

12K Support Training

Agenda

- **12K Product Overview**
- **System Architecture**
- **Forwarding Architecture**
- **Services and Applications**
- **Troubleshooting**

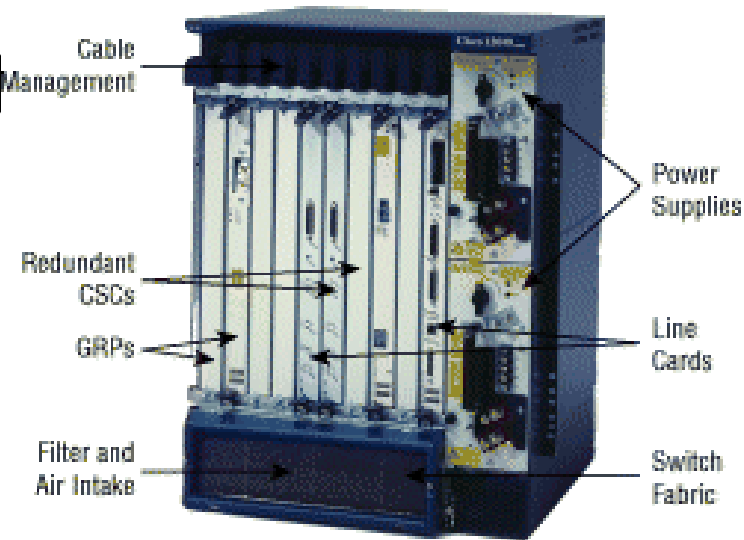
12K Architecture Overview

- **Fully distributed, multi-gigabit IP Router**
 - GRP provides routing and control services**
 - Line cards perform IP forwarding**
- **Advanced QoS capabilities**
- **Bandwidth scalable (OC12, OC48, OC192)**

Cisco 12008 Product Highlights

Cisco.com

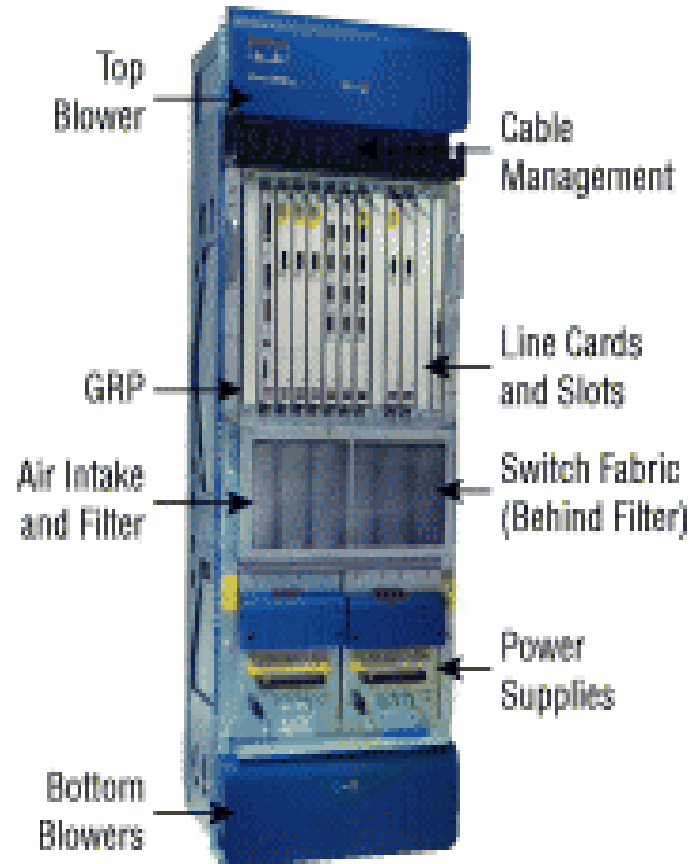
- **Crossbar switch fabric architecture**
- **8 slot card cage (7 for interfaces)**
- **Components:**
 - Switch Fabric Cards (SFC)**
 - Clock and Scheduler Cards (CSC)**
 - Route Processor (RP)**
 - Line Cards (LCs)**



Cisco 12012 Product Highlights

Cisco.com

- **Crossbar switch fabric architecture**
- **12 slot card cage (11 for interfaces)**
- **Components:**
 - Switch Fabric Cards (SFC)**
 - Clock and Scheduler Cards (CSC)**
 - Route Processor (RP)**
 - Line Cards (LCs)**



Cisco 12016 Product Highlights

Cisco.com

- **Switching performance**
16 Slot System, 2.5Gbps switching capacity/slot – can support 10Gb LCs if fabric is upgraded
- **Increased number of linecards**
- **Configuration**
 - 2 Interface Shelves
 - 16 slots
 - 1 Fabric Shelf, with 5 slots
 - 2 Alarm cards – 1 top shelf, 1 bottom shelf



Cisco 12416 - Product Overview

Cisco.com

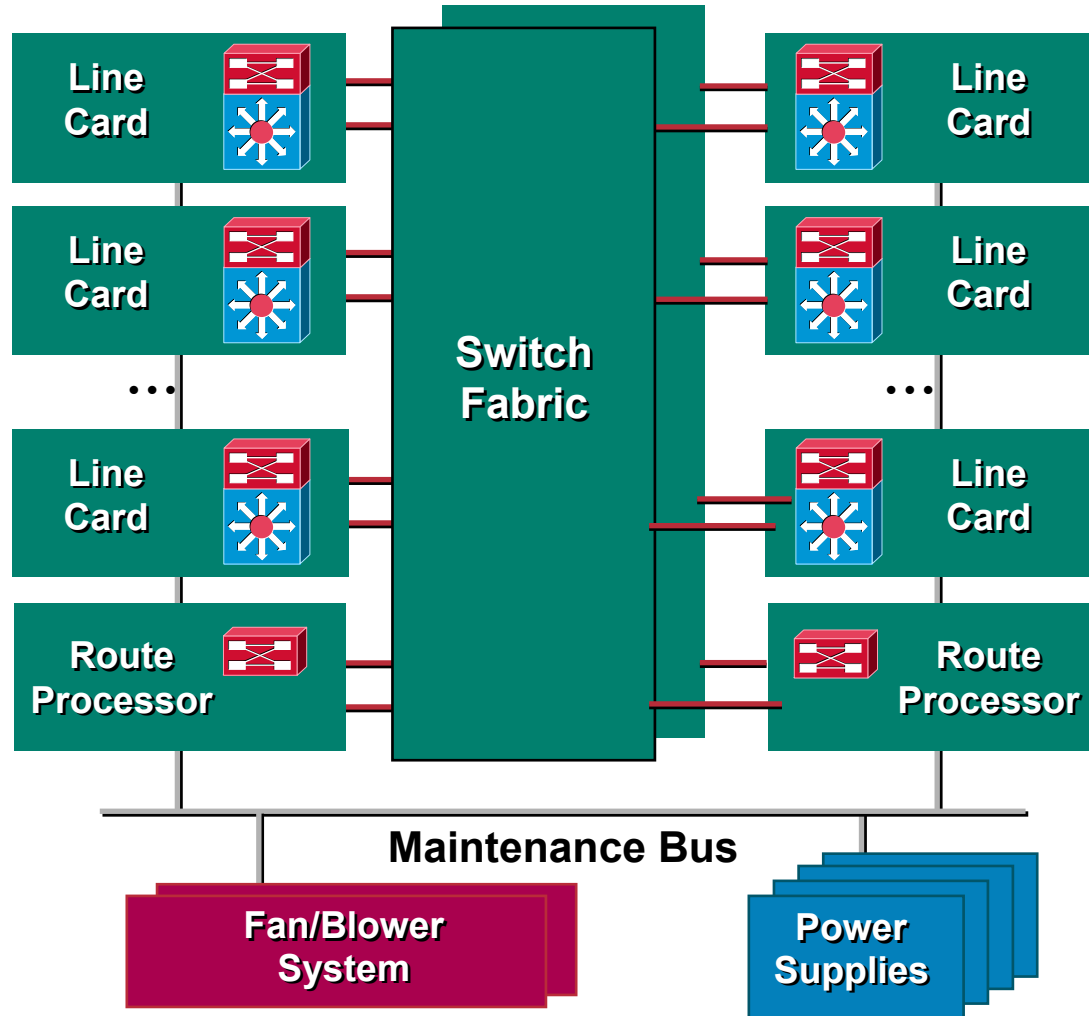


- **Switching performance**
16 Slot System each with 10Gbps switching capacity/slot
- **Supports 10G linecards**
Support for existing GSR line cards
Slots are wider to accommodate 10 Gb LCs
- **Configuration**
2 interface shelves
16 slots
1 fabric shelf, with 5 slots
2 Alarm cards – 1 top shelf, 1 bottom shelf

System Components

- **Route Processor**
- **Switching fabric**
- **Line cards**
- **Power/Environmental Subsystems**
- **Maintenance BUS**

GSR Architecture - Components

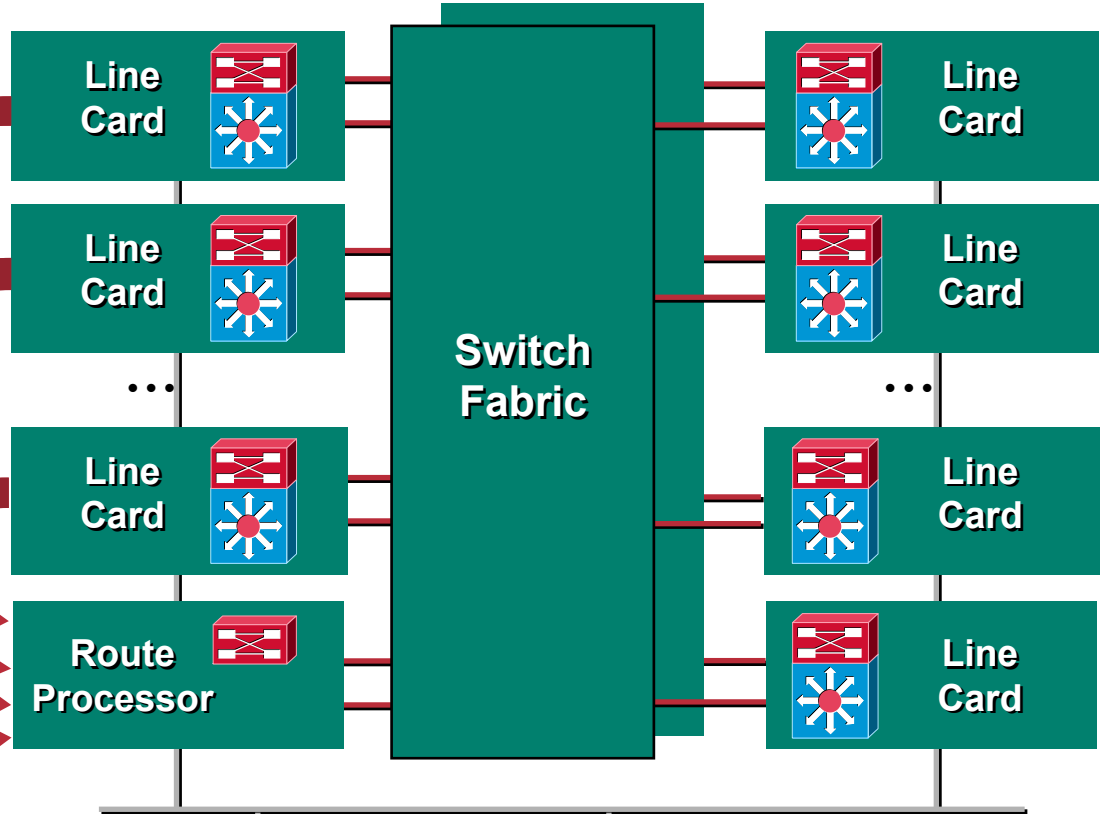


Route Processor

- **Boots and manages line cards**
- **Provides and coordinates routing services**
- **Builds, distributes, and maintains FIB**
Adjacency table, FIB table, MPLS label table
- **Provides out-of-band console/aux ports**
- **Provides intelligence behind system monitoring and access**

GRP—System Monitor/Controller

Routing Protocol Updates
Process-level Traffic
System health monitoring
Interface Status Msgs
Statistics



Temperature,
Voltage,
Current Monitoring

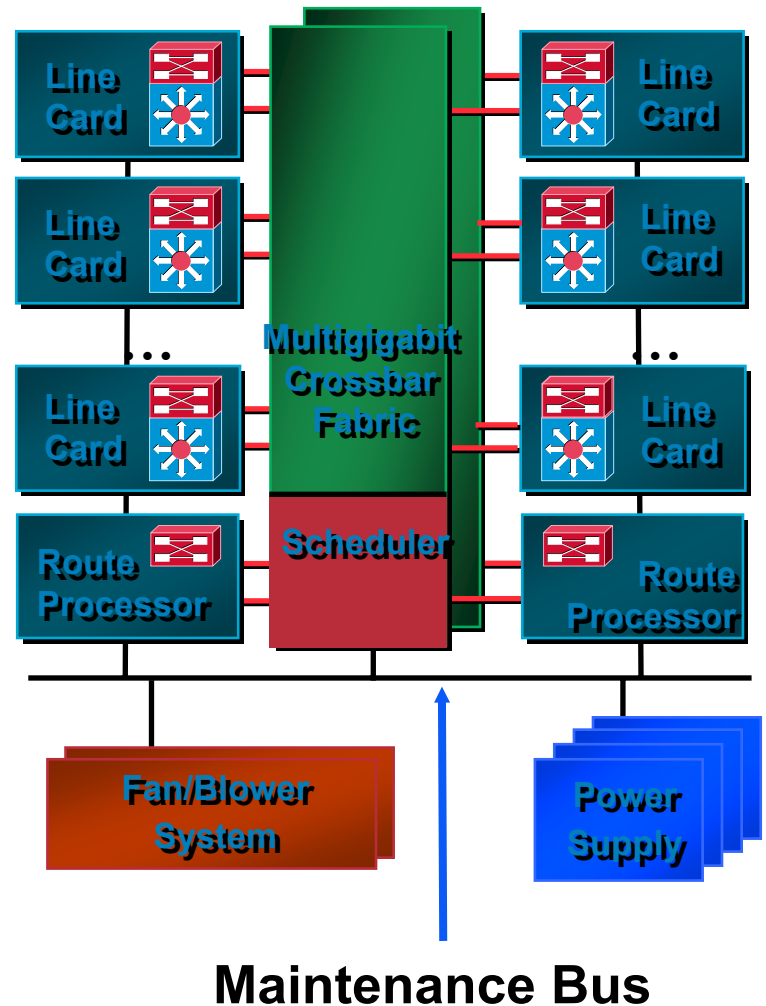
Maintenance Bus

Line Cards

- **Perform all packet switching**
- **Statistics collection and reporting**
- **Run IOS**
- **Six different forwarding architectures**

MBUS

- Out-of-band communications channel to linecards
- 1 Mbps - 2 wire serial interface
- Based on Controller Area Network (CAN) 2.0 Spec. (ISO 11898)
<http://www.can-cia.org/can/>
- A daughter card on each linecard having it's own CPU w/ integrated CAN controller, A/D converter and other peripherals, dual CAN interface, SRAM, Flash and Serial EEPROM.
CSCs and BusBoard can proxy and/or multiplex MBUS signals for power supplies
- Control pins reach into LED, Serial ID EEPROM, DC/DC power converter, clock select FPGA, temp sensor, voltage sensor
Very large set of functions
- ENG-6773 and 9327



MBUS Functions

- **Power and boot LC**
- **Device Discovery**
- **GRP arbitration**
- **OIR management**
- **Environmental monitoring**
- **Diagnostics download**
- **LC console access**
 - Via “attach” command
- **Logging**

Alarm Cards

- **LED pull-out for fabric cards**
- **External alarm connection**
- **Power conversion/supply for 5v MBUS power plane**

On the 12008, this functionality is on the CSC.

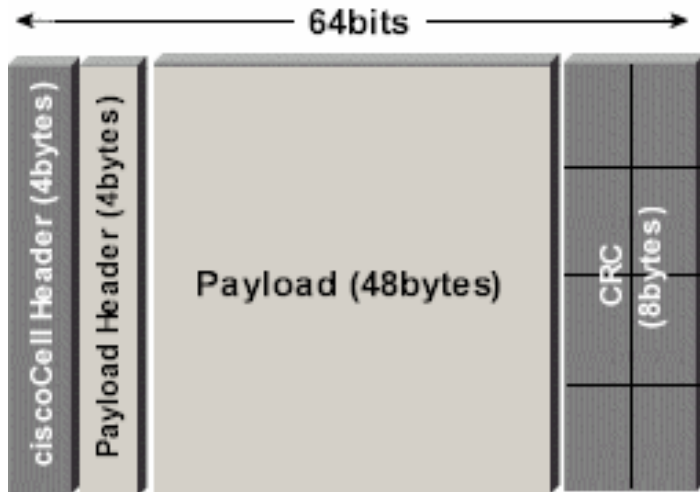
Switch Fabric - Overview

- **Provides the data path connecting the LCs and the GRP**
- **Active CSC card provides the master clock for the system**
- **Everything traverses fabric in Cisco cell.**
 - **Data is 8B/10B encoded**
- **Two components**
 - **Clock & Scheduler Cards (CSC)**
 - **Switch Fabric Cards (SFC)**

ciscoCell

- Packet are chopped into ciscoCells before they are sent across the switching fabric.
- A ciscoCell is 64bytes of data consisting of 48bytes of IP payload and 8bytes of header and 8bytes of CRC.

ciscoCell General Format



ciscoCell Detail Format

ciscoCell Header		Output Mask		Payload Header (always send first)			
byte 0	byte 1	byte 2	Payload 0	byte 3	byte 4	byte 5	byte 6
Payload 1							
Payload 2							
Payload 3							
Payload 4							
Payload 5							

Clock Scheduler Card (CSC)

- **Scheduler (SCA)**

Handles scheduling requests and issues grants to access the crossbar switching fabric

- **Cross-bar (XBAR)**

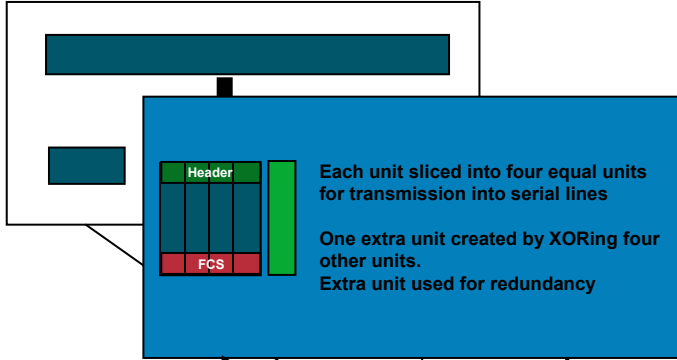
Sets the fabric lines for transmissions following the scheduling decision

Fabric Redundancy

- **Each fabric card provides a slice of the Cisco cell data path**
- **Up to 5 data paths are available – for up to 4+1 redundancy**
- **The 5th data path carries an XOR of other streams**
Used for recovery of a errored stream
- **Grants travel exclusively between the LC and the active CSC**
Never traversing the SFC cards

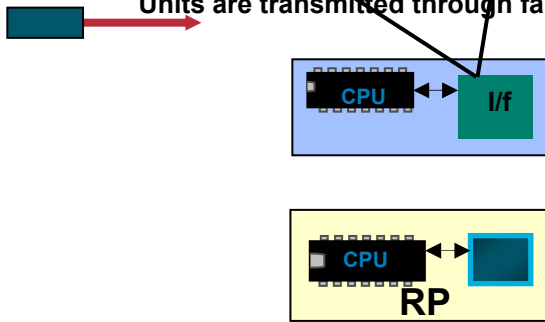
Cross Bar Data Path

Packet sliced into equal sized fixed units for transmission into fabric

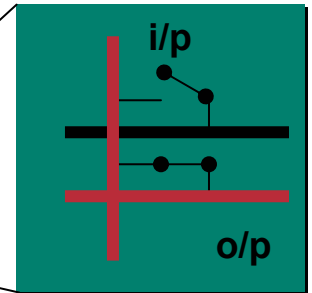
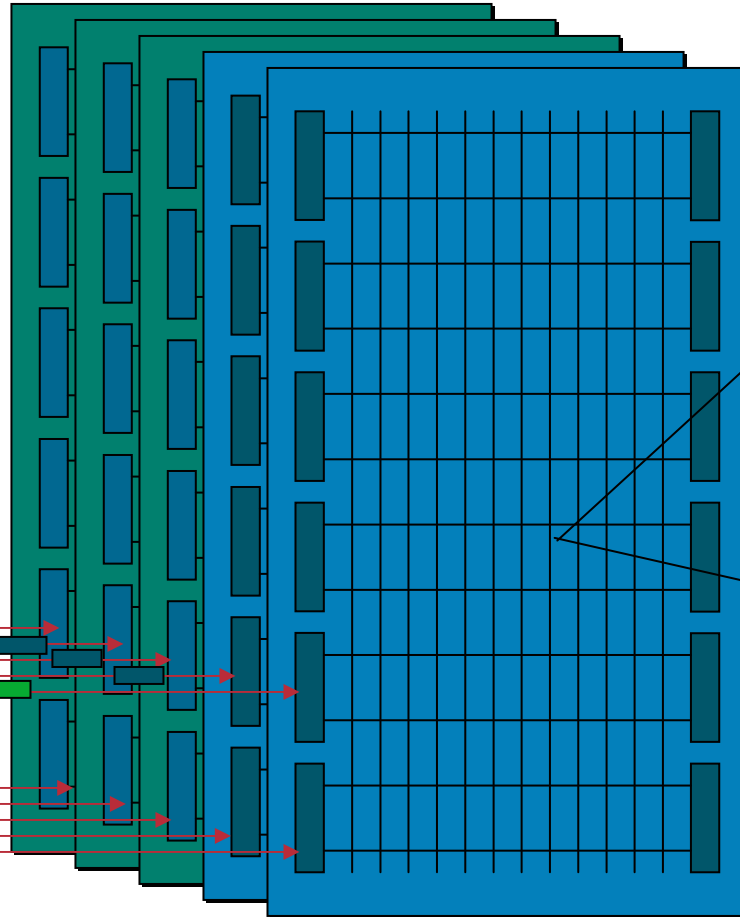


Data Packet

Units are transmitted through fabric



1.25Gbps per trace



Animated

Scheduling Algorithm (“ESLIP”)

- **Request**

Each input LC makes request to output highest priority queued cell (unicast or multicast)

- **Grant**

Each destination LC grants the request to the highest priority request

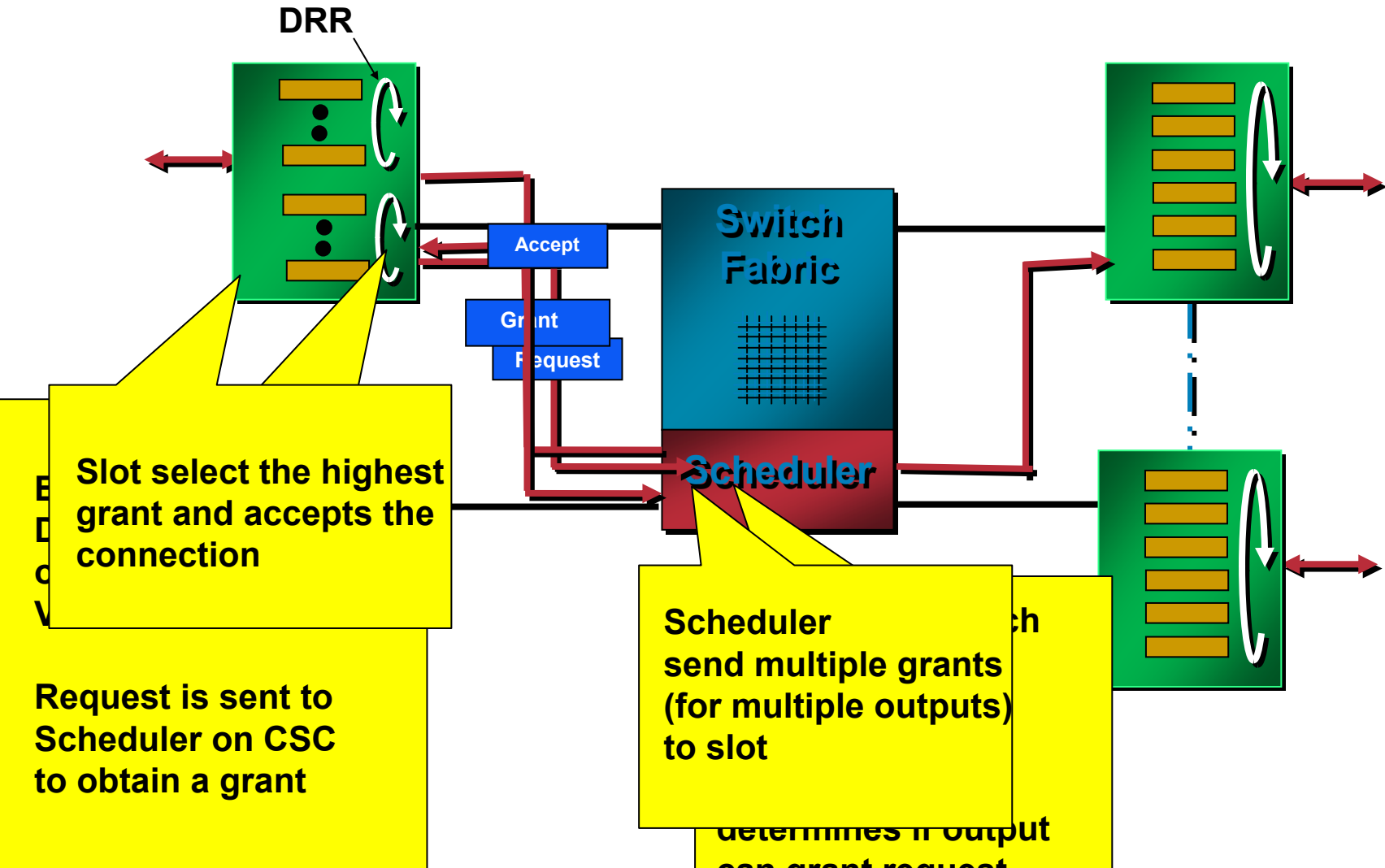
- **Accept**

Each input LC selects the highest grant

- **Transmit**

XBAR set and cells transmitted

ESLIP Illustrated



Bootup process

Startup/Boot Process

- **Initial Power On**
- **GRP Boot Process**
- **Clock Scheduler Boot Process**
- **Line Card Boot Process**
- **Switch Fabric Boot Process**
- **Fabric Initialization**
- **IOS Download**

Initial Power On

- **When the chassis is powered on, the Mbus module on each card is powered on.**
- **After the Mbus module powers on its processor it boots from a module on EEPROM.**
- **Card power up order varies depending on type.**

GRP Boot Process

- **Mbus module powers first**
- **Board logic starts, image begins booting and Mbus code is loaded to the Mbus module.**
- **The CPU, Tiger ASIC, CSAR ASIC and FIA ASICs are then issued power for startup.**
- **GRP arbitration process is executed using the Mbus.**
- **Master GRP instructs Line Cards and Switch Fabric Cards to power on.**
- **GRP waits for Line Cards to power and finish booting.**

Switch Fabric Card Startup/Boot

- **Master GRP instructs each SFC Mbus module to power on at the same time the Line Card Mbus modules are told.**
- **SFC obtains clock the same way each LC does.**
- **The SLI ASICs and XBAR initialize and power up.**
- **SFC Mbus code is downloaded from the GRP.**
- **All cards are now powered on but not usable.**

Line Card Startup/Boot

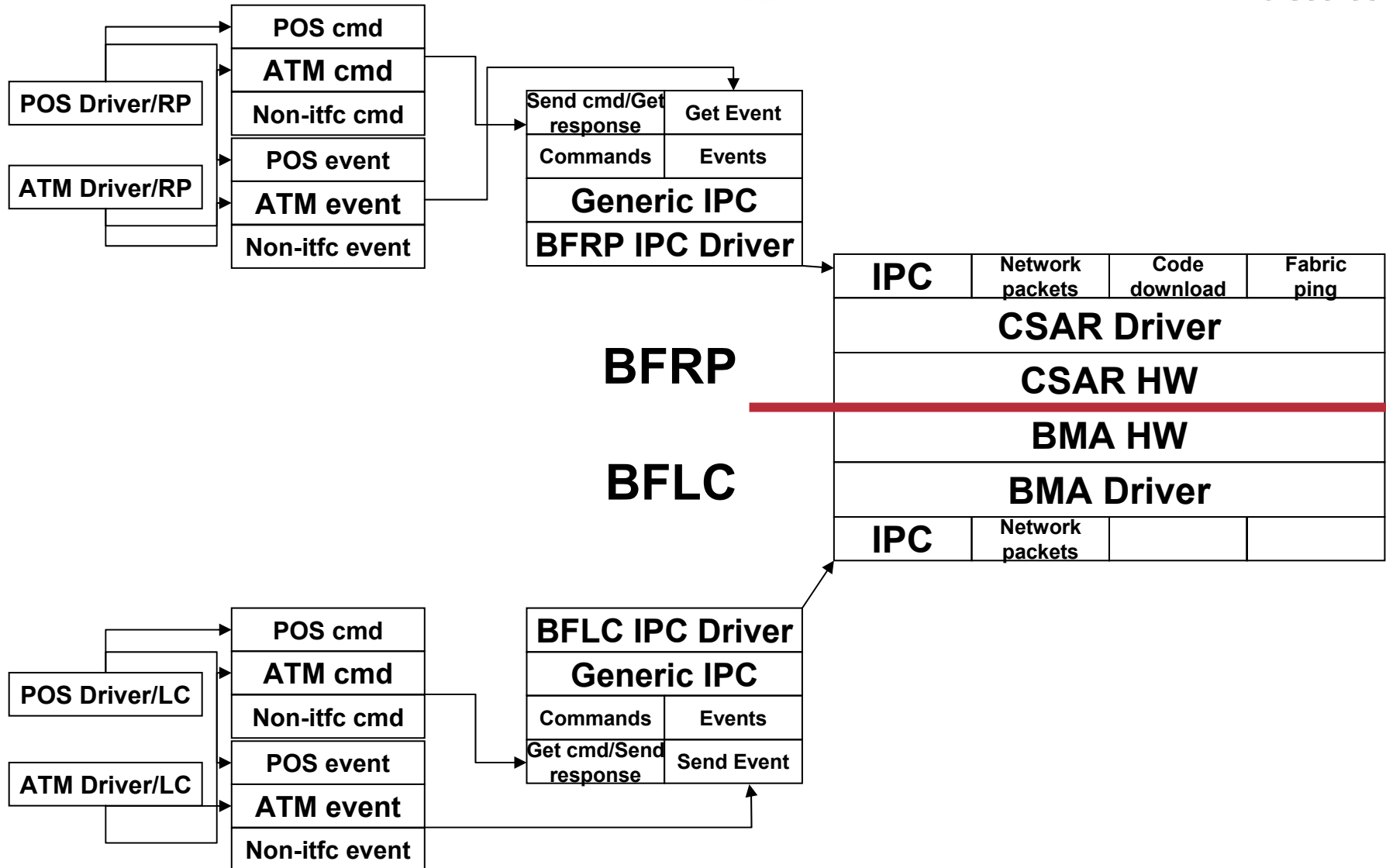
- **Each LC Mbus module powers up after being told to do so by the GRP.**
- **Clock selection takes place.**
- **The Line Card R5000 is powered on and boots.**
- **Mbus module code is loaded.**
- **The Line Cards CPU notifies the GRP it has booted.**
- **Switch Fabric access is not available yet.**

Line Card IOS Downloads

- **The Line Card may already have enough code in its flash to become operational on the Switch Fabric, or it may require an Mbus download.**
- **Only enough code for the Line Card to become operational on the fabric will be loaded using the Mbus.**
- **Once all cards are operational on the fabric, the fabric is initialized and the main IOS software is downloaded.**

IPC Services

Generic IPC Usage Heirarchy



IPC Overview

- **The GSR is a distributed multiprocessor system. The processors communicate via IPC ... an essential architectural service.**

Hierarchical organization of zones, seats and ports.

Simplex communications channels.

Zone mastership arbitration and locally unique namespace maintenance.

Multiple message types ... command messages, raw data, RPCs identifiable by header. Flags in message headers discriminate ACK/NAK, RPC Reply etc.

Multiple structures cluster around IPC ... message tables, RPC tables, cache/queues, emergency message cache/queue

Interrupt or Process level message handling.

Unicast or Multicast delivery (though multicast is rarely if ever used).

Behavior/rate control is up to the client

- **IPC has a reliable (acknowledged) and unreliable mode of transport (with or without sequence number or notification). The application uses an appropriate method.**

e.g. ... CEF uses reliable transport ... MDFS used to use unreliable (but with a sequence number) but changed. Nobody uses unreliable with notification.

Ports can use different receive methods - callbacks or queues to service messages.

- **Applications (clients) can build their own queue structures and feed the IPC queue/cache as well as choose to block or not until an ACK or imbedded response is received.**
 - e.g. ... CEF uses a multi-priority queue and it's own cache in front of the IPC queue (controlled by "ip cef linecard ipc memory") ... it's got it's own message handling routines defined in the same registry as direct IPC interrupt or process level message handling.
 - Many (most) applications use the CEF packaging (XDR) message types and queues as an interface to IPC.
 - e.g. ... route-map updates and acl updates to linecards
- **Applications are also responsible for being "well-behaved".**
- **Utility applications like slavelog and slavecore use IPC directly.**