TCP/IP Model and Ethernet LAN

Ashwini Mathur

Overview

TCP provides a wide variety of services to applications, whereas UDP does not. For example, routers discard packets for many reasons, including bit errors, congestion, and instances in which no correct routes are known.

Most data link protocols notice errors (a process called **error detection**) but then discard frames that have errors. TCP provides retransmission (**error recovery**) and helps to avoid congestion (**flow control**), whereas **UDP does not**.

Table 5-1 TCP/IP Transport Layer Features

Function	Description
Multiplexing using ports	Function that allows receiving hosts to choose the correct application for which the data is destined, based on the port number.
Error recovery (reliability)	Process of numbering and acknowledging data with Sequence and Acknowledgment header fields.
Flow control using windowing	Process that uses window sizes to protect buffer space and routing devices from being overloaded with traffic.
Connection establishment and termination	Process used to initialize port numbers and Sequence and Acknowledgment fields.
Ordered data transfer and data segmentation	Continuous stream of bytes from an upper-layer process that is "segmented" for transmission and delivered to upper-layer processes at the receiving device, with the bytes in the same order.

TCP Header fields

The message created by TCP that begins with the TCP header, followed by any application data, is called a TCP segment. Alternately, the more generic term Layer 4 PDU, or L4PDU, can also be used.

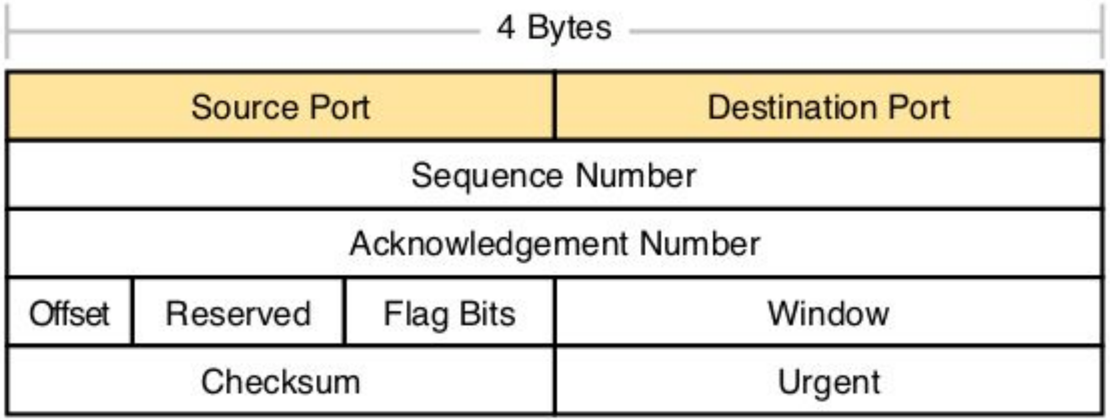
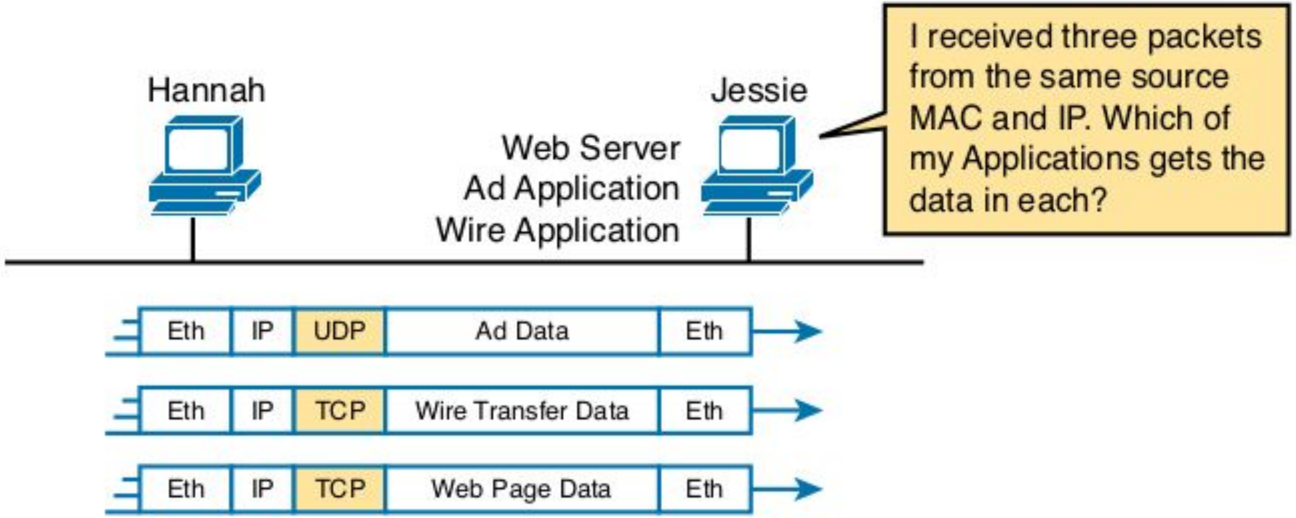


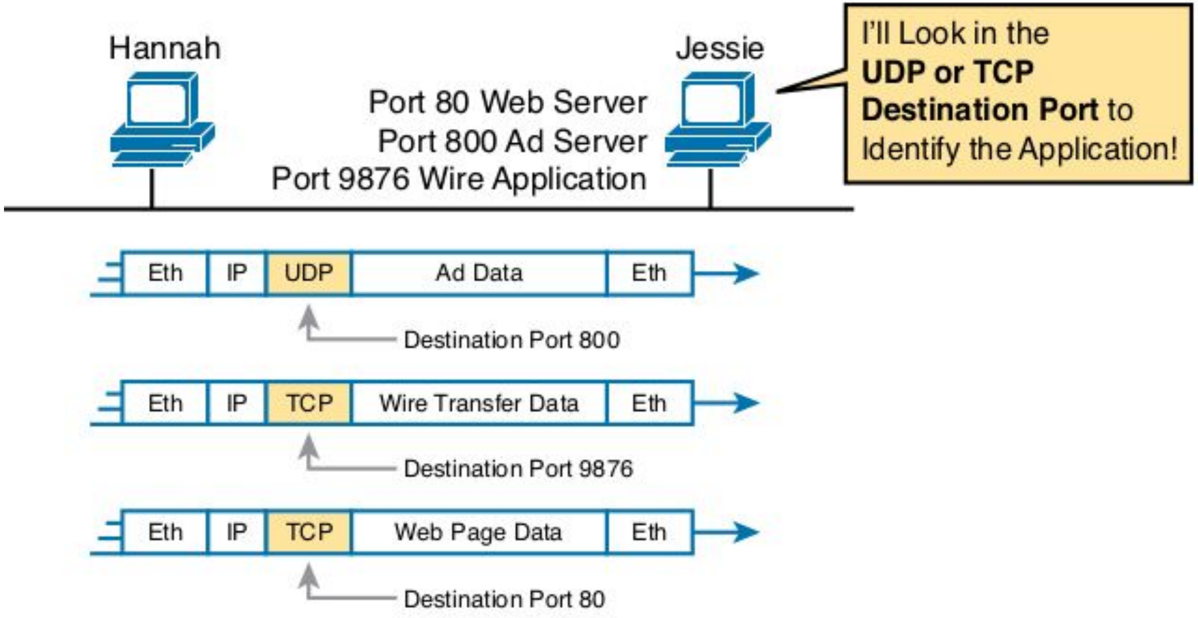
Figure 5-2 shows the sample network, with Jessie running three applications:

- A UDP-based ad application
- A TCP-based wire-transfer application
- A TCP web server application



Jessie needs to know which application to give the data to, but *all three packets are from the same Ethernet and IP address*. You might think that Jessie could look at whether the packet contains a UDP or TCP header, but as you see in the figure, two applications (wire transfer and web) are using TCP.

TCP and UDP solve this problem by using a port number field in the TCP or UDP header, respectively. Each of Hannah's TCP and UDP segments uses a different *destination port number* so that Jessie knows which application to give the data to. Figure 5-3 shows an example.



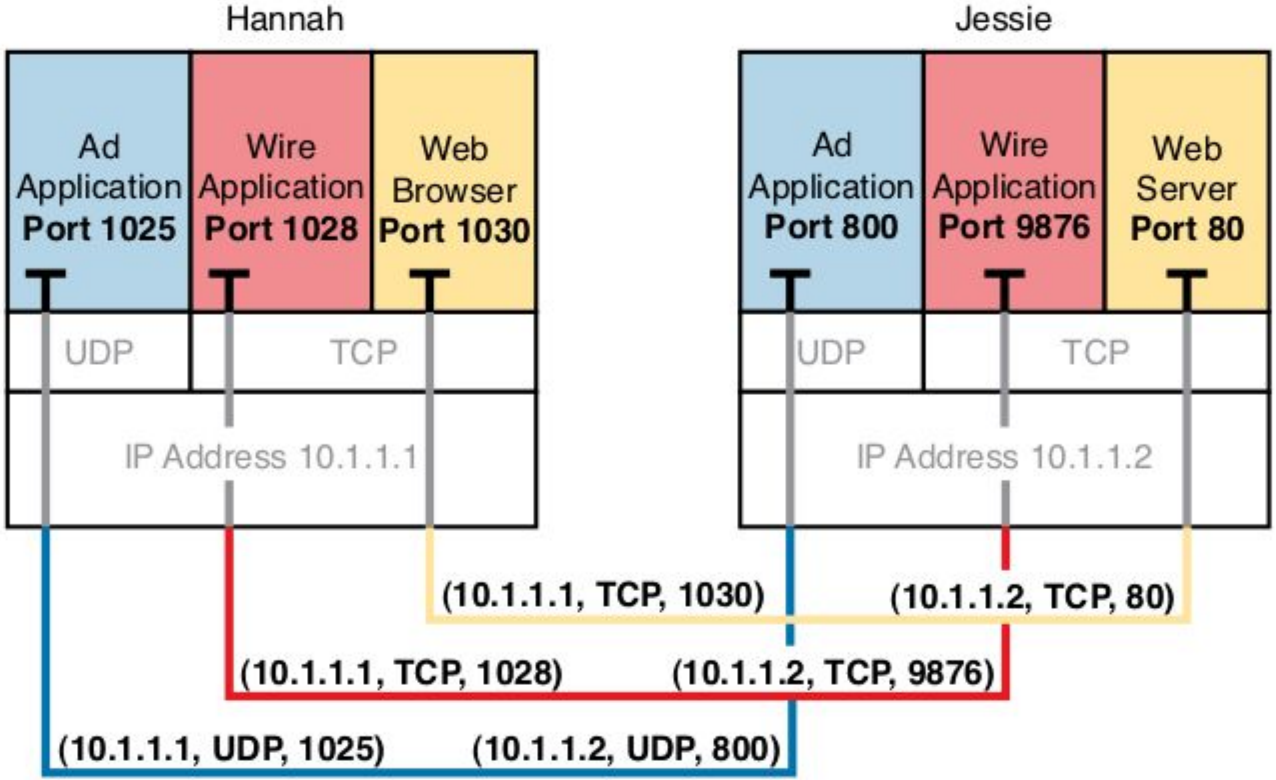
Multiplexing relies on a concept called a *socket*. A socket consists of three things:

- An IP address
- A transport protocol
- A port number

So, for a web server application on Jessie, the socket would be (10.1.1.2, TCP, port 80)

Port numbers are a vital part of the socket concept. Well-known port numbers are used by servers; other port numbers are used by clients.

Applications that provide a service, such as FTP, Telnet, and web servers, open a socket using a well-known port and listen for connection requests.



Connection between Sockets

Popular Application and well known Port Numbers

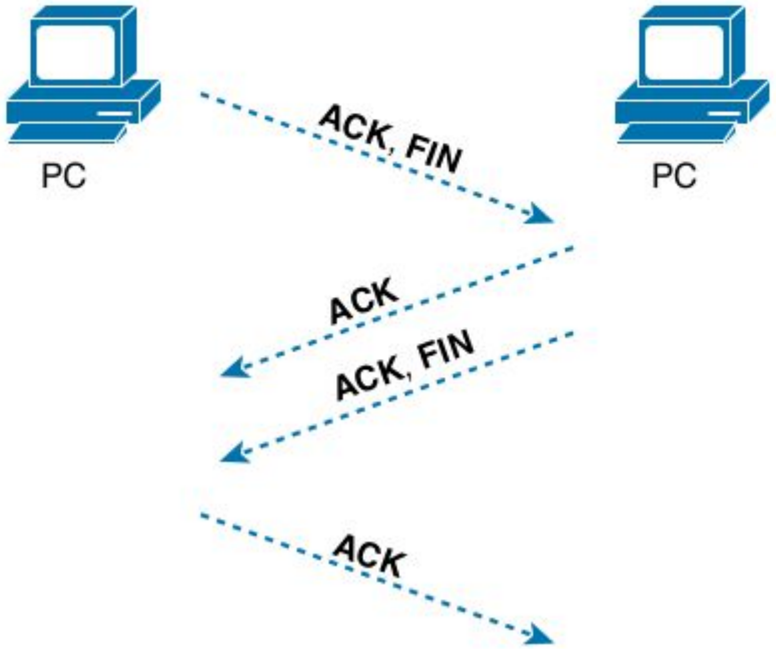
Port Number	Protocol	Application
20	TCP	FTP data
21	TCP	FTP control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP, TCP	DNS
67, 68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP (WWW)
110	TCP	POP3
161	UDP	SNMP
443	TCP	SSL

TCP CONNECTION ESTABLISHMENT :

Connection establishment refers to the process of initializing sequence and acknowledgment fields and agreeing on the port numbers used.

TCP CONNECTION TERMINATION :

Connection establishment refers to the process of initializing sequence and acknowledgment fields and agreeing on the port numbers used.

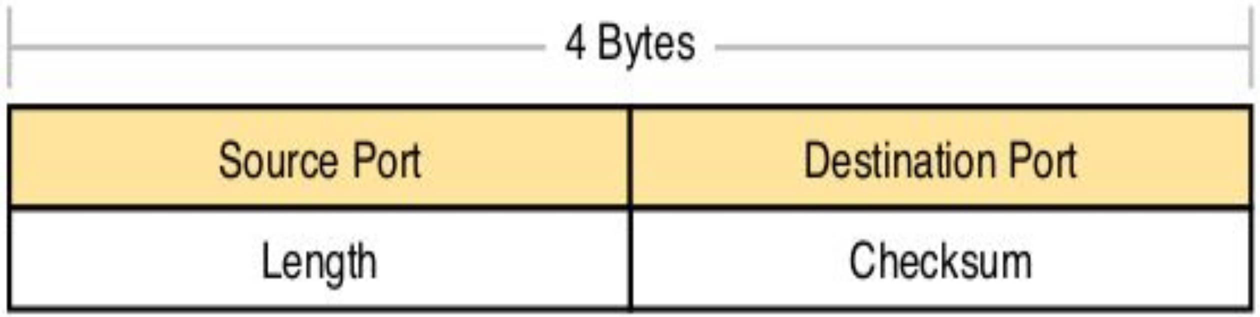


TCP establishes and terminates connections between the endpoints, whereas UDP does not. Many protocols operate under these same concepts, so the terms *connection-oriented* and *connectionless* are used to refer to the general idea of each. More formally, these terms can be defined as follows:

- **Connection-oriented protocol:** A protocol that requires an exchange of messages before data transfer begins, or that has a required preestablished correlation between two endpoints.
- **Connectionless protocol:** A protocol that does not require an exchange of messages and that does not require a preestablished correlation between two endpoints.

User Datagram Protocol

The UDP header format. Most importantly, note that the header includes source and destination port fields, for the same purpose as TCP. However, the UDP has only 8 bytes, in comparison to the 20-byte TCP header.



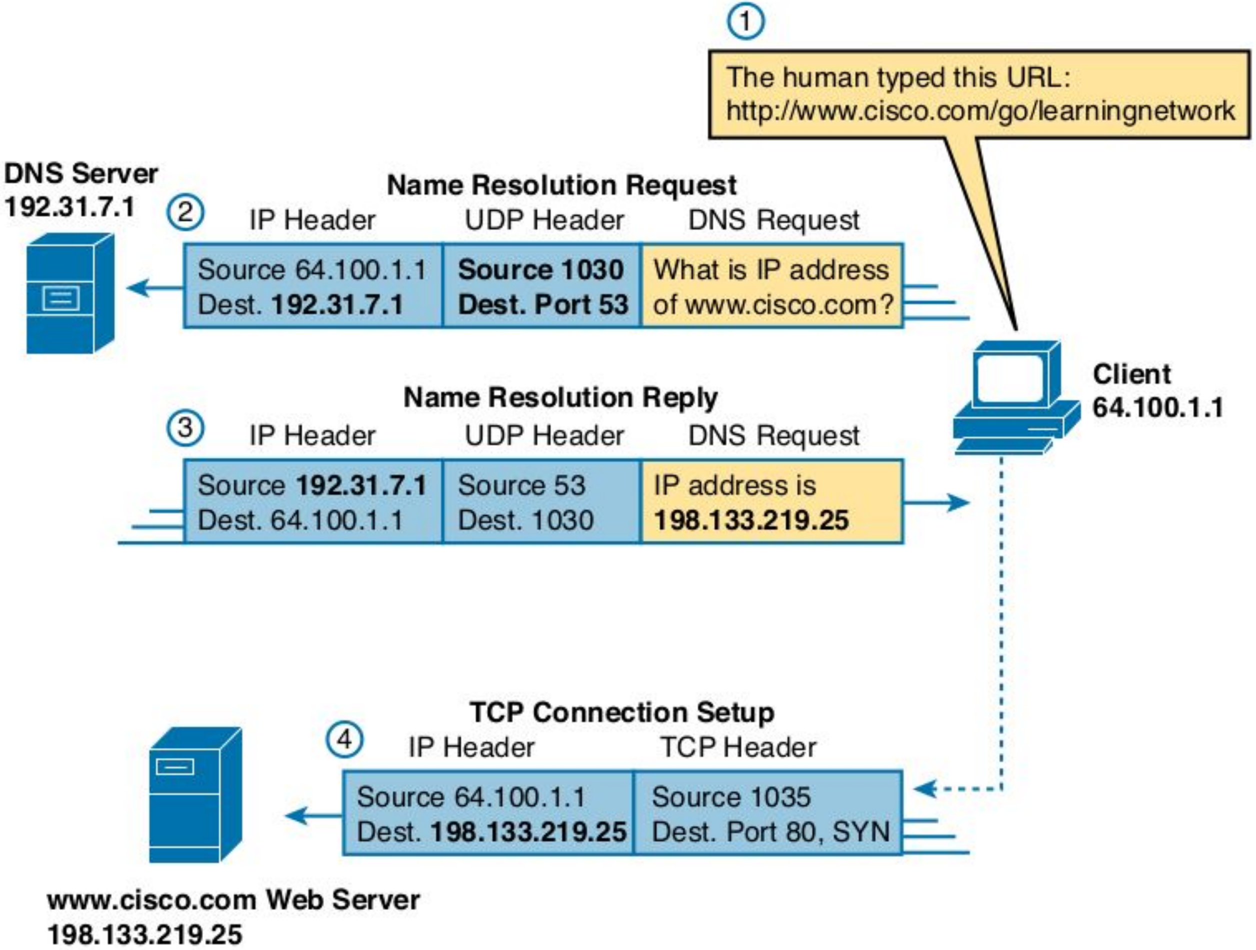
QoS in general defines the quality of the data transfer between two applications and in the network as a whole. QoS often breaks down these qualities into four competing characteristics:

Bandwidth: The volume of bits per second needed for the application to work well; it can be biased with more volume in one direction, or balanced.

Delay: The amount of time it takes one IP packet to flow from sender to receiver.

Jitter: The variation in delay.

Loss: The percentage of packets discarded by the network before they reach the destination, which when using TCP, requires a retransmission.



1
The human typed this URL:
http://www.cisco.com/go/learningnetwork
DNS Server
192.31.7.1
2
Name Resolution Request
IP Header
UDP Header
DNS Request
Source 64.100.1.1
Dest. 192.31.7.1
Source 1030
Dest. Port 53
What is IP address
of www.cisco.com?
Client
64.100.1.1
3
Name Resolution Reply
IP Header
UDP Header
DNS Request
Source 192.31.7.1
Dest. 64.100.1.1
Source 53
Dest. 1030
IP address is
198.133.219.25
4
TCP Connection Setup
IP Header
TCP Header
Source 64.100.1.1
Dest. 198.133.219.25
Source 1035
Dest. Port 80, SYN
www.cisco.com Web Server
198.133.219.25

- Bridges separated devices into groups called *collision domains*.
- Bridges reduced the number of collisions that occurred in the network, because frames inside one collision domain did not collide with frames in another collision domain.
- Bridges increased bandwidth by giving each collision domain its own separate bandwidth, with one sender at a time per collision domain.

Figure 6-2 shows the effect of migrating from using a 10BASE-T hub without a bridge (as in Figure 6-1) to a network that uses a bridge. The bridge in this case separates the network into two separate collision domains (CD).

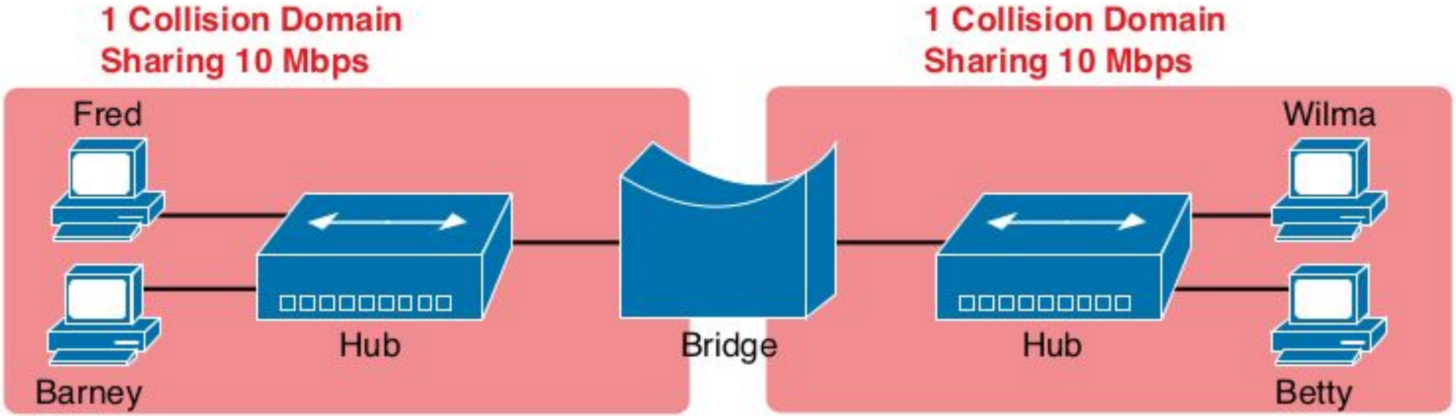
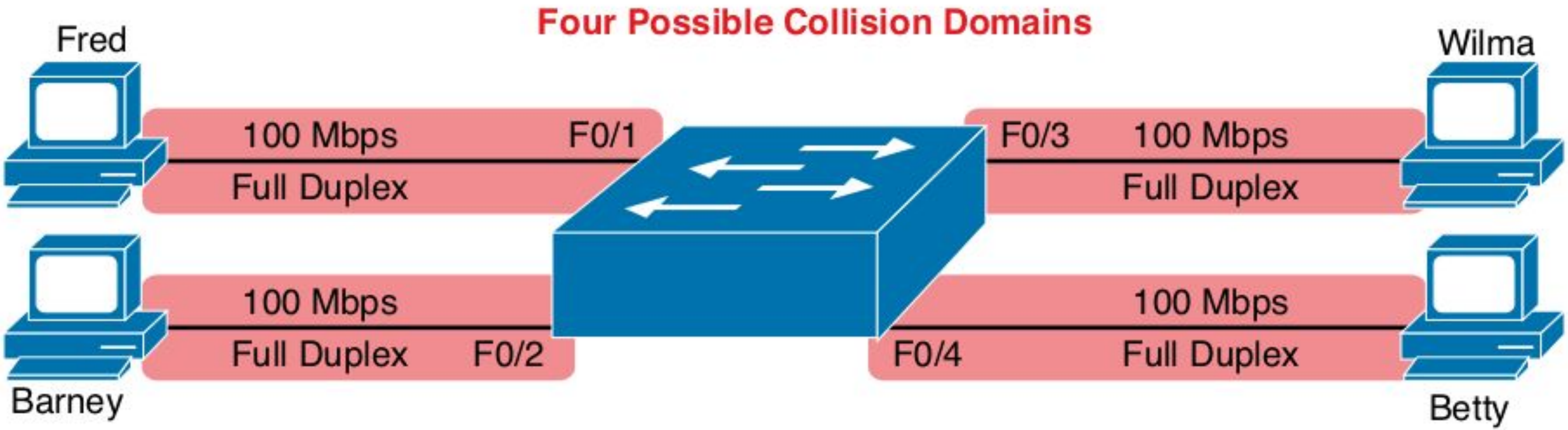


Figure 6-2 *Bridge Creates Two Collision Domains and Two Shared Ethernets*


Four Possible Collision Domains
Fred
Wilma
Barney
Betty
100 Mbps
Full Duplex
F0/1
F0/2
F0/3
F0/4

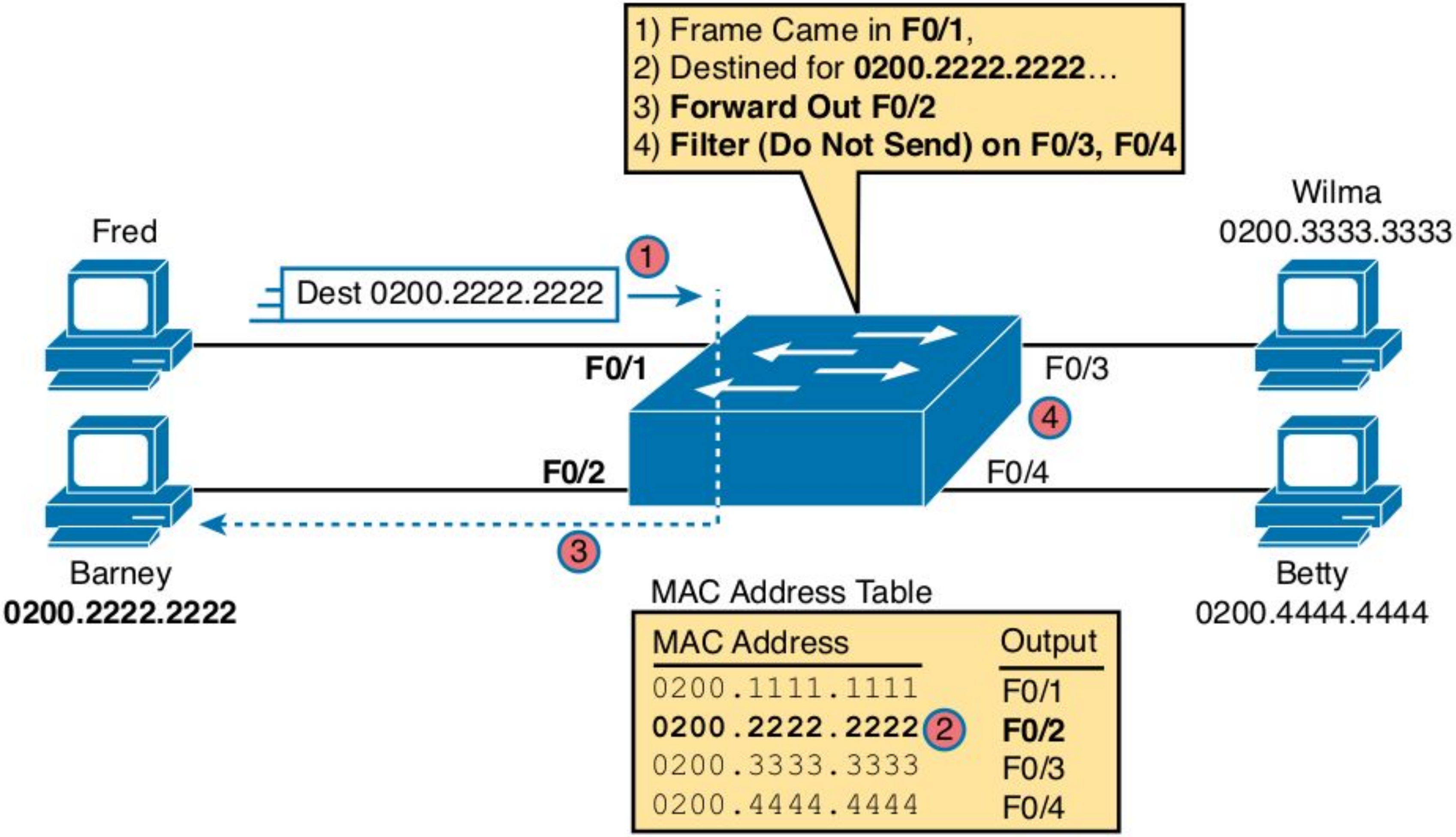
Switching Logic

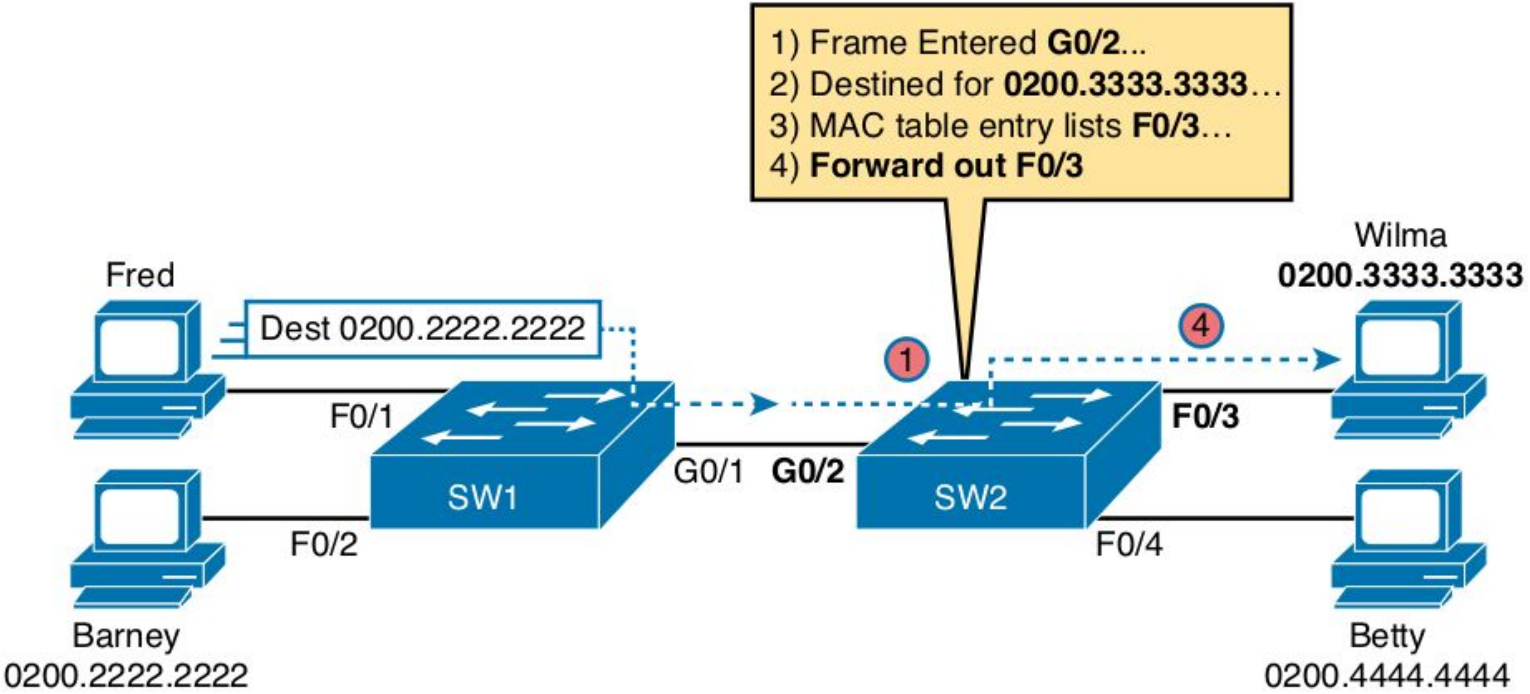
Ultimately, the role of a LAN switch is to forward Ethernet frames. To achieve that goal, switches use logic—logic based on the source and destination MAC address in each frame's Ethernet header.

This book discusses how switches forward unicast frames and broadcast frames, ignoring multicast Ethernet frames. Unicast frames have a unicast address as a destination; these addresses represent a single device. A broadcast frame has a destination MAC address of FFFF.FFFF.FFFF; this frame should be delivered to all devices on the LAN.

LAN switches receive Ethernet frames and then make a switching decision: either forward the frame out some other port(s) or ignore the frame. To accomplish this primary mission, transparent bridges perform three actions:

1. Deciding when to forward a frame or when to filter (not forward) a frame, based on the destination MAC address.
2. Learning MAC addresses by examining the source MAC address of each frame received by the switch.
3. Creating a (Layer 2) loop-free environment with other bridges by using Spanning Tree Protocol (STP).


1) Frame Came in F0/1,
2) Destined for 0200.2222.2222…
3) Forward Out F0/2
4) Filter (Do Not Send) on F0/3, F0/4
Fred
Dest 0200.2222.2222
1
Wilma
0200.3333.3333
F0/1
F0/3
4
F0/2
F0/4
3
Barney
0200.2222.2222
Betty
0200.4444.4444
MAC Address Table
MAC Address
Output
0200.1111.1111
F0/1
0200.2222.2222
2
F0/2
0200.3333.3333
F0/3
0200.4444.4444
F0/4

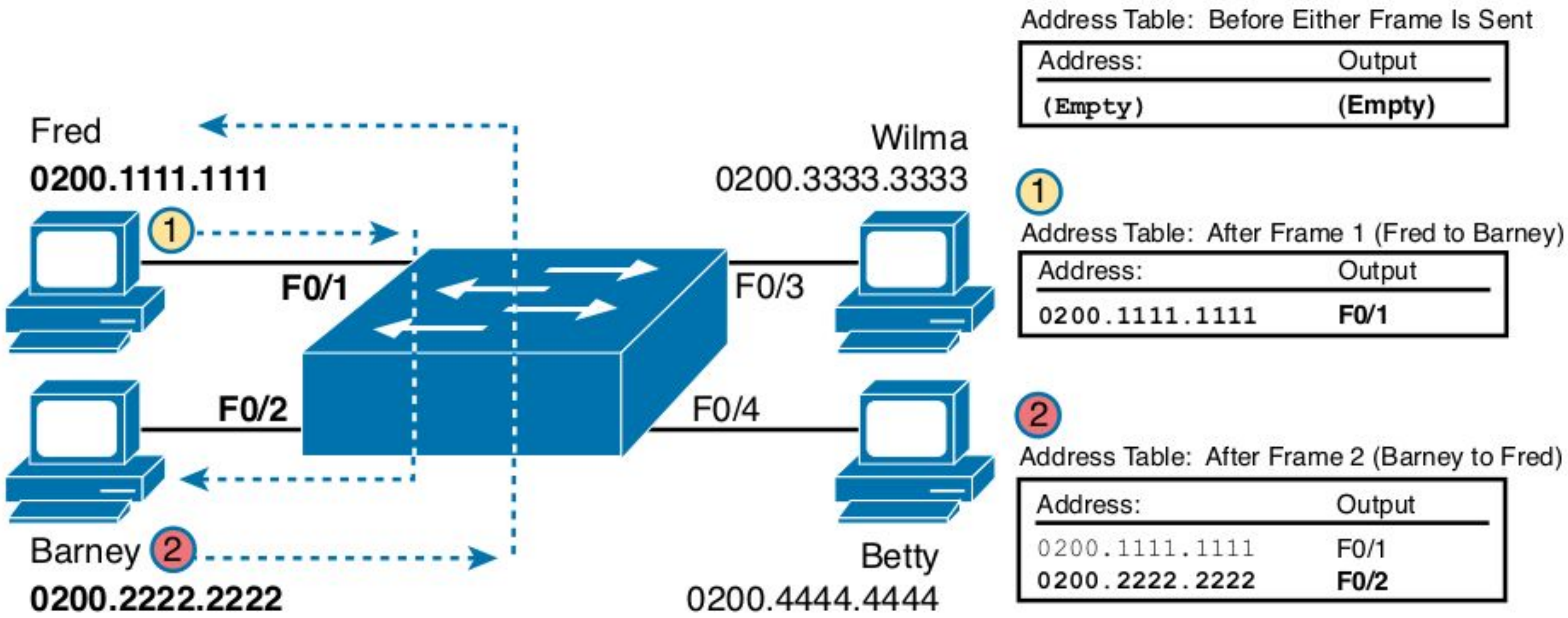


SW1 Address Table

MAC Address	Output
0200.1111.1111	F0/1
0200.2222.2222	F0/2
0200.3333.3333	**G0/1**
0200.4444.4444	G0/1

SW2 Address Table

MAC Address	Output
0200.1111.1111	G0/2
0200.2222.2222	G0/2
0200.3333.3333 2	**F0/3** 3
0200.4444.4444	F0/4



Fred
0200.1111.1111
1
F0/1
Wilma
0200.3333.3333
F0/3
F0/2
F0/4
Barney
2
0200.2222.2222
Betty
0200.4444.4444
Address Table: Before Either Frame Is Sent
Address: Output
(Empty) (Empty)
1
Address Table: After Frame 1 (Fred to Barney)
Address: Output
0200.1111.1111 F0/1
2
Address Table: After Frame 2 (Barney to Fred)
Address: Output
0200.1111.1111 F0/1
0200.2222.2222 F0/2

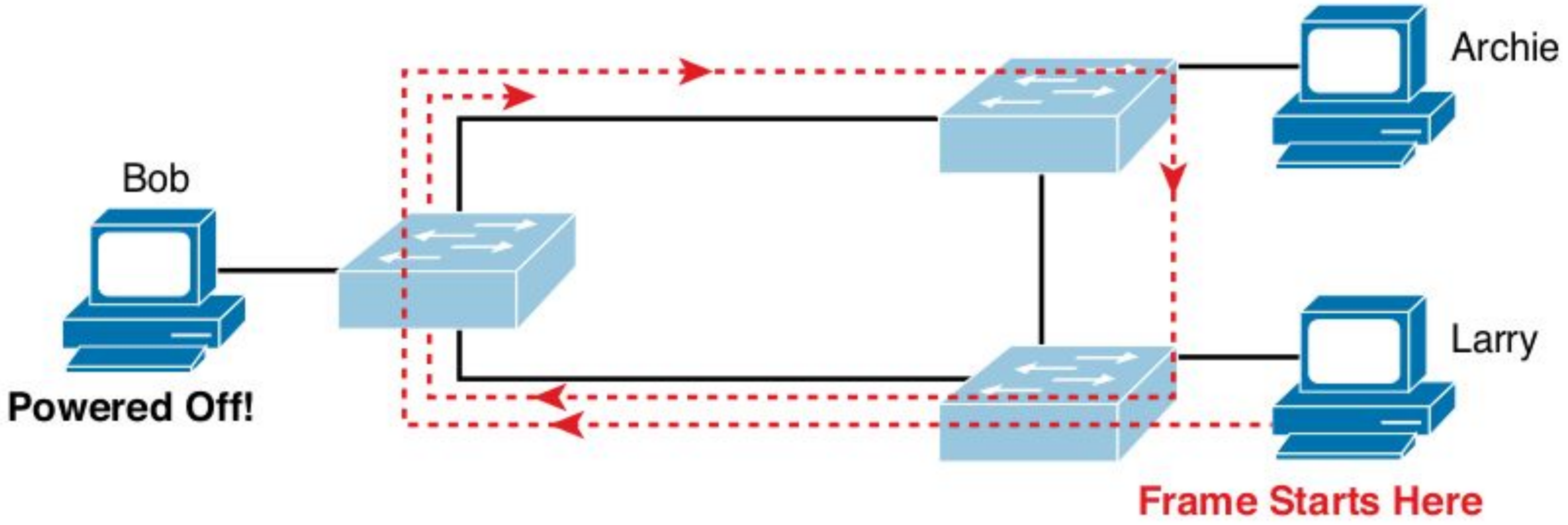

Archie
Bob
Larry
Powered Off!
Frame Starts Here

Table 6-1 Switch Internal Processing

Switching Method	Description
Store-and-forward	The switch fully receives all bits in the frame (store) before forwarding the frame (forward). This allows the switch to check the FCS before forwarding the frame.
Cut-through	The switch forwards the frame as soon as it can. This reduces latency but does not allow the switch to discard frames that fail the FCS check.
Fragment-free	The switch forwards the frame after receiving the first 64 bytes of the frame, thereby avoiding forwarding frames that were errored because of a collision.

Collision Domains

Originally, the term *collision domain* referred to an Ethernet concept of all ports whose transmitted frames would cause a collision with frames sent by other devices in the collision domain. To review the core concept, Figure 6-8 illustrates collision domains.

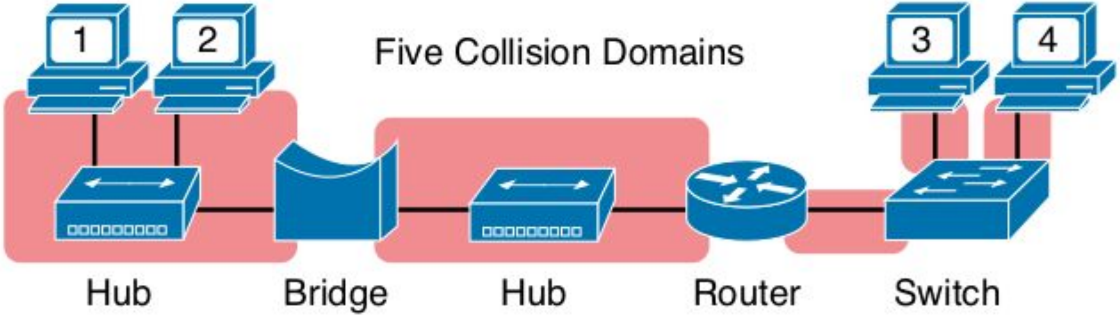
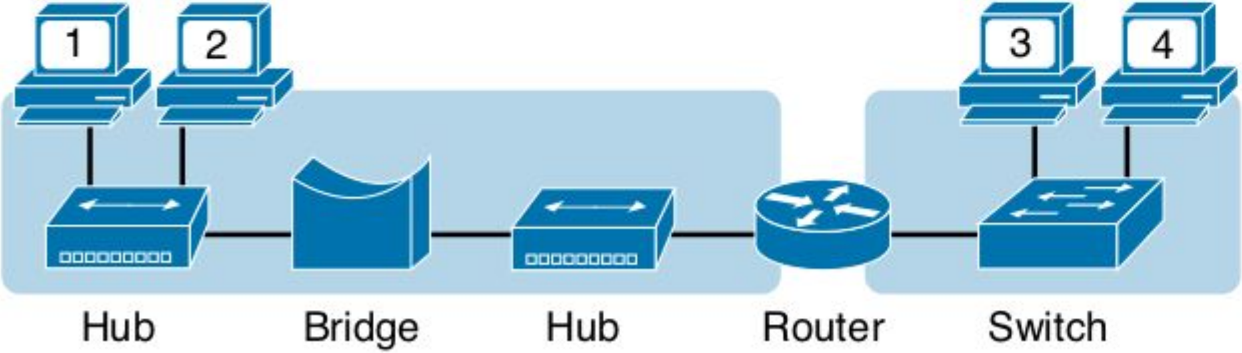


Figure 6-8 *Collision Domains*

Two Broadcast Domains

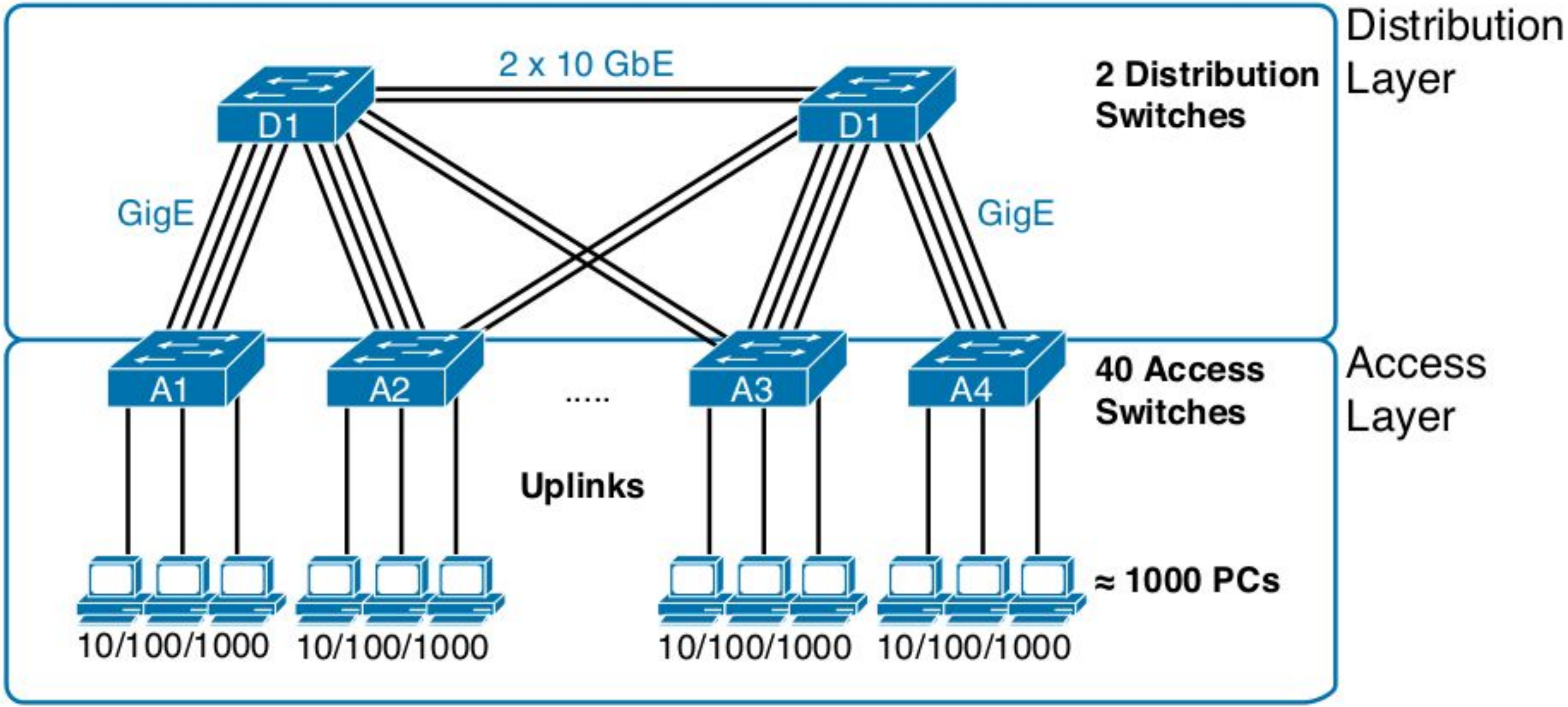


General definitions for a collision domain and a broadcast domain are as follows:

- A *collision domain* is a set of network interface cards (NIC) for which a frame sent by one NIC could result in a collision with a frame sent by any other NIC in the same collision domain.
- A *broadcast domain* is a set of NICs for which a broadcast frame sent by one NIC is received by all other NICs in the same broadcast domain.

Table 6-2 Benefits of Segmenting Ethernet Devices Using Hubs, Switches, and Routers

Feature	Hub	Switch	Router
Greater cabling distances are allowed	Yes	Yes	Yes
Creates multiple collision domains	No	Yes	Yes
Increases bandwidth	No	Yes	Yes
Creates multiple broadcast domains	No	No	Yes


Distribution Layer
2 x 10 GbE
D1
D1
2 Distribution Switches
GigE
GigE
A1
A2
.....
A3
A4
40 Access Switches
Access Layer
Uplinks
≈ 1000 PCs
10/100/1000
10/100/1000
10/100/1000
10/100/1000