

Cisco 2600 Series Router Architecture

Document ID: 23852

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Block Diagram

- CPU
- WIC Slots
- NM Expansion Slot
- LAN 0 / LAN 1
- AIM Socket
- Host PCI Bridge
- System Bus
- CPU Bus
- Memory
- Power Supply

Memory Details

- BootROM
- Flash
- DRAM
- NVRAM
- Boot Sequence

Basic Operation

- Memory Structure Overview – Buffer Allocation
- Process–Switching
- Fast–Switching
- CEF–Switching

Performance

Related Information

Introduction

This document is an overview of the hardware and software architecture of the 2600 Series Routers.

Note: You must be a registered user and you must be logged in in order to use the troubleshooting tools described in this document.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco 2610
- Cisco 2610XM
- Cisco 2611
- Cisco 2611XM
- Cisco 2612
- Cisco 2613
- Cisco 2620
- Cisco 2620XM
- Cisco 2621
- Cisco 2621XM
- Cisco 2650
- Cisco 2650XM
- Cisco 2651
- Cisco 2651XM
- Cisco 2691

Refer to Cisco 2600–DC Series Modular Access Platform for more information.

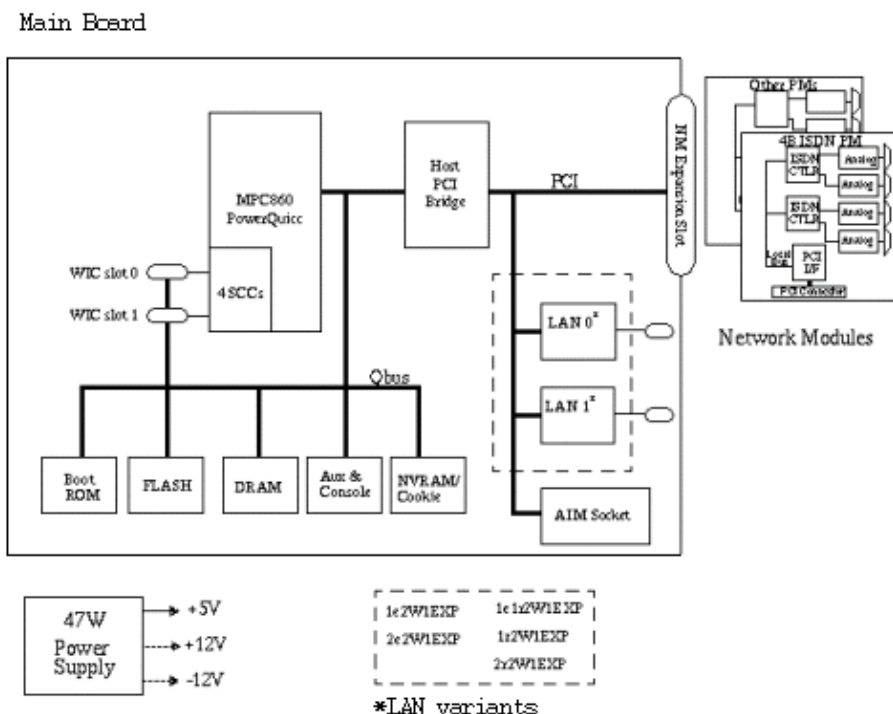
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Block Diagram

This diagram applies to the 2610–2651XM.



CPU

The type is Motorola PowerQUICC MPC860; the 2691 uses the RM7061A.

As the function of the CPU, it executes instructions coded in the operating system and its subsystems to perform the basic operations necessary in order to accomplish the functionality of the router, for example, all of the routing functions, network module high-level control, and system initialization.

WIC Slots

There are two fixed WAN Interface Card (WIC) slots, while there are three on the 2691, to support fully-functional WICs, some of which are compatible across the different modular access router platforms such as the Cisco 1600, 1700, 3600 and 3700 Series routers. Refer to Overview of Cisco Interface Cards for Cisco Access Routers and the Software Advisor (registered customers only) for information on platform and Cisco IOS® software support.

The communications controller has four on-chip Serial Communication Channels (SCCs), while there are six for the 2691, with dedicated individual links to the WIC slots.

NM Expansion Slot

There is one expansion slot to install a WAN or LAN network module (NM). Some NMs can be used with the Cisco 3600 Series routers. Refer to Overview of Cisco Network Modules for Cisco Access Routers and the Software Advisor (registered customers only) for information on platform and Cisco IOS software support.

Online insertion and removal (OIR) is not supported.

LAN 0 / LAN 1

These are the integrated LAN controllers on the motherboard. There are three different types:

- Ethernet
- Fast Ethernet
- Token Ring

Availability depends on the specific model of the 2600 router.

AIM Socket

This socket can accommodate the Advanced Interface Module (AIM) cards. This is an internal 100-pin socket, which allows functions that do not require an external connection such as compression, encryption, and so forth. The 2691 has two AIM sockets.

Host PCI Bridge

This is the bridge interface between the CPU bus and the system bus, PCI bus, where the Network Modules and other interface boards are connected.

System Bus

This is used for communication between the CPU Board and the interface boards. On the 2600 platform, this is a PCI bus.

CPU Bus

This is used by the CPU in order to access the various components of the system and to transfer instructions and data to or from specified memory addresses.

Memory

The memory is used in various forms for several storage purposes such as to store the operating system (Cisco IOS software), the configuration, the bootstrap, packets, and so forth. You can find different kinds of memory on the 2600 platforms such as Flash, Dynamic RAM (DRAM), nonvolatile RAM (NVRAM), and BootROM. See the Memory Details section for more information..

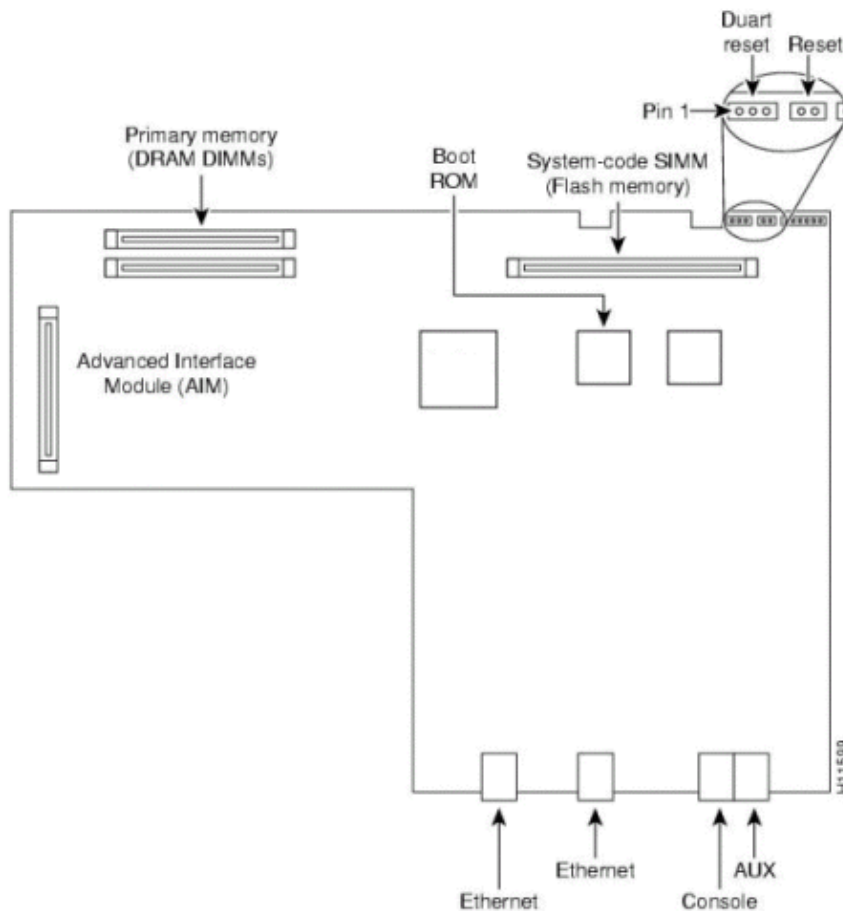
Power Supply

Refer to Cisco 2600–DC Series Modular Access Platform for the power supply specifications.

The 2600 series can also operate from Redundant Power Supply (RPS). The PWR600–AC–RPS is a Cisco RPS for the 2500, 2600, 3600, and 4000 series routers.

Memory Details

There are four kinds of memory in the 2600 series router:



BootROM

BootROM is used for permanently the storage of the startup diagnostic code (ROM Monitor).

The main task for the BootROM is to perform some hardware diagnostics during bootup on the router, Power On Self Test (POST), and to load the Cisco IOS software from the Flash to the Memory.

The BootROM is not erasable; it is socketed so it can be replaced.

On the 2691, the boot code and non-volatile data are loaded in the Flash device described here.

Flash

Flash is used for permanent storage of a full Cisco IOS software image in compressed form. On the 2691, the Flash is also used to store the boot image and the NVRAM data.

- The 2610 and 2651 have one Flash SIMM socket that supports the Cisco 80-pin Flash SIMMs (4 MB, 8 MB, and 16 MB). 8 MB and 16 MB Flash SIMMs are dual bank, which can be partitioned into two banks.
- The 2600XM has 16 MB Flash soldered to the motherboard plus one Flash SIMM socket.
- The 2691 has the same Flash memory architecture as the 3700 series. There is a compact Flash card on the motherboard plus an external compact Flash slot. The compact flash can use the DOS file system, but it must be formatted on the 2691 or 3700 in order to write the proper code in the boot sectors. Once the compact flash is formatted in the 3700, you can use a 2691, 3631, 3700, 7400, 7300, or PC in order to write to the compact Flash.

The 2600 series uses a Class B file system. Note that when you replace the Flash SIMM, you must use the ROMmon in order to copy a Cisco IOS software image onto that SIMM.

DRAM

DRAM is used at run time for executable Cisco IOS software, and its subsystems, routing tables, Fast Switching cache, running configuration, packets, and so forth.

The 2600 has two DRAM sockets and uses **non-parity** DRAM.

The 2610–2621 uses 100-pin EDO DRAM DIMMs. The 265x and the 2600XM use 100-pin SDRAM DIMMs, which are not compatible with the current 2610–2621 DIMMs. The 2691 uses 168-pin SDRAM DIMMs (two sockets).

DRAM is logically divided in Main Processor Memory and Shared Input/Output (I/O) Memory. Shared I/O Memory is shared among interfaces for temporary storage of packets. I/O memory is the memory that is used for temporary storage of packets for both the input and output packets. Process memory is used to store the Cisco IOS software code, the routing tables, and the system buffers. The 2600 can reallocate the split between processor and I/O memory, as can the 3600 series, with the **memory-size iomem <percent>** command.

NVRAM

NVRAM is used for writable permanent storage of the startup configuration. It is an EPROM, except in the 2691, where the startup configuration is stored in the same Flash device where the boot code is loaded.

Boot Sequence

Complete these steps:

1. After you power on the router, the ROM monitor starts first. ROMMON/BOOTSTRAP functions are important at router boot, and complete these operations at bootup:
 - ◆ Configure power-on register settings These settings are of the Control Registers of the processor and of other devices such as Dual Universal Asynchronous Receiver Transmitter (DUART) for console access, as well as the configuration register.
 - ◆ Perform power-on diagnostics Tests are performed on NVRAM and DRAM, writing and reading various data patterns.
 - ◆ Initialize the hardware Initialization of the interrupt vector and other hardware is performed, and memory, for example, DRAM, SRAM, and so forth, is sized.
 - ◆ Initialize software structures Initialization of the NVRAM data structure occurs so that information about the boot sequence, stack trace, and environment variables can be read. Also, information about accessible devices is collected in the initial device table.
2. Next, the ROMmon looks for the Cisco IOS software image in the Flash. Since the 2600 router has no separate BOOT-HELPER image, it needs to have a valid image in the Flash.

If the router does not find a valid image in the Flash, it is not able to come up. Use the **tftpdnld** or **xmodem** command if there is no valid image in Flash. Refer to ROMmon Recovery for the Cisco 2600 Series Router and the VG200 for more information.

Even if you want to boot the router with the Trivial File Transfer Protocol (TFTP), you need a valid image in the Flash in order to boot that image first, and to use that image as a boot-helper image in order to initialize the system, and bring up the interfaces in order to load the main image from the TFTP server.

3. After you find the image, the router decompresses it and loads it into the DRAM. Then the Cisco IOS software image starts to run.

Cisco IOS software performs important functions during bootup, such as:

- ◆ Recognition and analysis of interfaces and other hardware
- ◆ Setup of proper data structures such as Interface Descriptor Blocks (IDBs)
- ◆ Allocation of buffers
- ◆ Reading the configuration from NVRAM to RAM (running-config) and the configuration of the system

This is an example of a boot sequence from a 2600 router:

```
System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
C2600 platform with 65536 Kbytes of main memory

program load complete, entry point: 0x80008000, size: 0x43b7fc

Self decompressing the image:
#####
#####
#####
#####
##### [OK]
```

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.1(8), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Tue 17-Apr-01 04:55 by kellythw
Image text-base: 0x80008088, data-base: 0x8080853C

cisco 2611 (MPC860) processor (revision 0x203) with 56320K/9216K bytes of memory.
Processor board ID JAD05020BV5 (1587666027)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 Ethernet/IEEE 802.3 interface(s)
2 Serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)

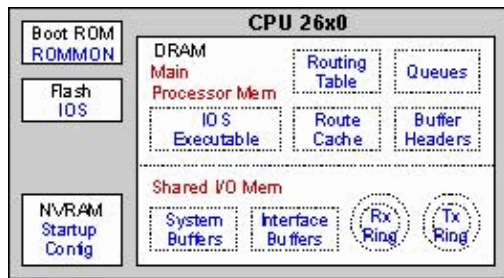
Press RETURN to get started!
00:00:09: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
00:00:09: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
00:00:09: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
00:00:09: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
00:00:10: %SYS-5-CONFIG_I: Configured from memory by console
00:00:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0,
changed state to up
00:00:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/1,
changed state to up
00:00:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
00:00:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1,
changed state to up
00:00:13: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.1(8), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Tue 17-Apr-01 04:55 by kellythw

router>

Basic Operation

Memory Structure Overview – Buffer Allocation

Within low-end and mid-range systems, packet flow through a router, regardless of the switching mechanism, is closely related to buffer usage. This section provides a quick overview in order to ensure that you are familiar with the buffer and memory structures used for process- and fast-switched packets.



There are two basic buffer pools available:

- public buffers (system buffer)
- private buffers (interface buffer)

Some interface processors create pools of private buffers when they initialize, some do not.

Private buffer pools can be viewed with the **show buffers interface** command.

Public buffer pools are created by Cisco IOS software, and they are used to process switch packets, or by interfaces that either run out of private buffers or do not support the private buffers function.

When a packet first arrives on an interface, it is placed in a buffer on the receive ring. The interface processor then tries to replace this used buffer with a free buffer, either from its private pool, or if this is not possible, from a buffer from the public pool.

If the packet is to be process-switched, then ownership of that buffer passes from the interface processor to the CPU. If the packet is to be fast-switched, then ownership passes either to the output queue, or to the outbound transmit ring.

It is important to note that in low-end and mid-range systems, packets are never copied from buffer to buffer. Only ownership of the buffer changes, with the use of pointers.

Once the packet is transmitted, the buffer returns to its original owner.

Interface buffers are particle-based. This means that you use 1524 bytes of long particles instead of continuous buffers. You only use particles for interfaces with a large maximum transmission unit (MTU) in order to avoid overhead. You cannot handle particles at process level, so packets are reassembled when transferred to the small and medium system buffers.

Interfaces have a private particles pool. When they are short of private buffers, they can fall back to a public particle pool that corresponds to their buffer size.

Process-Switching

Process-switched refers to the fact that the CPU is directly involved in the decision process required to forward the packet.

After a packet arrives on an inbound interface, the interface driver must first copy that packet into a packet buffer in shared memory. This buffer can be pulled from either a public or a private pool, and is done without the signal of an interrupt to the CPU.

The interface driver next determines what type of Layer 3 protocol is encapsulated in the packet. This information is also buffered.

Once the interface driver has buffered the packet and identified the Layer 3 protocol, it then generates an interrupt to the processor, which indicates that a packet waits in the input queue for processing.

Once the processor receives the interrupt generated by the interface driver, it assumes ownership of the packet buffer and determines which process must be called in order to handle this packet, and then schedules that process to run.

At this point, there is a period of idle time for the packet, as it waits for the called process to be run. Exactly how much idle time depends on the number of outstanding processes that wait to run, the number of additional packets that wait to be forwarded, and so forth.

When the process that handles the required packet type finally runs, it does a route table lookup in order to determine out of which interface this packet should be forwarded. If it is determined that this packet is to be forwarded, then a new Layer 2 header is added to the packet and it is placed on the relevant output queue.

But, if the packet is destined for the router itself, then it is requeued for additional processing. The process that handles the destination output queue then places the packet onto the interface transmit ring. The interface driver identifies that there is a packet in the transmit ring that waits to be sent, and forwards it out onto the physical media. The interface driver then signals an interrupt back to the processor and requests that counters be updated and buffers placed back into free pools.

Fast-Switching

Fast-switching relies on the forwarding of Layer 3 packets by referencing a cache of destination Layer 3 addresses, corresponding Layer 2 addresses, and the associated outbound interfaces.

Fast-switching is applicable to all Cisco IOS software platforms, although it does not support all protocols or packet features. For example, TCP header compression requires CPU processing, and some IBM and X.25/LAPB protocols cannot be fast-switched.

For fast-switching paths, after a packet arrives on an inbound interface, the interface driver first copies that packet into a packet buffer in shared memory. This buffer can be pulled from either a public or a private pool, and is done without interruption to the CPU.

The interface driver next determines what type of Layer 3 protocol is encapsulated in the packet. This information is also buffered.

The interface driver then examines the switching cache, in order to determine whether or not an entry exists for the required destination. If a valid entry exists, the new Layer 2 header is copied from the cache and prepended onto the Layer 3 packet. Then, with the information from the cache, the interface driver determines which outbound interface should be used in order to forward this packet.

If the outbound interface already has packets queued in its outbound, the driver adds the new packet to the end of the queue. If the queue is empty, the driver places the new packet directly onto the transmit ring.

After successful transmission onto the physical media, the transmitting interface processor signals a transmit interrupt to the processor, so that counters can be updated, buffers returned, and so forth.

If the cache does not contain a valid entry for the required destination, the interface driver signals a receive interrupt to the processor, and the packet is process-switched, which is described in the previous section. But, in addition to forwarding the packet, the processor uses the results of the forwarding decision in order to populate the fast cache, so that subsequent packets to the same destination can be fast-switched.

CEF–Switching

The fast–switching mechanisms discussed in the previous sections all suffer these drawbacks:

- They are all traffic–driven, in that they depend on the reception of the first packet in order to populate the cache.
- It is possible for caches to grow larger than routing tables, and therefore consume significant amounts of memory.
- Periodic aging of the cache entries can consume large amounts of CPU time if the cache is large.
- Cache invalidation due to a route–flap relies on process–switching in order to repopulate the cache with valid entries.
- On some platforms, the size of the cache can lead to cache entry churn if there are too many entries for the cache to support.
- They are unable to do per–packet load–balancing from an interrupt level.

It is the inherent drawbacks to traditional demand–based caches that led to the development of Cisco Express Forwarding. The two main components of Cisco Express Forwarding are the Forwarding Information Base (FIB) and the adjacency table. Both tables are stored in DRAM memory.

The FIB table is used to make IP destination prefix–based forwarding decisions. It contains a mirror image of the information stored in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next–hop address information based on the information in the IP routing table.

Cisco Express Forwarding also uses adjacency tables in order to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next–hop addresses for all FIB entries. The entries allow the Route/Switch Processor (RSP) to perform fast Layer 2 header rewrites when switching the packet from source interface to destination interface.

The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created, for example, through the Address Resolution Protocol (ARP), a link–layer header for that adjacent node is precomputed and stored in the adjacency table. Once a route is determined, it points to a next hop and corresponding adjacency entry. That entry is then subsequently used for encapsulation during Cisco Express Forwarding switching of packets.

Refer to Cisco Express Forwarding for more information about Cisco Express Forwarding.

Performance

These performance figures are based on the Cisco 2600–DC Series Modular Access Platform:

Platform	Throughput (max, fast–switching)
2610–12	15 K packets per second (pps)
2620/21	25 K pps
2650/51	37 K pps
2610/11XM	20 K pps
2620/21XM	30 K pps
2650/51XM	40 K pps
2691	70 K pps

Related Information

- **Password Recovery for Cisco 2600 Series Routers**
 - **Software Installation and Upgrade Procedure for Cisco 2600 Series Routers**
 - **ROMmon Recovery for the Cisco 2600 Series Router and the VG200**
 - **Maximum Number of Interfaces and Subinterfaces for Cisco IOS Platforms: IDB Limits**
 - **Router Product Support**
 - **Technology Support**
 - **Technical Support & Documentation – Cisco Systems**
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Apr 03, 2007

Document ID: 23852
