Cisco 4000 Series Router Architecture

Document ID: 12758

Interactive: **This document offers customized analysis of your Cisco device.**

Contents

Introduction

This document is an overview of the hardware and software architecture of the Cisco 4x00 Series Routers.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

The information in this document is based on the hardware versions below:

- Cisco 4x00 Series Routers

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live

network, ensure that you understand the potential impact of any command before using it.

Hardware Overview

Chassis

The Cisco 4x00 chassis has a CPU motherboard with three sockets for Network Interface Modules (NIMs), and a single power supply. The difference among platforms is performance due to different processors and memory on CPU.
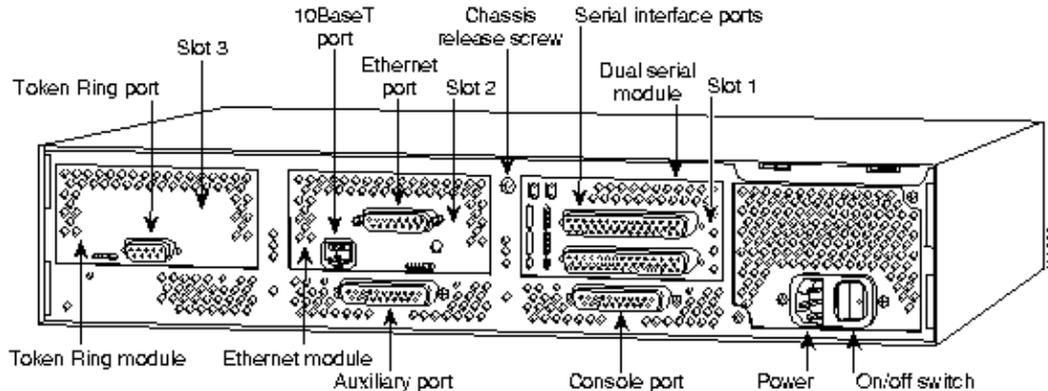
Motherboard Revision

The Cisco 4000 Series comprises the Cisco 4000 and 4000–M, the Cisco 4500 and 4500–M, and the Cisco 4700 and 4700–M. The different versions can be identified by the serial number and the revision version obtained from the **show version** command as shown in the following the table.

Series	show version revision output	Serial Number
4000	Revision A0	440xxxxx
4000–M	Revision B0, C0	445xxxxx
4500	Revision 0x00	450xxxxx
4500–M	Revision B, C, D, E	455xxxxx
4700	Revision B	470xxxxx
4700–M	Revision C, D, E, F	475xxxxx

The examples below come from a 4500–M and a 4000:

```
cisco 4500 (R4K) processor (revision D) with 32768K/8192K bytes of memory.
cisco 4000 (68030) processor (revision 0xA0) with 16384K/4096K bytes of memory.
```

If you have the output of a **show version** command from your Cisco device, you can use to display potential issues and fixes. To use, you must be a registered customer, be logged in, and have JavaScript enabled.

You can use Output Interpreter to display potential issues and fixes. To use Output Interpreter, you must be a registered customer, be logged in, and have JavaScript enabled.

Network Processor Modules

Network Processor Modules (NPMs) are the interface modules of the Cisco 4000 Series. The table below lists the NPMs and their corresponding NIM type codes in hexadecimal and decimal.

An NPM can be identified by checking the NIM type code using the **show controllers** command. For instance, the NPM below is an NP–4T.

```
router#show controllers serial 0
HD unit 0, NIM slot 0, NIM type code 12, NIM version 1
```

NIM Type Code (hex)	NIM Type Code (dec)	Network Processor Module
0x00	0	
0x01	1	
0x02	2	FDDI, single–attached station, NP–1F–S–M
0x03	3	FDDI, dual–attached station, NP–1F–D–MM/NP–1F–D–SS
0x04	4	Ethernet, NP–1E
0x05	5	Token Ring, NP–1R
0x06	6	
0x07	7	Serial, 2–port, NP–2T
0x08	8	
0x09	9	Ethernet, 2–port, NP–2E
0x0A	10	Token, 1–port, NP–1RV2
0x0B	11	Token, 2–port, NP–2R
0x0C	12	Quad Serial, NP–4T
0x0D	13	Multichannel Serial, NP–CE1B/NP–CE1U/NP–CT1
0x0E	14	6–port Ethernet, NP–6E
0x0F	15	ATM
0x10	16	ATM reserved
0x11	17	ATM reserved
0x12	18	ATM reserved
0x13	19	Serial with sync/async ports, NP–2T16S–XX
0x14	20	Fast Ethernet, NP–1FE
0x15	21	High–speed Serial, NP–1HSSI
0x16	22	Ethernet, 2–port full–duplex,

		NP–2E–FDX

The different models in the Cisco 4000 Series Routers support different combinations of network processor modules. The following list addresses router–module and module–module compatibility issues and slot placement considerations:

- A maximum of two high–speed interfaces is supported.
- The original Cisco 4000 and Cisco 4000–M can support only one Fiber Distributed Data Interface (FDDI) module in combination with any two other types of network processor modules (except ATM). Additionally, the 4000 and 4000–M do not accommodate the six–port Ethernet or ATM modules.
- The Cisco 4500, Cisco 4500–M, Cisco 4700, and Cisco 4700–M support all network processor modules except the single–port Ethernet network processor module and the early versions of the single and dual Token Ring, dual Ethernet, and FDDI modules.
- Each of these models supports two FDDI modules: a single ATM interface, and up to three six–port Ethernet modules.
- The Basic Rate Interface (BRI) four–port and eight–port modules, when configured for ISDN Primary Rate Interface (PRI), are incompatible with the CT1/ISDN PRI or E1/ISDN PRI modules.
- If an FDDI module is present, install it in the center slot position for optimum heat dissipation.

For more information on the installation of the Network Processor Modules, see Installing Network Processor Modules in the Cisco 4000 Series.

Impact of Number and Types of Physical Interfaces on Shared Memory Requirements

The standard configuration for shared memory is 4 MB for the Cisco 4000 Series. 4 MB of memory is enough for most configurations with fewer than 24 physical or virtual interfaces. Routers with multiple ISDN BRI network processor modules or with 24 or more physical and virtual interfaces require 8 to 16 MB of shared memory.

To successfully operate an ISDN BRI, ATM, channelized T1/ISDN PRI, or channelized E1/ISDN PRI module when the number of interfaces is fewer than 24, the system requires at least 8 MB of main memory DRAM (16 MB in the original Cisco 4000) and 4 MB of shared memory.

For routers with NP–8B, NP–CT1, and NP–CE1 NIMs with 24 or more ISDN interfaces, a minimum of 8MB of shared DRAM is required. Starting with Cisco IOS® Software Release 11.0 and later, all routers ordered with NP–4B, NP–8B, NP–CT1, and NP–CE1 also require 16MB of main DRAM, in addition to the above shared memory requirement.

The charts below show the amount of shared memory required per NPM, but does not take into account the 24 ISDN interface rule for NP–8B, NP–CT1, and NP–CE1 that supersedes this requirement. When configured in channelized mode, both the NP–CT1 and NP–CE1 require the amount specified below.

4000	Shared memory required per NPM, in MBytes
NP–1E, NP–2E, NP–2T, NP–2R	0.1 MB
NP–2E, NP–2T	0.2 MB
NP–2R, NP–4T, NP–4B	0.4 MB
NP–8B, NP–1P	1.0 MB

4500/4700	Shared memory required per NPM, in Mbytes
NP–1F	2.0 MB
NP–1E	0.2 MB
NP–2E, NP–2T	0.4 MB
NP–2R, NP–4T, NP–4B, NP–4G	0.6 MB
NP–6E	1.2 MB
NP–8B, NP–1P	1.2 MB
NP–8B, NP–1P	2.0 MB
Two NP–1F	3.0 MB
NP–FE	1.7 MB
NP–2T16S	0.6 MB
`NP–HSSI	1.0 MB

CPU and Block Diagram
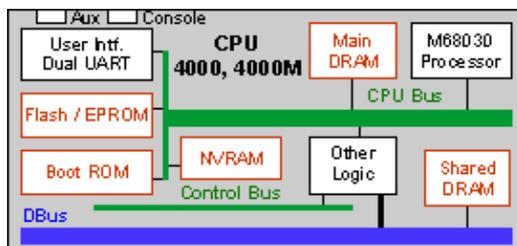
Processor: There are four types of processors used on the 4000 Series platforms:

- The 4000 and 4000–M use a Motorola 68030 CISC 40 MHz
- The 4500 uses IDT Orion R4600 RISC 100 Mhz
- The 4500–M uses IDT Orion R4700 RISC 100 MHz
- The 4700 and 4700–M use IDT Orion R4700 RISC 133 MHz

Memory: Used in various forms for storage purposes such as storing an operating system (IOS), configuration, RxBoot, packets, and so on. Details are provided in the Memory Details section.

System control logic helps the main processor in device control, interrupt handling, counting/timing, data transfer, minimal First In First Out (FIFO) buffering, and communication with DRAM and slower input/output (I/O) devices.

- The 4000 and 4000–M use various specialized controllers and discrete logic for this purpose.
- All other 4x00 Series platforms use Reno Application–Specific Integrated Circuit (ASIC) for data transfer to DRAM and DBus, and Nevada ASIC for accessing slower I/O devices. These are shown as System Control ASICs (one block) in the above diagrams.

Buses are used by the CPU for accessing various components of the system and transferring instructions and data to or from specified memory addresses.

- **CPU Bus** is for high speed operations, with direct processor access. Its characteristics depend upon the processor used. In the 4000 and 4000–M, DRAM is directly accessible through the CPU Bus, while in other 4x00 Series platforms, it is accessible through Reno ASIC.
- **I/O Bus** allows Nevada ASIC to access slower I/O devices in the 4500 and 4700 platforms. In the 4000 and 4000–M, the main processor has direct access to I/O devices through the CPU Bus.
- **DBus** (asynchronous, 32 bit data, 24 bit address) is the system bus used for communication with NIMs and shared DRAM.
- **Control Bus** (asynchronous, 16 bit data, 18 bit address) allows the CPU to perform NIM interrupt handling and access NIM status registers.

Dual Universal Asynchronous Receiver/Transmitter (UART) provides the necessary user interface. It has two RS232 asynchronous ports, data communications equipment (DCE) (console, DB25, male) and data terminal equipment (DTE) (auxiliary, DB25, female).

Memory Details

The 4x00 platforms are **shared memory** routers: there is only one region of shared memory, which is shared between a single main processor and all the interface controllers. This shared memory is used for all the switching paths: fast switching and process switching.

Main DRAM: The Cisco 4000 Series runs from a Cisco IOS software image stored in main DRAM. This memory is also used by the system for tables and stacks. Therefore, the main memory is used for code space and routing data. The amount of main memory required is determined by the size of the image and by the configuration of the router.

Physical Memory Differences for the 4x00 Platform

Router	Number of Single In–Line Memory Module (SIMM) Sockets
4000	Four – 4 x 1 or 4 x 4 MB
4000–M	One – 8, 16, 32 MB
4500 and 4700	Two – Both must be equally filled (4 & 4 or 16 & 16 MB)
4500–M	Two – Both must be equally filled (4 & 4, 8 & 8, 16 & 16 MB)
4700–M	Two – Both must be equally filled (4 & 4, 16 & 16, 32 & 32 MB)

Shared DRAM: Also called packet memory, it contains the interface's private and public system buffers. As the 4000 Series routers are **shared memory** platforms, the system buffers are used for **all** packet switching, not just process switching. The recommended amount of shared memory is determined by the type, and the number of physical and/or virtual interfaces supported by the router.

Router	Number of Single In–Line Memory Module (SIMM) Sockets
4000	Four – 4 x 1MB
4000–M and 4500	One – 4 or 16 MB
	One – 4, 8 or 16 MB

4500–M, 4700, and 4700–M	

Flash: Used for permanent storage of full Cisco IOS software, which is loaded in DRAM for execution.

Router	Number of Single In–Line Memory Module (SIMM) Sockets
4000	One socket available for a memory daughter card with either EEPROM or Flash SIMM support. Original daughter cards supported either 2 or 4MB of EEPROM; later versions support 8 or 16MB of Flash SIMM, using either two 4MB SIMMs or two 8MB SIMMs on the daughter card.
4000–M and 4500	Two SIMM sockets for Flash memory. One or both may be filled, but if both are filled, equal memory sizes are required (4 or 4 & 4 MB).
4500–M, 4700, and 4700–M	Two SIMM sockets for Flash memory. One or both may be filled, but if both are filled, equal memory sizes are required (4, 4 & 4, 8 or 8 & 8 MB).

Boot Flash: Used for permanent storage of **RxBoot**, a subset of Cisco IOS software, which is loaded in DRAM for execution.

Router	Number of Single In–Line Memory Module (SIMM) Sockets
4500, 4500–M, 4700, and 4700–M	One SIMM slot for Boot Flash (4, 8, or 16 MB)
4000 and 4000–M	These routers don't have Boot Flash, and store RxBoot in **Boot ROM** (which also runs from Boot ROM).

Boot ROM: Erasable programmable read–only memory (EPROM) used for permanently storing startup diagnostic code (ROM Monitor).

Router	Boot ROM Size
4500, 4500–M, 4700, and 4700–M	256 KB
4000 and 4000–M	2 MB (more space is needed for the RxBoot)

Nonvolatile RAM (NVRAM): Used for writable permanent storage of the startup configuration.

- It is a battery–backed SRAM.
- NVRAM size is 128 KB.

Boot Sequence

Below is a typical boot sequence on a 4500–M router.

The first stage is the loading of the bootstrap, which is stored in Boot ROM. The purpose of the bootstrap is to initialize the CPU and launch the bootloader:

```
System Bootstrap, Version 5.3(10) [tamb 10], RELEASE SOFTWARE (fc1)
Copyright (c) 1994 by cisco Systems, Inc.
C4500 processor with 32768 Kbytes of main memory
```

The second stage is the bootloader, which is used to boot the system image. If a bootloader is not present in bootflash memory, the bootstrap attempts to load the first file available in Flash memory. The bootloader does not have all the routing code, and only knows how to read the configuration, access the file system, and behave as a network host for net booting:

```
program load complete, entry point: 0x80008000, size: 0x1fa770
Self decompressing the image : ################################################
#############################################################################
################### [OK]

%SYS-6-BOOT_MESSAGES: Messages above this line are from the boot loader.
```

The third stage is the boot of the system image. This is the main Cisco IOS software image located in Flash memory or on a Trivial File Transfer Protocol (TFTP) server:

```
program load complete, entry point: 0x80008000, size: 0x685e64
Self decompressing the image : ################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#############################################################################
#### [OK]

             Restricted Rights Legend

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) 4500 Software (C4500-DS-M), Version 12.1(9), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 13-Jun-01 18:37 by kellythw
Image text-base: 0x60008958, data-base: 0x60CDE000

cisco 4500 (R4K) processor (revision D) with 32768K/16384K bytes of memory.
```

```
Processor board ID 04059850
R4700 CPU at 100Mhz, Implementation 33, Rev 1.0
G.703/E1 software, Version 1.0.
Bridging software.
X.25 software, Version 3.0.0.
2 Ethernet/IEEE 802.3 interface(s)
1 FastEthernet/IEEE 802.3 interface(s)
4 Serial network interface(s)
128K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)
4096K bytes of processor board Boot flash (Read/Write)



Press RETURN to get started!
```

The last stage is the initialization of the interfaces and the message %SYS−5−RESTART: System restarted.

```
00:00:11: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to up
00:00:11: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
00:00:11: %LINK-3-UPDOWN: Interface Ethernet1, changed state to up
00:00:11: %LINK-3-UPDOWN: Interface Serial0, changed state to down
00:00:11: %LINK-3-UPDOWN: Interface Serial1, changed state to down
00:00:11: %LINK-3-UPDOWN: Interface Serial2, changed state to down
00:00:11: %LINK-3-UPDOWN: Interface Serial3, changed state to down
00:00:12: %SYS-5-CONFIG_I: Configured from memory by console
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0,
changed state to up
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1, changed state to down
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to down
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2, changed state to down
00:00:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3, changed state to down
00:00:14: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (TM) 4500 Software (C4500-DS-M), Version 12.1(9), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 13-Jun-01 18:37 by kellythw
00:00:14: %LINK-5-CHANGED: Interface Serial0, changed state to administratively down
00:00:14: %LINK-5-CHANGED: Interface Serial1, changed state to administratively down
00:00:14: %LINK-5-CHANGED: Interface Serial2, changed state to administratively down
00:00:14: %LINK-5-CHANGED: Interface Serial3, changed state to administratively down
00:00:15: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
00:00:15: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1, changed state to up
router>
```

Packet Switching

The switching architecture of the Cisco 4x00 series is based on the *shared memory* architecture. The 1600, 2500 and AS5300 also use this switching architecture.

The Cisco IOS software on shared memory routers uses the system buffers for *all packet switching*, not just process switching. In addition to the standard public buffer pools, the Cisco IOS software also creates *private system buffer pools* and special buffer structures for the interface controllers called *RX rings* and *TX rings*.

Private Buffer Pools

Private buffer pools are static and are allocated with a fixed number of buffers during initialization of the Cisco IOS software. New buffers cannot be created on demand for these pools. If a buffer is needed and none are available in the private pool, Cisco IOS software *falls back* to the public buffer pool for the size that

matches the interface's maximum transmission unit (MTU).

Receive Rings and Transmit Rings

Cisco IOS software creates these rings on behalf of the media controllers and then manages them jointly with the controllers. Each interface has a pair of rings: a receive ring for receiving packets and a transmit ring for transmitting packets.

Receive rings have a constant number of packet buffers allocated to them that equals the size of the ring. The **show controller** command below displays the size and the location of the receive and transmit rings.

```
router#show controllers ethernet 1
LANCE unit 1, NIM slot 1, NIM type code 9, NIM version 3
Media Type is 10BaseT, Link State is Up, Squelch is Normal
idb 0x612F2730, ds 0x612F42F8, eim_regs = 0x3C100800
IB at 0x401E7AE4: mode=0x0000, mcfilter 0000/0000/0100/0000
station address 0060.5c5e.8a84  default station address 0060.5c5e.8a84
buffer size 1524
RX ring with 32 entries at 0x401E7B28
Rxhead = 0x401E7BD0 (21), Rxp = 0x612F4364 (21)
00 pak=0x612F9FD8 Ds=0xA81F7916 status=0x80 max_size=1524 pak_size=67
01 pak=0x612FB55C Ds=0xA81FB5B2 status=0x80 max_size=1524 pak_size=64
...
TX ring with 32 entries at 0x401E7D68, tx_count = 0
TX_head = 0x401E7E50 (29), head_txp = 0x612F4498 (29)
TX_tail = 0x401E7E50 (29), tail_txp = 0x612F4498 (29)
00 pak=0x000000 Ds=0xA82B6B3A status=0x03 status2=0x0000 pak_size=60
01 pak=0x000000 Ds=0xA82B6B3A status=0x03 status2=0x0000 pak_size=60
02 pak=0x000000 Ds=0xA82B6B3A status=0x03 status2=0x0000 pak_size=60
...
```

RX ring with 32 entries at 0x401E7B28: The size of the receive ring is 32 and it begins at the address 0x401E7B28 in I/O memory.

TX ring with 32 entries at 0x401E7D68, TX_count = 0: The size of the transmit ring is 32, it begins at the address 0x401E7D68 in I/O memory and there are no packets awaiting transmission on this interface.

Switching Paths

This description is based on the book *Inside Cisco IOS Software Architecture*, Cisco Press.[1]

1 – Receiving the packet

Step 1: The interface media controller detects a packet on the network media and copies it into a buffer pointed to by the first free element in the receive ring. Media controllers use the Direct Memory Access (DMA) method to copy packet data into memory.

Step 2: The media controller changes ownership of the packet buffer back to the processor and issues a receive interrupt to the processor. The media controller does not have to wait for a response from the CPU and continues to receive incoming packets into the receive ring.

It's possible for the media controller to fill the receive ring before the processor processes all the new buffers in the ring. This condition is called an overrun. When this occurs, all incoming packets are dropped until the processor recovers.

Step 3: The CPU responds to the receive interrupt, and attempts to remove the newly–filled buffer from the receive ring and replenish the ring from the interface's private pool. Notice that packets are not physically

moved within the I/O memory; only the pointers are changed. If the interface's input hold queue is full, the packet is *dropped*; otherwise three outcomes are possible:

- **3.1:** A free buffer is available in the interface's private pool to replenish the receive ring: the free buffer is linked to the receive ring and the packet now belongs to the interface's private buffers pool.
- **3.2:** A free buffer is not available in the interface's private pool, so the receive ring is replenished by falling back to the global pool that matches the interface's MTU. The *fallback* counter is incremented for the private pool.
- **3.3:** If a free buffer is not available in the public pool as well, the incoming packet is dropped and the *ignore* counter is incremented. In addition, the interface is *throttled* and all incoming traffic is ignored on this interface for a short period of time.

2 – Switching the Packet

Step 4: After the receive ring is replenished, the CPU begins switching the packet. Cisco IOS software attempts to switch the packet using the fastest method configured on the interface. On shared memory routers, it first tries Cisco Express Forwarding (CEF) switching (if configured), then fast switching (unless "no ip route−cache" is configured on the interface), and finally, process switching if none of the others work.

Step 5: While still in the receive interrupt context, the Cisco IOS software attempts to use the CEF table or the fast switching cache to make a switching decision.

- **5.1: CEF switching** – If there are valid CEF and adjacency table entries, the Cisco IOS software rewrites the Media Access Control (MAC) header on the packet and begins transmitting it (Step 8). If there is no CEF entry for the destination, the packet is dropped.
- **5.2: Fast switching** – If CEF is not enabled or the packet cannot be CEF switched, the Cisco IOS software attempts to fast switch the packet. If there is a valid fast cache entry for this destination, the Cisco IOS software rewrites the MAC header information and begins transmitting the packet (Step 8). If there is no valid fast cache entry, the packet is queued for process switching (Step 6).

Step 6: Process switching – If both CEF and fast switching fail, the Cisco IOS software falls back to process switching. The packet goes in the queue of the appropriate process (for instance, an IP packet is placed in the queue for the IP Input process), and the receive interrupt is dismissed.

Step 7: Eventually the packet switching process runs, switching the packet and rewriting the MAC header as needed. Note that the packet still has not moved from the buffer it was originally copied into. After the packet is switched, the Cisco IOS software continues to the packet transmit stage.

3 – Transmitting the Packet

Step 8: If the packet was CEF or fast switched, then while still in receive interrupt context, the Cisco IOS software checks to see if there are packets on the output queue of the outbound interface.

- **8.1:** If there are packets already on the output hold queue for the interface, the Cisco IOS software places the packet on the output hold queue instead of directly into the transmit ring to reduce the possibility of out−of−order packets, and then proceeds to Step 8.3.
- **8.2:** If the output hold queue is empty, the Cisco IOS software places the packet on the transmit ring of the output interface by linking the packet buffer to a transmit ring descriptor. The receive interrupt is dismissed and processing continues with Step 11. If there is no room on the transmit ring, the packet is placed on the output hold queue instead, and the receive interrupt is dismissed.
- **8.3:** If the output hold queue is full, the packet is dropped, the *output drop* counter is incremented, and the receive interrupt is dismissed.

Step 9: If the packet was process switched, the packet is placed on the output queue for the input interface. If the output queue is full, the packet is dropped and the *output drop* counter is incremented.

Step 10: The Cisco IOS software attempts to find a free descriptor in the output interface transmit ring. If a free descriptor exists, the Cisco IOS software removes the packet from the output hold queue and links the buffer to the transmit ring. If the ring is full, the Cisco IOS software leaves the packet in the output hold queue until the media controller transmits a packet from the ring and frees a descriptor.

Step 11: The outbound interface media controller polls its transmit ring periodically for packets that need to be transmitted. As soon as the media controller detects a packet, it copies the packet onto the network media and raises a transmit interrupt to the processor.

Step 12: The Cisco IOS software acknowledges the transmit interrupt, unlinks the packet buffer from the transmit ring, and returns the buffer to the pool of buffers from which it originated. The Cisco IOS software then checks the output hold queue for the interface. If there are any packets waiting in the output hold queue, the Cisco IOS software removes the next one from the queue and links it to the transmit ring. Finally, the transmit interrupt is dismissed.

Performance Figures

The table below indicates packets processed per second (pps), based on a 64−byte IP packet:

Switching Path	Cisco 4000	Cisco 4500	Cisco 4700
Process	1,800	3,500	4,600
Fast	14,000	45,000	75,000

1 "CCIE Professional Development: Inside Cisco IOS Software Architecture" by Vijay Bollapragada, Curtis Murphy, Russ White (ISBN 1−57870−181−3).

Related Information

- **Password Recovery for the 4000 Series Routers**
- **Password Recovery for the 4500 Series Routers**
- **Password Recovery for the 4700 Series Routers**
- **Boot Failure Recovery Procedure for the Cisco 4000 Series Routers**
- **Boot Failure Recovery Procedure for the Cisco 4500 Series Routers**
- **Boot Failure Recovery Procedure for the Cisco 4700 Series Routers**
- **Software Installation and Upgrade Procedure**
- **Cisco 4000 Series Configuration Notes**
- **Cisco 4000 Series Installation Guide**
- **Release Notes – Cisco 4000 Series Routers**
- **Technical Support – Cisco Systems**