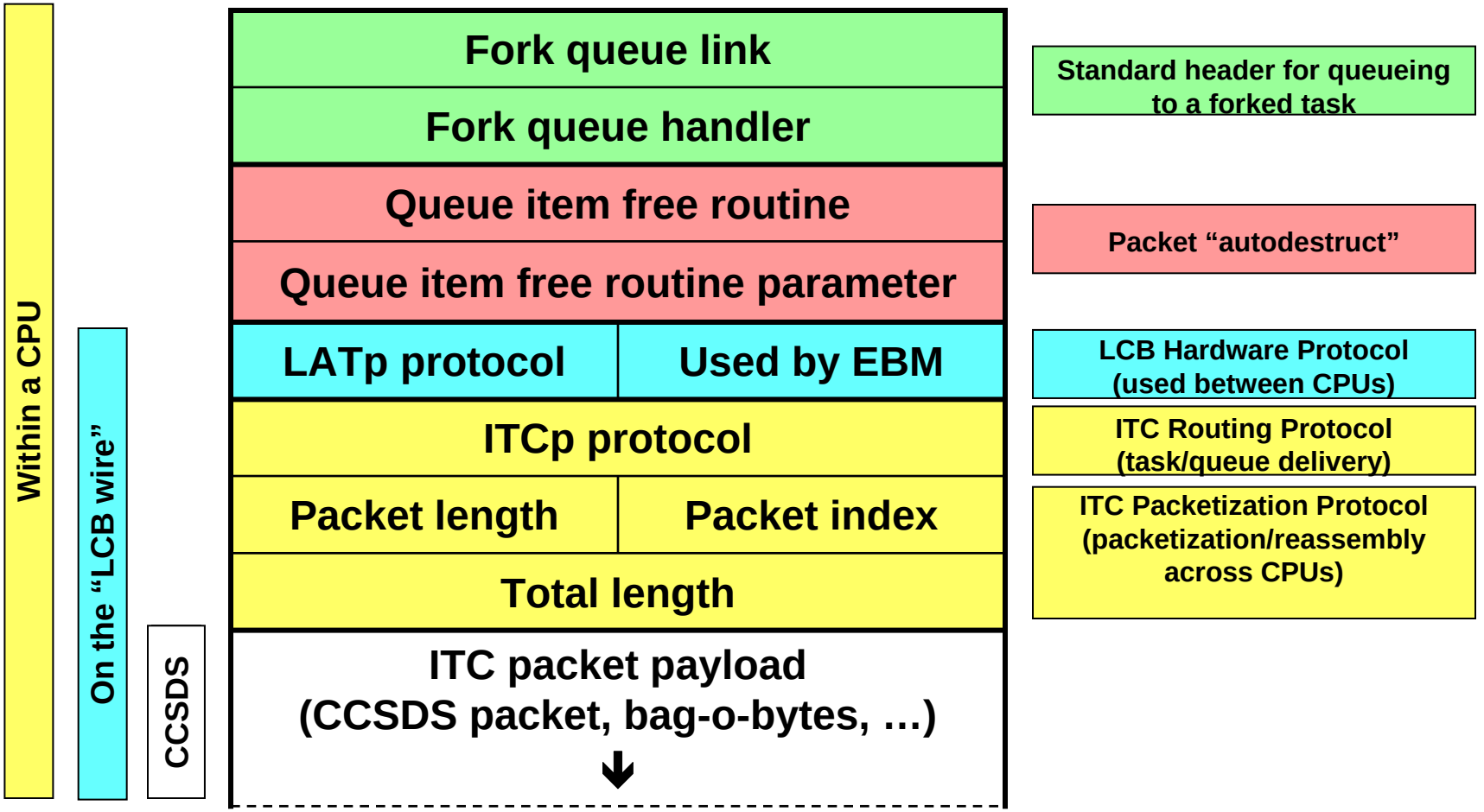
- These requirements are implemented as an integrated system, built from 4 FSW packages:
  - The ITC package defines the Inter-task Communications standard:
    - ITC defines a common communications standard to unify communications between tasks on either the same or different CPUs.
    - It also provides the services to build up tasks that are capable of communicating according to the standard.
    - ITC is not itself a task, it's a toolbox.
  - The CTS package defines the Command and Telemetry/1553 Service:
    - It provides the service layer between applications (high level tasks) and the Command and Telemetry Database (CTDB) hardware driver.
    - It is constructed using the ITC toolbox, allowing CTS to present an ITC compliant (uniform) interface to the application layer tasks.
    - It provides communication:
      - Spacecraft to CPU
      - CPU to spacecraft
  - The LCS package defines the LCB Service:
    - It provides the service layer between applications (high level tasks) and the LCB hardware driver.
    - It is constructed using the ITC toolbox, allowing LCS to present an ITC compliant (uniform) interface to the application layer tasks.
    - It provides communication:
      - CPU to CPU
      - CPU to SSR
  - The LCM package defines the LAT Communications Manager:
    - It controls the reporting/messaging behavior of all tasks on the system and provides access to global and task level communications statistics

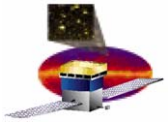
FSW Layer Model



The ITC Toolbox (1)

- ITC achieves uniformity by defining a single message header format applicable to all communications situations:





The ITC Toolbox (2)

- ITC pre-allocates resources for a maximum of 256 queues per CPU
  - Queue-space is segmented
    - Maximum of 8 (prioritized) queues per task
    - Maximum of 32 tasks
- ITC understands four protocols
  - LAT instrument data (payload is the instrument defined format)
  - Raw (payload is simply “bag-o-bytes”)
  - Command (payload contains CCSDS telecommand but task dispatch handled by the “ITC protocol” (ITCp)
  - APID (payload contains CCSDS telecommand and task dispatch is based on the APID)
- Tasks wishing to participate must initialize via ITC
  - Create an ITC task description (`ITC_createTask()`)
  - Add queues to task description (`ITC_addQueue()`)
  - Attach protocol handlers to queues (`ITC_attachApid()`)
  - Start task (`ITC_startTask()`)
- Once started, the process of sending a message is
  - Allocate memory (`ITC_allocW()`)
  - Fill in the message payload (user does this)
  - Send to destination (`ITC_send()`)
    - Destination specified by Node, Task, Queue

The ITC Toolbox (3)

- **Node, Task and Queue identifiers are pre-assigned by ITC**
  - **Pre-assigned node list**
    - **Note special nodes**
      - **Science Data Interface**
      - **Spacecraft**
    - **Broadcast provided for master/slave operations**

Node ID	Usage
0	SIU
1	EPU 0
2	EPU 1
3	EPU 2
4	EPU 3
5	Science Data Interface
6	Spacecraft
7	Broadcast

  - **Pre-assigned task list**
    - **17 assigned so far**
      - **Includes service tasks**

Task ID	Usage
0	Anonymous task (unused)
1	LCM
2	LFS master
3	LFS slave
4	LHK master
5	LHK slave
n	Etc.

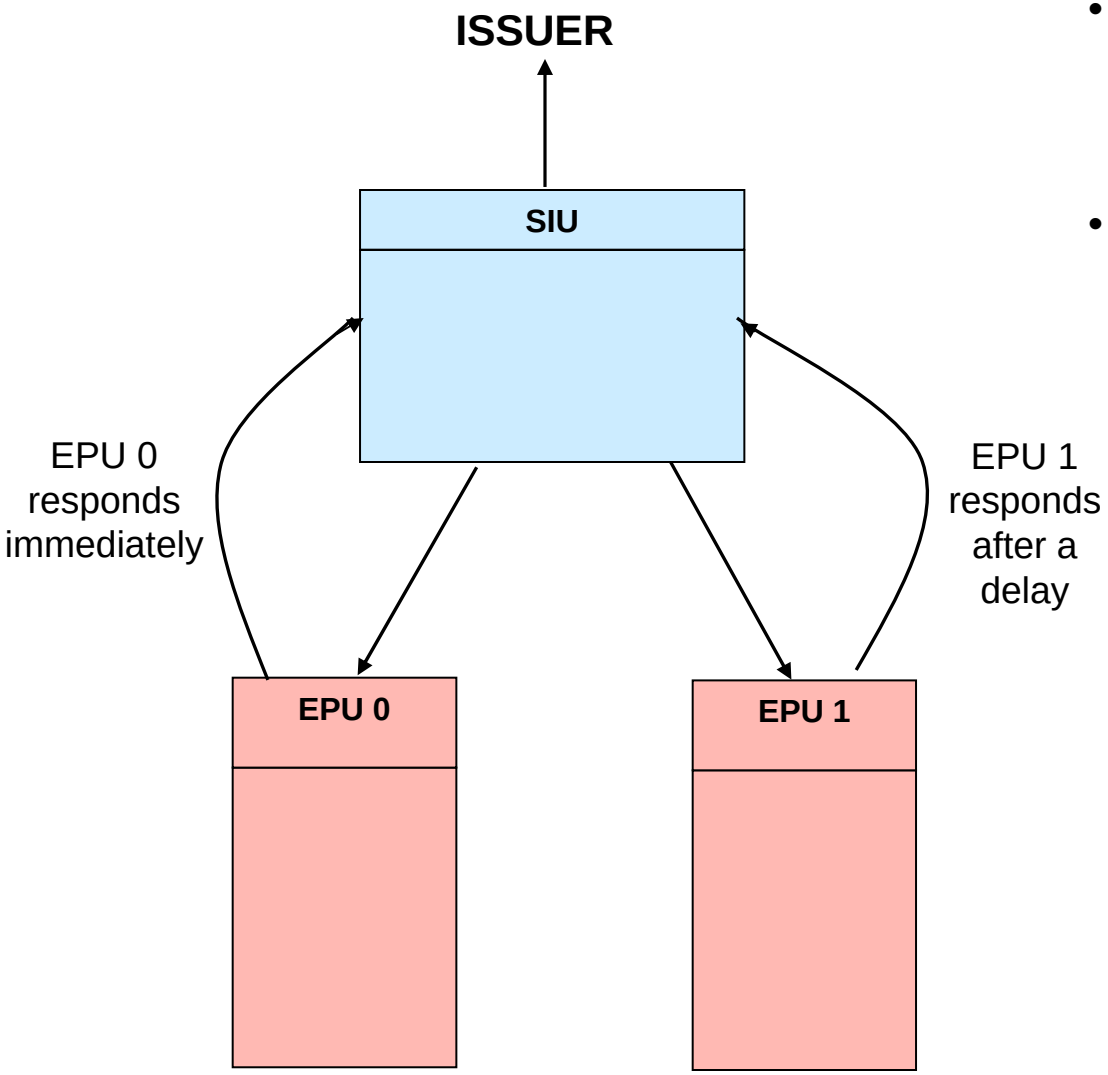
The ITC Toolbox (4)

- **Node, Task and Queue identifiers are pre-assigned by ITC**
  - **Pre-assigned queue list**
    - **ITC reserves first two queues for itself**
    - **Names are suggestive only: application writers may choose how to use the six user queues**
  - **Pre-assigned protocol list**
    - **"Command" dispatches:**
      - **To task by ITCp protocol**
      - **Within task by function code**
    - **"APID" dispatches:**
      - **To task by APID**
      - **Within task by function code**

Queue ID	Usage
0	Meta (reserved by ITC)
1	Synch (reserved by ITC)
2	Command Extra
3	Command
4	Control Extra
5	Control
6	Bulk Extra
7	Bulk

Task ID	Usage
0	LAT instrument format
1	Raw (bag-o-bytes)
2	Command (CCSDS payload)
3	APID (CCSDS payload)

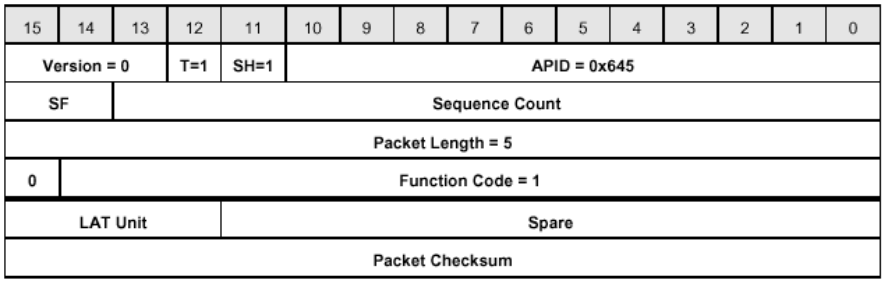
ITC: Scatter/Gather Problem



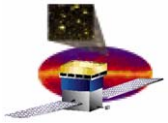
- When a master task sends a message requiring a response from multiple slaves, how are the responses synchronized?
- ITC provides the “synch” queue to address scatter/gather:
  - When a master sends a message, responses to which must be gathered, a special bit in the message packet header is set and master disables all queues except “meta” and “synch”
  - Slave tasks reply to the master on the master’s synch queue
  - Once all responses have been received on the master (or transaction times out), master re-enables all queues.

The CTS Service

- **Input Processing**
  - Receives CCSDS formatted telecommands from the spacecraft.
  - Verifies CCSDS header.
  - Copies bare CCSDS telecommand into standard ITC message (with protocol “APID”)
  - Uses the ITC dispatch mechanism to do the first tier dispatch: to a task based on CCSDS APID.



- **Output Processing**
  - Provides three ITC user input queues:
    - Telecommand (used for sending, for example, the repoint request to the spacecraft).
    - High priority telemetry (reserved for “alert” telemetry).
    - Low priority telemetry (“diagnostic” telemetry).
  - Messages appearing on queues are:
    - Checked for CCSDS compliance (general header format, APID in range, overall length check).
    - Forwarded to the CTDB driver for transmission to the spacecraft.



The LCS Service

- **Input Processing**
  - **Receives ITC formatted packets from LCB driver.**
  - **Reassembles packets as necessary.**
  - **Uses the ITC protocol header to dispatch to task.**

- **Output Processing**
  - **Provides all six ITC user input queues:**
    - **Preserves the queue designation across CPUs.**
  - **Packetizes a message as necessary.**
  - **Forwards the packets to the LCB driver for transmission.**
  - **Provides specialized processing to SSR:**
    - **Strips the ITC protocol from the packet.**
    - **Introduces the mandated `0xffffffff` framing word.**

LCM: Architecture

- **LCM is an application-level task that manages the overall state of the communications system and collects global statistics:**
  - **LCM responds to telecommands when the ground requires information about command processing and the inter-task communications system.**
  - **LCM can also change reporting levels and behavior of user tasks**
    - **Success responses**
    - **Error reporting behavior**

Inter-task Communications: Commands and Telemetry

- Telecommands (implemented in the LCM application task)
  - Task management (these functions apply task by task)
    - Suppress reporting of successful command execution
    - Alter the “messaging level”
    - Request detailed communications statistics (by queue/protocol)
      - NOTE: CCSDS command counters are in critical housekeeping
  - Global functions
    - Request statistics on ITC resource usage

- Telemetry
  - Task by task
    - Generate the command success/failure telemetry
      - Will not acknowledge spacecraft time/attitude commands
      - Will not acknowledge good “body” packets of file upload
  - Global
    - Generate responses to the commands outlined above

Forward Work

- **Coding progress**
  - **ITC is 50% complete**
  - **CTS coding is 80% complete**
  - **LCS coding is 20% complete**
  - **LCM coding will start in October 2004**
- **Inter-task communication between tasks on an SIU was demonstrated in July 2004**
  - **Full inter-task communication between SIU and EPU will be demonstrated November 2004**
- **Coding and unit testing on this system complete 11/30/04**

GLAST Large Area Telescope

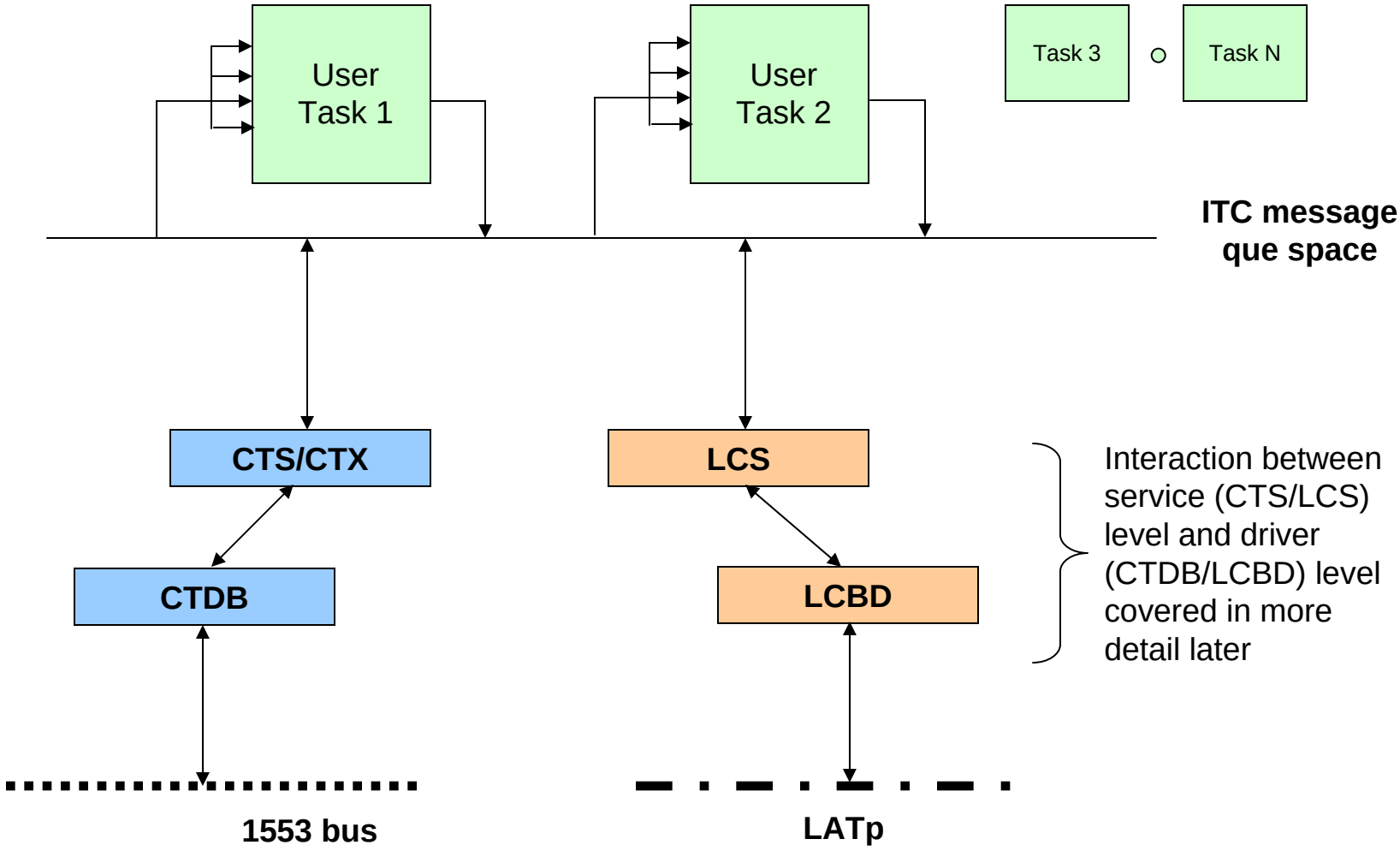
Backup

ITC, CTS, and LCS Package Layout and Configuration Management

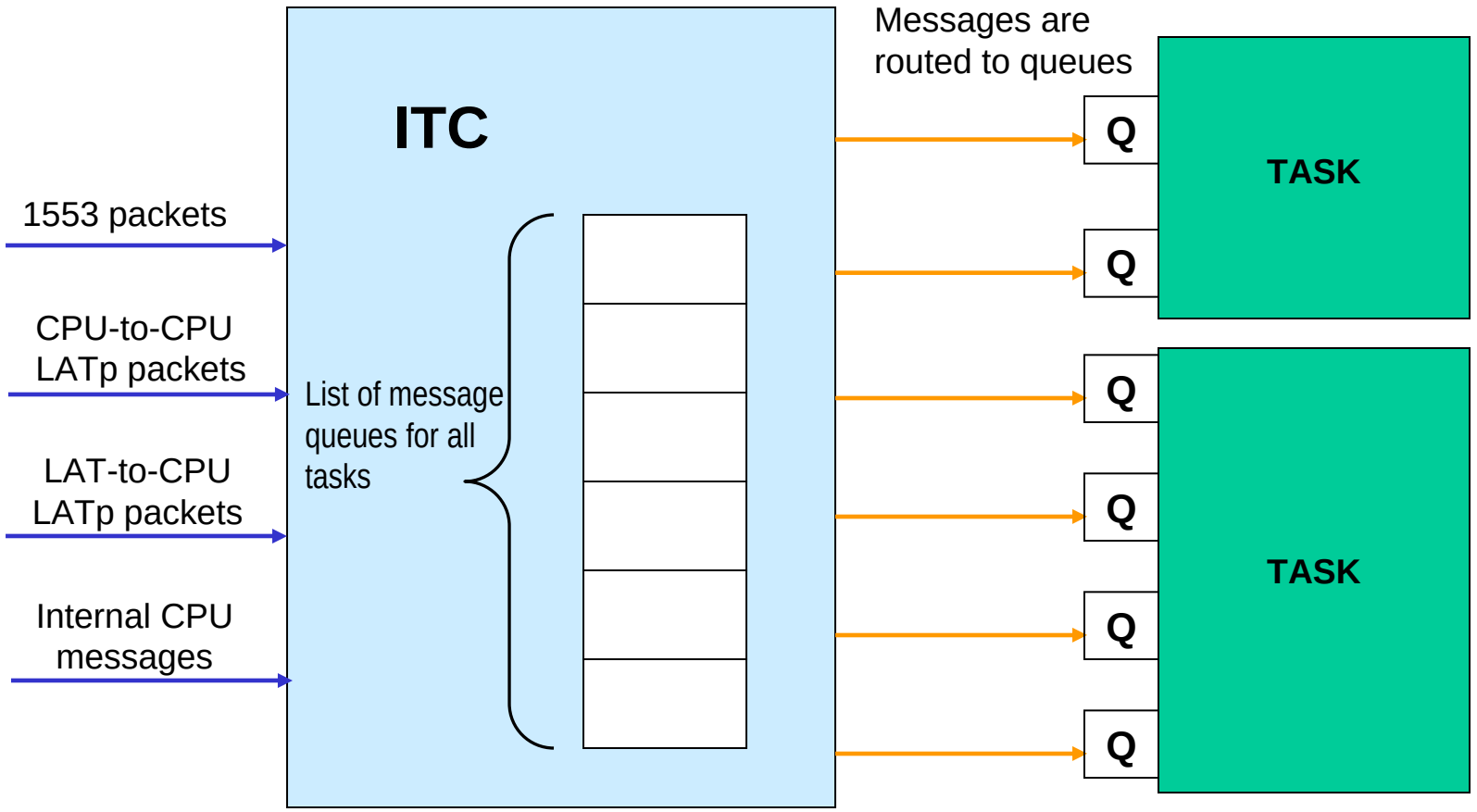
- ITC directly uses the following packages/constituents:
  - CCSDS, CMX, MSG, PBS, VXW
- CTS uses:
  - CMX, CTDB, ITC, MSG, PBS, VXW
- LCS uses:
  - LCBD, CMX, ITC, MSG, PBS, VXW
- LCM uses
  - LCBD, CMX, ITC, MSG, PBS, VXW

- The dependency tree, constituent list, command and telemetry list, and other configuration management information related to these packages are published dynamically on the FSW Web site at
  - http://www.slac.stanford.edu/exp/glast/flight/web/a_pnp/Pack_ITC.shtml
  - http://www.slac.stanford.edu/exp/glast/flight/web/a_pnp/Pack_CTS.shtml
  - http://www.slac.stanford.edu/exp/glast/flight/web/a_pnp/Pack_LCS.shtml

The Inter-task Communications System

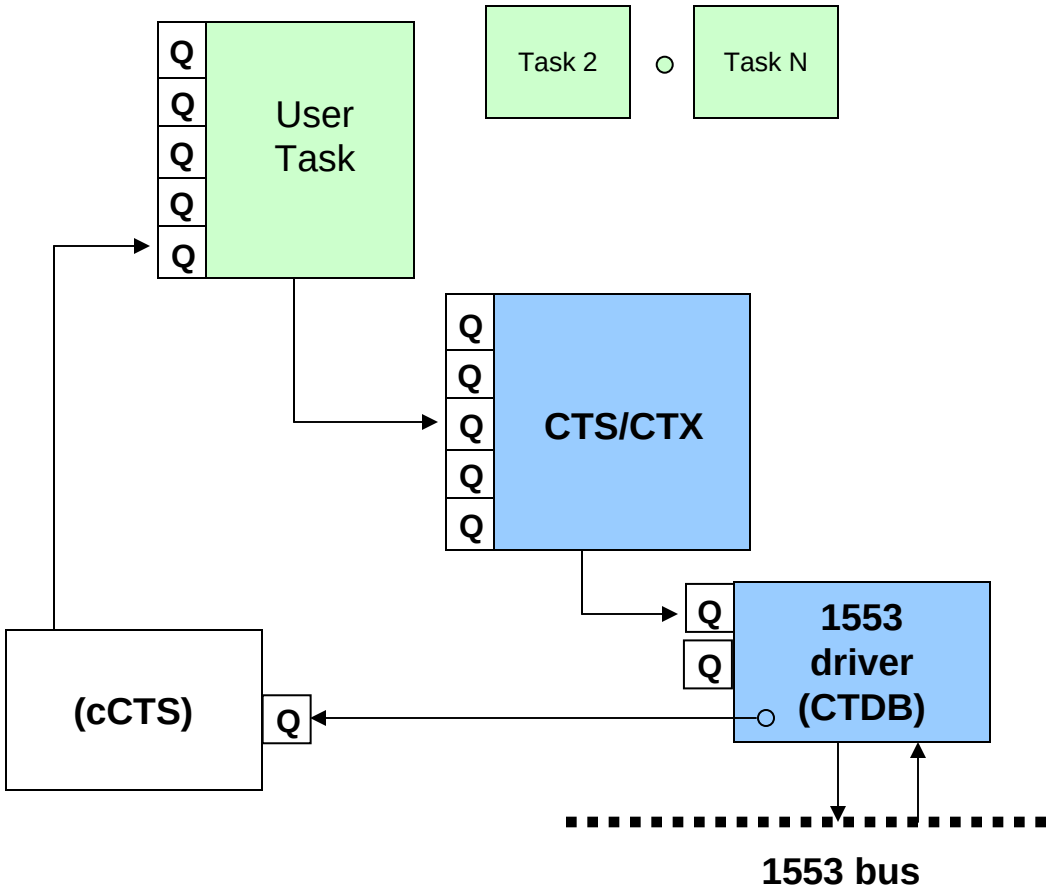


ITC: Architecture (cont'd)



Command and Telemetry/1553 Service: Architecture

- **The Command and Telemetry Service operates between application-level tasks and the 1553 driver.**
  - **Allows the inter-task communications system (and hence user applications) to be hardware blind.**
  - **Outbound messages pass through the C&T Service, on to the 1553 driver, and over the interface to the Spacecraft**
  - **Inbound messages are caught by the cCTS task before being passed to the appropriate queue on the user task**



LCS: Architecture

- **The LCB Service operates between application-level tasks and the LCB driver.**
  - **Outbound messages pass through the LCB Service, on to the LCB driver, and then on to other CPUs or to registers on the hardware**
  - **The LCB driver places incoming messages directly on user task queues**
    - **LCB message headers contain a flag to indicate which ITC protocol to use**
    - **Different approach than the C&T Service**