Computer Networks

Chapter 6 – The Link Layer and LANs

Edition8-1

Link layer and Physical layer

Link and Physical Layers provide:

- A network-wide communication service between any two network hosts (from source to destination)
- Datagrams travel over a series of communication links, some wired and some wireless, starting at source host, passing through a series of packet switches (switches and routers) and ending at destination host

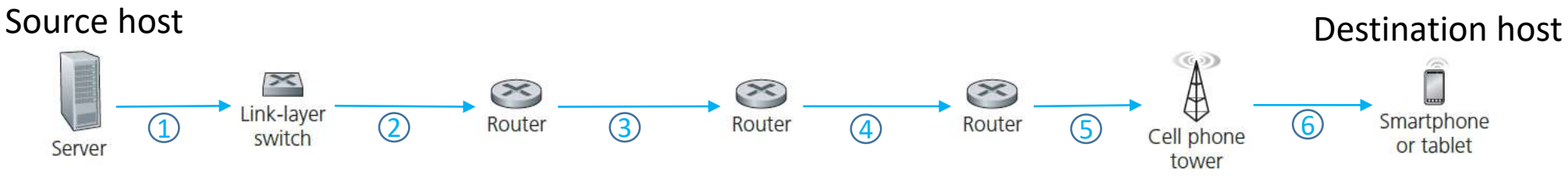


Figure 6.0 A series of communication links from source host to destination host

Link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

What we will learn

- How packets are sent across individual links
- How are network-layer datagrams encapsulated in link-layer frames for transmission over a single link?
- Are different link-layer protocols used in different links along communication path?
- How are transmission conflicts in broadcast links resolved?
- Is there addressing at link layer and, if so, how does link-layer addressing operate with network-layer addressing?
- What exactly is difference between a switch and a router?

Types of link-layer channel

- Two fundamentally different **types of link-layer channels**
  - Broadcast channels, which connect multiple hosts in wireless LANs, in satellite networks, and in hybrid fiber-coaxial cable (HFC) access networks
    - **Many hosts** are connected to **same broadcast** communication channel
    - A **medium access protocol** is needed to **coordinate frame transmission**
  - Point-to-point communication link, such as that often found between two routers connected by a long-distance link, or between a user's computer and Ethernet switch
    - Coordinating access to a **point-to-point** link is **simpler** than **broadcast** link

Contents

6.1 Introduction to the Link Layer

- Node: any device that runs a link-layer protocol (hosts, routers, switches, things, and WiFi access points)
- Communication channel (link): physical connection of adjacent nodes
- Example: a datagram from one of wireless hosts to one of servers, **Six links:**

1. a WiFi link between source host and WiFi access point
2. an Ethernet link between access point and a link-layer switch
3. Ethernet link between link-layer switch and router
4. link between two routers
5. Ethernet link between router and link-layer switch
6. Ethernet link between switch and server

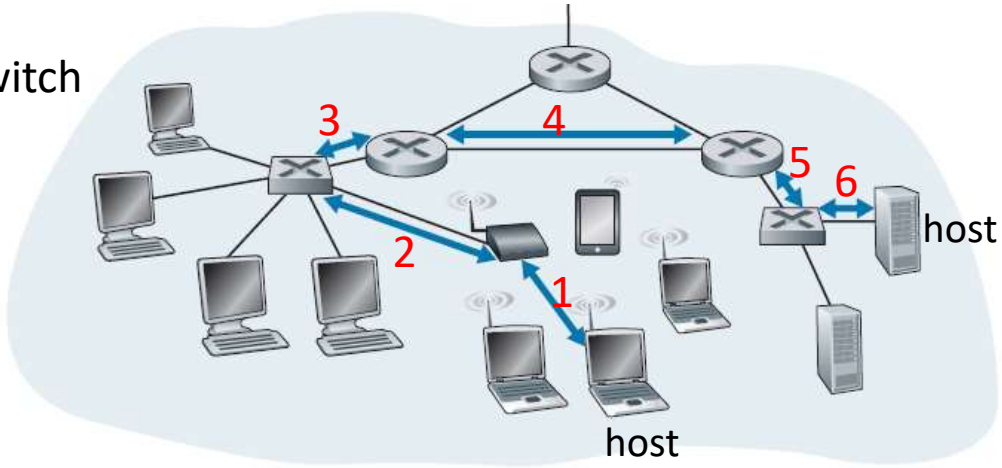


Figure 6.1 Six link-layer hops between wireless host and server

6.1.1 The Services Provided by the Link Layer

- Link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

1. Framing
   - Encapsulate each **network-layer datagram** within a **link-layer frame** before transmission over link
   - A frame consists of a data field, in which network-layer datagram is inserted, and a number of header fields
   - Structure of frame is specified by link-layer protocol
2. Link access
   - A **medium access control (MAC) protocol** specifies **rules** by which a frame is transmitted onto link
   - For point-to-point links that have a single sender at one end of link and a single receiver at other end of link, MAC protocol is simple (or nonexistent), sender can send a frame whenever link is idle
   - More interesting case is when multiple nodes share a **single broadcast link**, multiple access problem
     - MAC protocol serves to coordinate **frame transmissions of many nodes**

Services Provided by the Link Layer

3. Reliable delivery

- When a link-layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram **across a link without error**
- TCP also provide a reliable delivery service (end-to-end)
- Reliable delivery service can be achieved with acknowledgments and retransmissions
- This service is often used for links that are prone to high error rates, such as a **wireless link**, with goal of correcting an error locally
- It can be considered an unnecessary overhead for low bit-error links, including **fiber**, **coax**, and many **twisted-pair copper** links
  - Many **wired link-layer protocols** do not provide a reliable delivery service

Services Provided by the Link Layer

4. Error detection and correction
   - Bit errors are introduced by **signal attenuation** and **electromagnetic noise**
   - Many link-layer protocols provide a mechanism to detect such bit errors
     - error-detection bits included in frame
   - Error detection in link layer is usually more sophisticated and is **implemented in hardware**
   - Error correction is similar to error detection, except that a receiver not only detects when bit errors have occurred in frame but also **determines exactly where in frame the errors have occurred** (and then corrects these errors)

6.1.2 Where Is the Link Layer Implemented?

- It is integrated into motherboard chipset or implemented via a low-cost dedicated chip, called **network adapter** or **network interface controller (NIC)**
- NIC implements many link layer services including **framing**, **link access**, **error detection**, and so on
- Intel's 700 series adapters implements Ethernet protocols; Atheros AR5006 controller implements 802.11 WiFi protocols

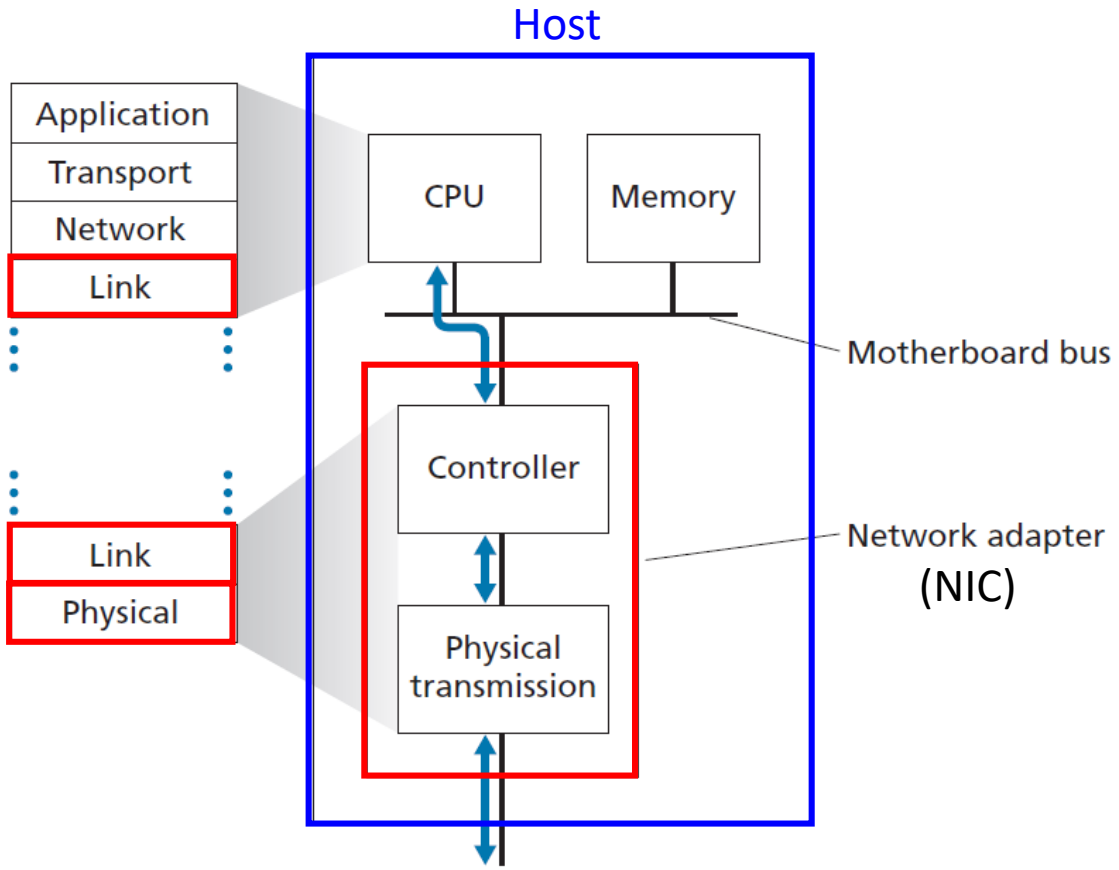


Figure 6.2 Network adapter: Its relationship to other host components and to protocol stack functionality

Where Is the Link Layer Implemented?

- **Part of link layer is implemented in software** that runs on host's CPU
- Software components of link layer implement higher-level link-layer functionality such as assembling link-layer addressing information and activating controller hardware (NIC driver)
- On receiving side, **link-layer software** responds to controller interrupts (for example, due to receipt of one or more frames), handling error conditions and passing a datagram up to network layer
- Link layer is a place in protocol stack where **software meets hardware**

Link layer in action

- Sending side:
  - Controller takes a datagram that has been created and stored in host memory by higher layers of protocol stack
  - Encapsulates datagram in a link-layer frame
  - Transmits frame into communication link, following link-access protocol
- Receiving side:
  - Controller receives entire frame
  - Performs error detection
  - Examine destination MAC address in received packet
  - Extracts network layer datagram and delivers it to network layer

Contents

6.2 Error-Detection and -Correction Techniques

- At receiving, a sequence of bits, D' and EDC' is received
- D' and EDC' may differ from original D and EDC as a result of in-transit bit flips
- Even with use of error-detection, there still may be **undetected bit errors**

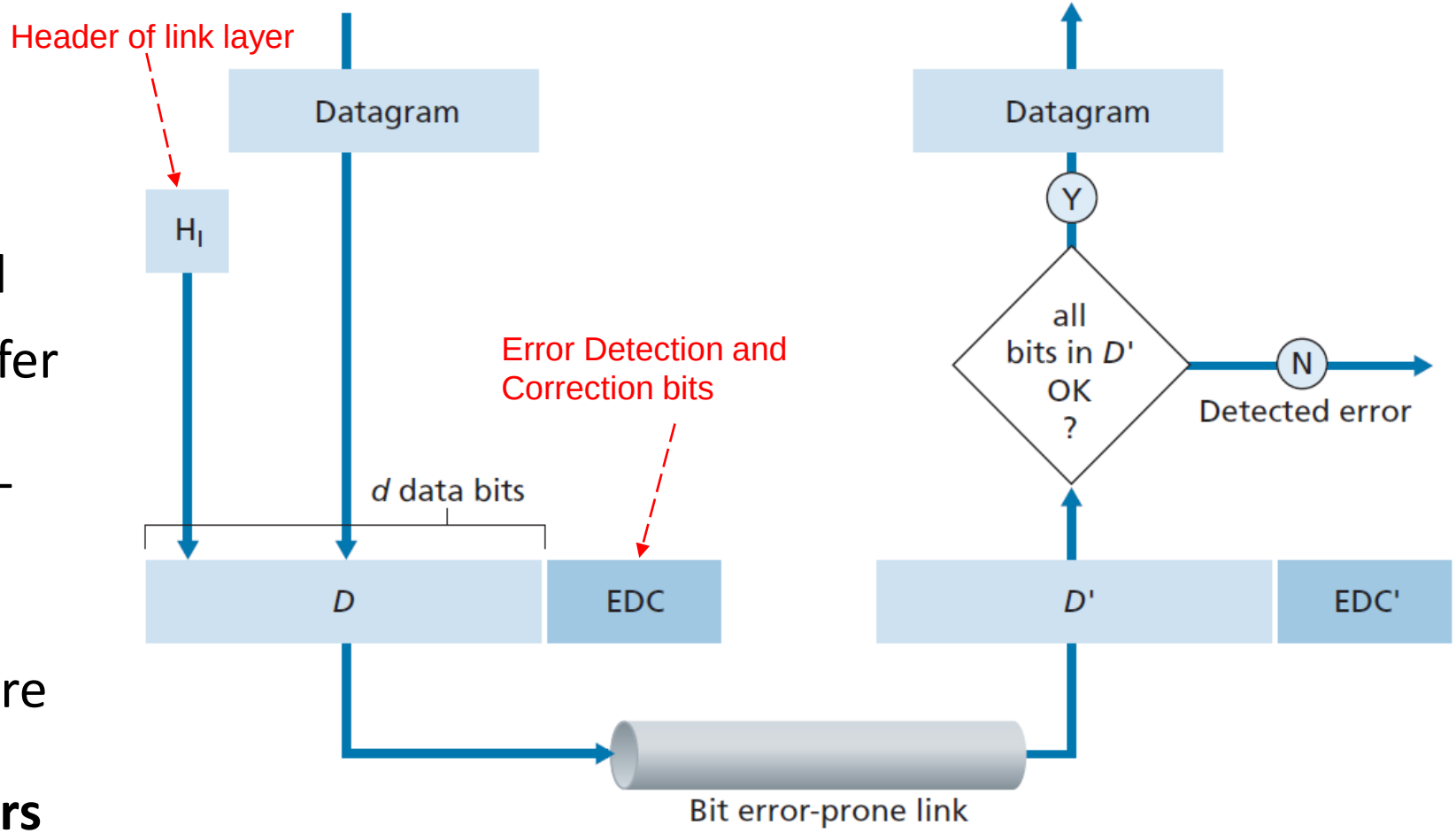


Figure 6.3 Error-detection and -correction scenario

6.2.2 Checksumming Methods

- Checksumming techniques: d bits of data are treated as a sequence of k-bit integers

- A checksumming method: **Sum** k-bit integers and use resulting sum as error-detection bits
- **Internet checksum (TCP, UDP, IP)** is based on this approach, k=16-bit, 1s complement of this sum is checksum that is carried in header
- Internet checksum provides relatively weak protection against errors

Why checksum at Transport, Network and Link layer

- Because transport-layer error detection is implemented in software (OS), it is important to have a simple and fast error-detection scheme such as checksumming

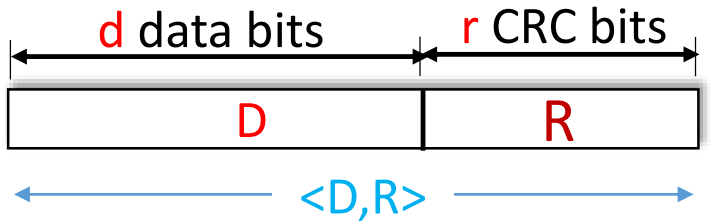
- Error detection at link layer is **implemented in dedicated hardware** in adapters, which can rapidly perform **more complex** CRC operations

6.2.3 Cyclic Redundancy Check (CRC)

- Widely used in link layer of computer networks
- CRC codes are also known as **polynomial codes**, since it is possible to view bit string as a polynomial whose coefficients are 0 and 1 values in bit string:
- $1101 \rightarrow 1x^4 + 1x^3 + 0x^2 + 1x^1$

CRC:

- D: data bits (think D is a binary number), datagram from upper layer
- G: bit pattern (generator), of r+1 bits (**given**)



<D,R> = $D \cdot 2^r$ XOR R

G_{CRC32}

Method: choose R [32 bits], such that <D,R> divisible by G (modulo 2)

- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected
- widely used in practice (Ethernet, 802.11 WiFi)

CRC-32 International standard:

- G_{CRC32}=100000100110000010001110110110111 → r+1=33 bits
- Can detect burst errors of fewer than 33 bits
- Under appropriate assumptions, a burst of length greater than 33 with probability $1- 0.5^{32}=0.9999999998$
- Can detect any odd number of bit errors

Modulo-2 arithmetic

- Addition and subtraction are identical, and both are equivalent to bitwise exclusive-or (XOR) of operands
- Addition example

  1011 + 0101 → 1011 XOR 0101 = 1110

  1001 + 1101 → 1001 XOR 1101 = 0100
- Subtraction example

  1011 - 0101 → 1011 XOR 0101 = 1110

  1001 - 1101 → 1001 XOR 1101 = 0100
- Multiplication and division: same as in base-2 arithmetic, except any addition or subtraction is done without carries or borrows

CRC example

We want: $D \cdot 2^r$ XOR $R = nG$

or equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R: $R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$

In Sender: Frame=<D,R>= 101110011

In Receiver: R[<D,R> / G] = 0

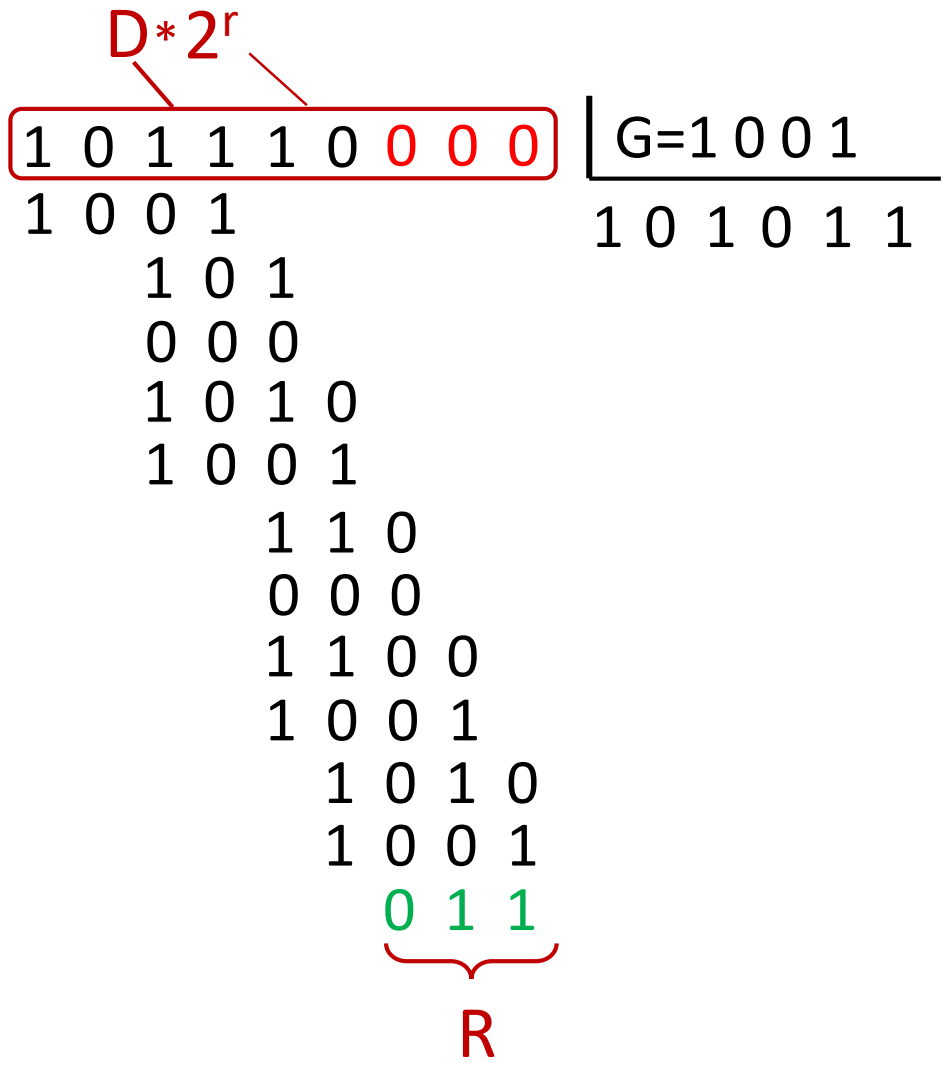


Figure 6.7 A sample CRC calculation, r=3, G=1001

Contents

6.3 Multiple Access Links and Protocols

Two types of network links

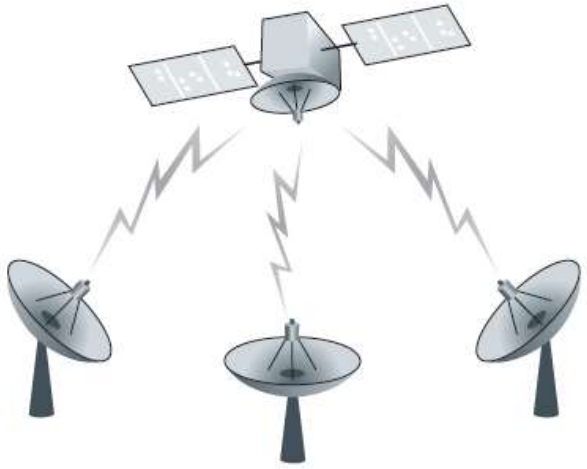
- **Point-to-point link:**
  - A single sender at one end of link and a single receiver at other end of link
  - Protocols for point-to-point links: PPP (point-to-point protocol), HDLC (high-level data link control)
- **Broadcast link**:
  - Multiple sending and receiving nodes all connected to same, single, shared broadcast link (channel)
  - Broadcast: any node transmits a frame then each other nodes receives a copy
  - Protocols for broadcast links: Ethernet, wireless LANs
    - (LAN: network that are geographically concentrated in a single building or on a corporate or university campus)
  - Multiple access problem: how to coordinate access of multiple sending and receiving nodes to a shared broadcast channel

Figure 6.8 Various multiple access channels

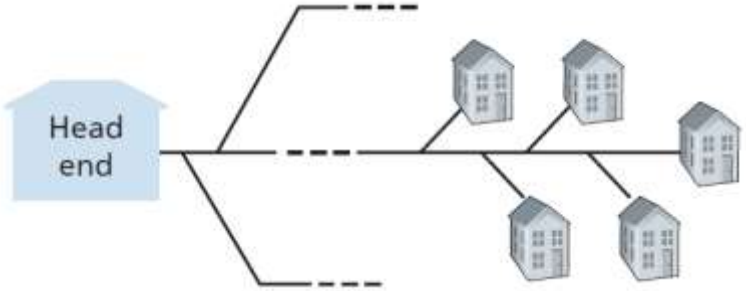
If more than one node transmit at same time:

- All of other nodes receive multiple frames at same time; transmitted frames **collide** at all of receivers, none of receiving nodes can make any sense of any of frames that were transmitted, frames involved in collision **are lost**

Shared wireless (e.g., WiFi)



Satellite



Shared wire (e.g., cable access network)

Ideal multiple access protocol

For a broadcast channel of rate R bps

1. When only one node has data to send, that node has a throughput of R bps
2. When M nodes have data to send, each of these nodes has a average throughput of R/M bps
3. Protocol is decentralized; that is, there is no master node that represents a single point of failure for network
4. Protocol is simple, so that it is inexpensive to implement

Categories of multiple access protocols

- Channel partitioning
  - Divide channel into smaller “pieces” (time slots, frequency, code)
  - Allocate a piece to a node for exclusive use
- Random access
  - Channel not divided, a node transmits at channel capacity rate R, collision is possible
  - “recovery” from collisions should be implemented
- Taking turns
  - Nodes take turns, but nodes with more to send can take longer turns

6.3.1 Channel Partitioning Protocols

- Time-division multiplexing (TDM)
  - Whenever a node has a packet to send, it transmits during its assigned time slot
  - Typically, slot sizes are chosen so that a single packet can be transmitted during a slot time
  - Advantage: No collisions and perfectly fair
  - Drawbacks:
    - a node is limited to an average rate of R/N bps even when it is only node with packets to send
    - a node must always wait for its turn in transmission sequence, even when it is only node with a frame to send
- Frequency-division multiplexing (FDM)
  - It creates N smaller channels of R/N bps out of single R bps channel
  - FDM shares both advantages and drawbacks of TDM

6.3.2 Random Access Protocols

- There are dozens of random access protocols
- We describe ALOHA protocols and Carrier Sense Multiple Access Collison Detection (CSMA/CD) protocols
  - Ethernet is a popular and widely deployed CSMA/CD protocol

Slotted ALOHA

- All frames consist of exactly L bits
- Time is divided into slots of size L/R seconds (transmit one frame)
- Nodes start to transmit frames only at beginnings of slots
- Each receiving node knows when slots begin (synchronized)

Slotted ALOHA

Operation of slotted ALOHA:

- When node has a fresh frame to send, it waits until beginning of next slot and transmits entire frame
- If there isn't a collision, node has successfully transmitted its frame
- If there is a collision, node detects collision before end of slot
- Node retransmits its frame in each subsequent slot with probability p until frame is transmitted without a collision

Advantages:

- a node transmits continuously at full rate, R, when it is only node with packets to send
- highly decentralized, a node detects collisions and independently decides when to retransmit, but require slots to be synchronized
- It is also an extremely simple protocol

Efficiency of Slotted Aloha

- Efficiency: Suppose there are many nodes, all with many frames to send
- N nodes, each retransmits in slot with probability p
- Probability a node has success in a slot = $p(1-p)^{N-1}$
- Efficiency: Probability any node has a success = $Np(1-p)^{N-1}$
- Max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- Taking limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives: Max efficiency=1/e=0.37
- At best: channel used for useful transmissions 37% of time

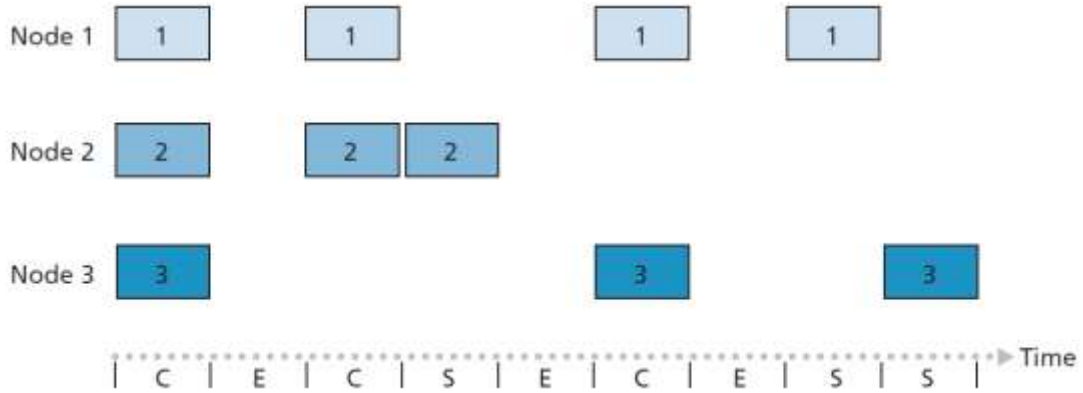


Figure 6.10 Nodes 1, 2, and 3 collide in first slot. Node 2 finally succeeds in fourth slot, node 1 in eighth slot, and node 3 in ninth slot

ALOHA

- When a frame gets ready in sender's link layer, node immediately transmits
- If collision:
  - Node will immediately (after completely transmitting its collided frame) retransmit frame with probability p
  - Otherwise, node waits for a frame transmission time. After this wait, it then transmits frame with probability p, or waits (remaining idle) for another frame time
- Frame sent at t_0 collides with other frames sent in $[t_0\text{-}1, t_0\text{+}1]$
- Probability that all other nodes do not begin a transmission in intervals of time $[t_0 - 1, t_0]$ and $[t_0, t_0\text{-}1]$=

$$=(1 - p)^{N-1} * (1 - p)^{N-1}$$

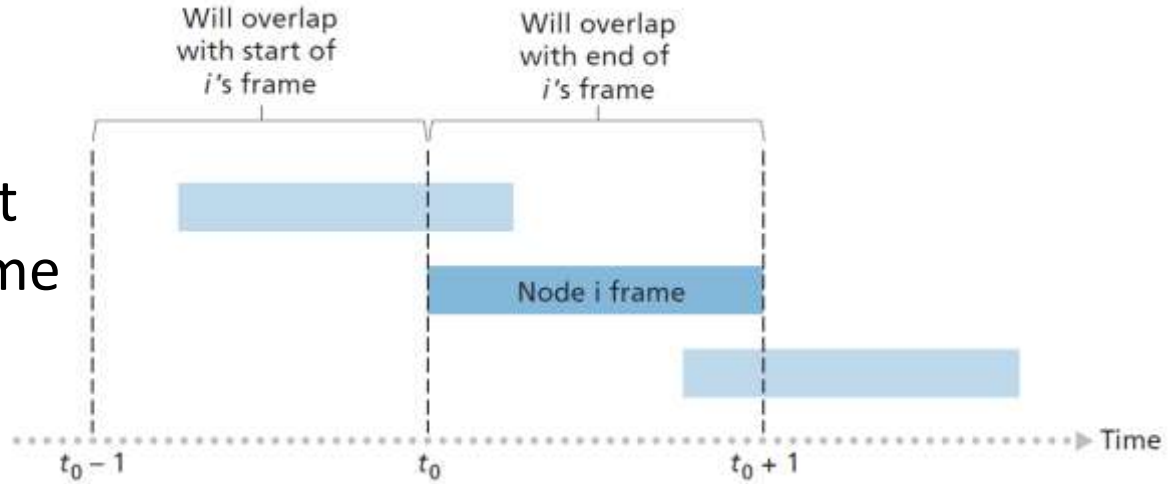


Figure 6.11 Interfering transmissions in pure ALOHA

Efficiency of Aloha

- Thus, probability that a given node has a successful transmission=

  $=p(1 - p)^{N-1} *(1 - p)^{N-1}=p(1 - p)^{2(N-1)}$

- Probability that any node has a successful transmission=$Np(1 - p)^{2(N-1)}$

- Maximum efficiency of ALOHA protocol is only $1/(2e)$, exactly half that of slotted ALOHA
- This is the price to be paid for a fully decentralized (no requirement for slots to be synchronized) ALOHA protocol

Carrier Sense Multiple Access (CSMA)

- Carrier Sense: **Listen before transmit**
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy: defer transmission
- Collisions can still occur with carrier sensing:
  - At time t_1, D senses channel is idle, then, starts to transmit, but, collision occurs
  - Distance & Propagation delay play role in collision probability

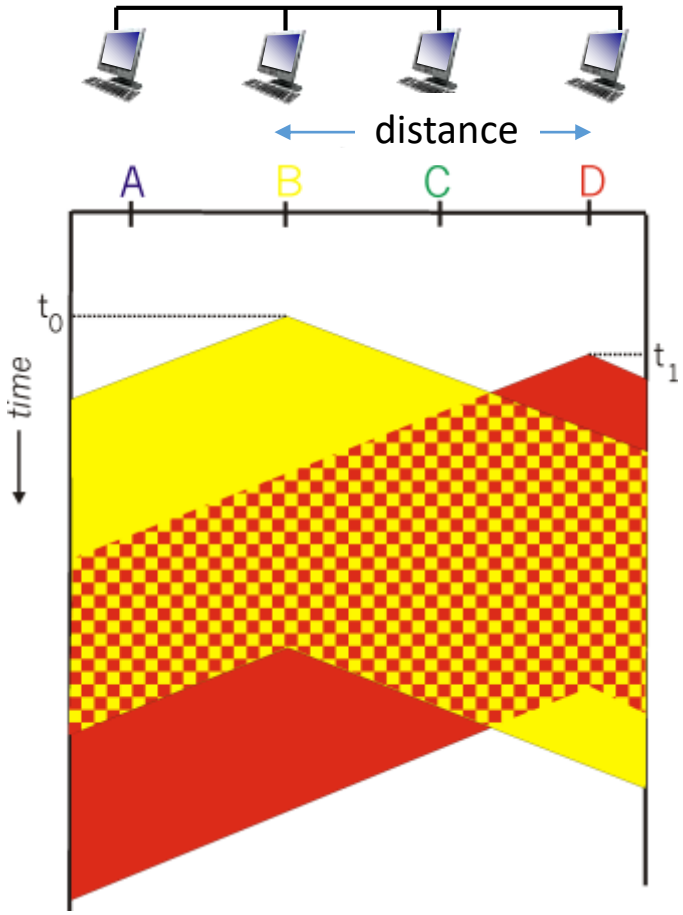


Figure 6.12 Space-time diagram of two CSMA nodes with colliding transmissions

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- CSMA/CD:
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing channel wastage
  - Collision detection easy in wired, difficult with wireless

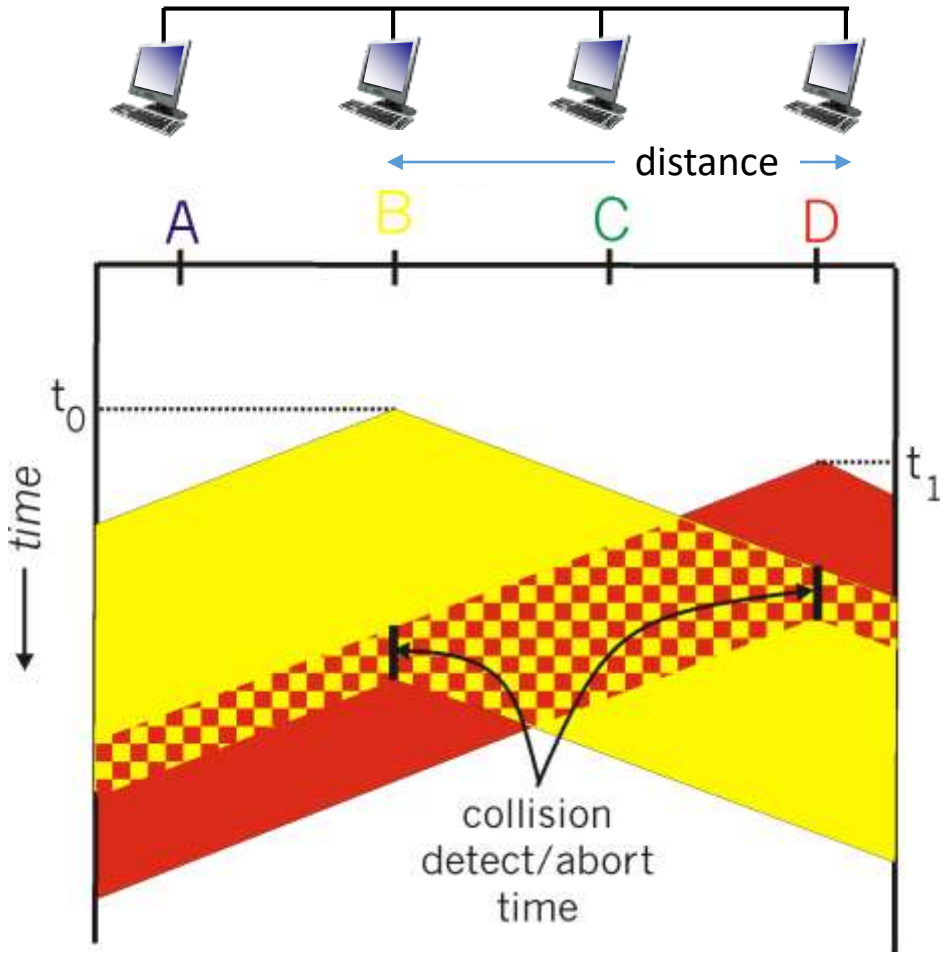
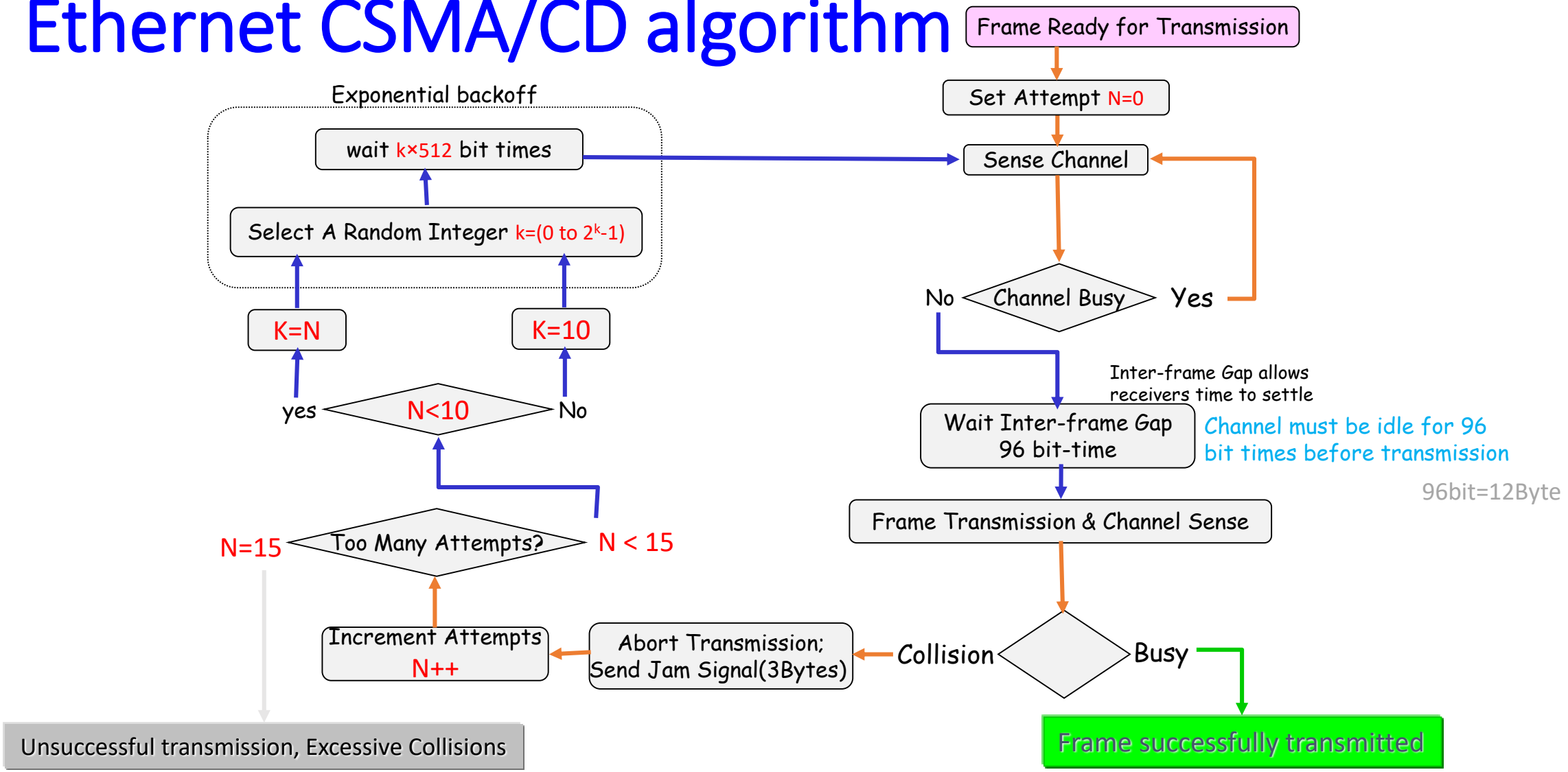


Figure 6.13 CSMA with collision detection

Ethernet CSMA/CD algorithm

1. Ethernet NIC receives datagram from network layer, creates frame
2. Ethernet NIC senses channel:
    1. if idle: start frame transmission
    2. if busy: wait until channel idle, then transmit
3. If NIC transmits frame without collision, NIC is done with frame
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters binary (exponential) back off:
    - after N_{th} collision, NIC chooses K at random from { 0,1,2, …, $2^N - 1$ }
    - NIC waits K*512 bit times, returns to Step 2 (512 bit times=0.512 µs for a 1000 Mbps Ethernet)

Ethernet CSMA/CD algorithm

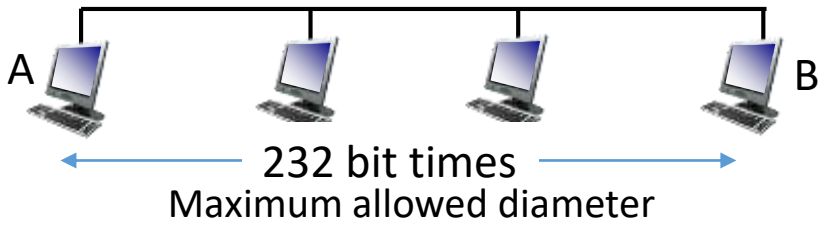


Maximum Jam-Time

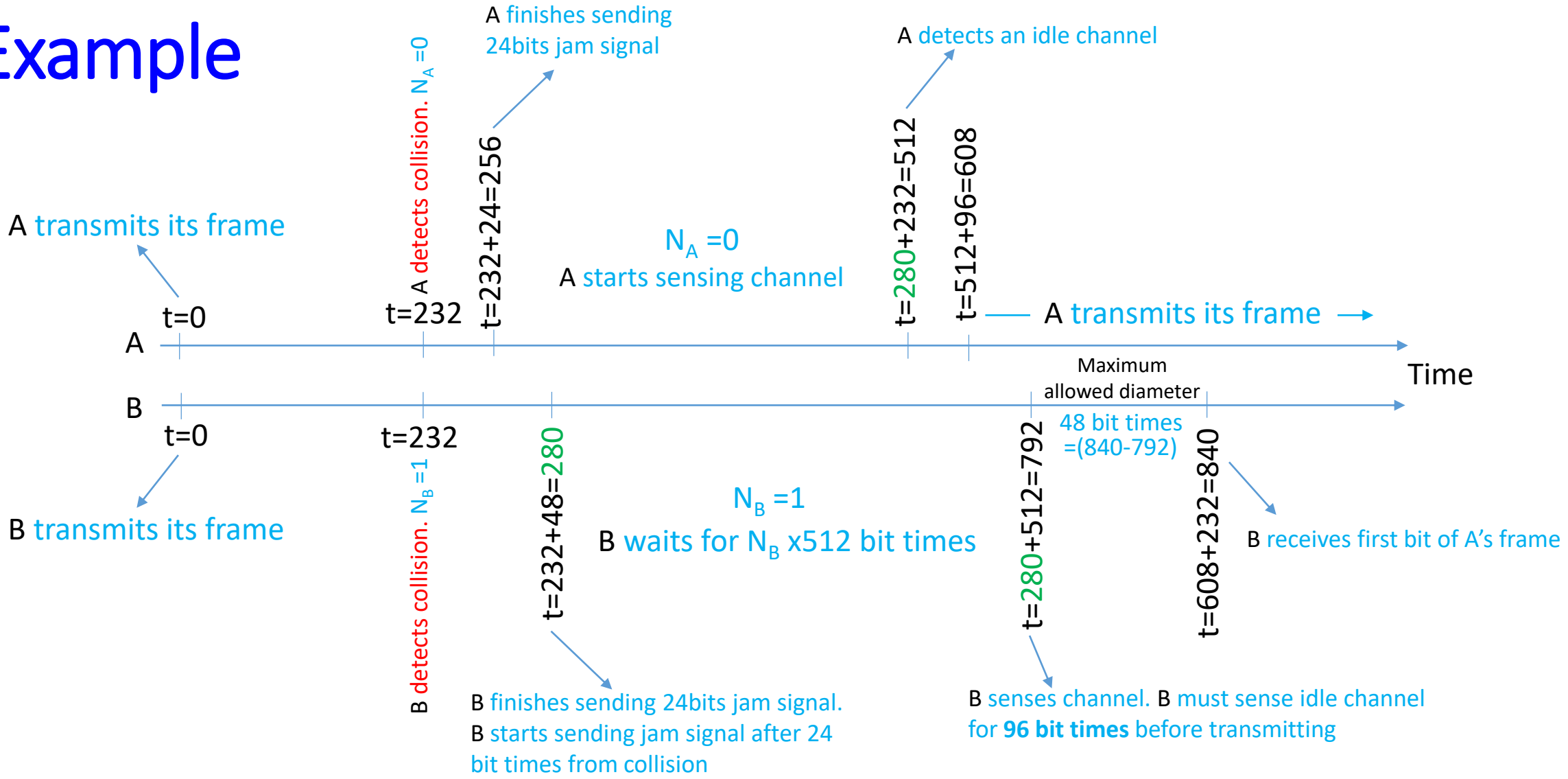
Maximum jam-time is calculated as follows:

- Maximum allowed diameter of an Ethernet installation is limited to 232 bit times (1 bit time=1/R)
  - This makes a round-trip-time of 2*232=464 bit times
- As slot time in Ethernet is 512 bit times, difference between slot time and round-trip-time is maximum "jam-time“:

Maximum jam-time = 512 – 2*232 = 48 bit times = 6 Byte times



Example


A finishes sending
24bits jam signal
A detects an idle channel
A transmits its frame
t=0
t=232
A detects collision. N_A =0
t=232+24=256
N_A =0
A starts sensing channel
t=280+232=512
t=512+96=608
A transmits its frame
A
Time
Maximum
allowed diameter
B
48 bit times
=(840-792)
t=0
t=232
N_B =1
B detects collision.
t=232+48=280
N_B =1
B waits for N_B x512 bit times
t=280+512=792
t=608+232=840
B transmits its frame
B receives first bit of A's frame
B finishes sending 24bits jam signal.
B starts sending jam signal after 24
bit times from collision
B senses channel. B must sense idle channel
for **96 bit times** before transmitting

CSMA/CD Efficiency

- t_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}} \equiv \frac{1}{\text{Transmission Rate (bps)}} \times \frac{1}{\text{LAN Size (m)}}$$

Frame Size (bits) / Transmission Rate (bps)

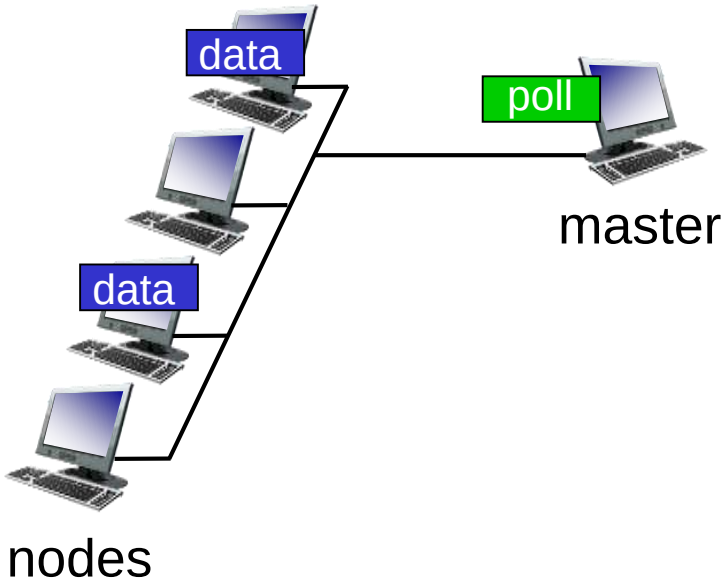
LAN Size (m) / Energy Propagation Velocity (m/s)

- efficiency goes to 1
  - as t_{prop} goes to 0 (very small LAN)
  - as t_{trans} goes to infinity (very low transmission rate)
- better performance than ALOHA, and simple, cheap, decentralized

6.3.3 Taking-Turns Protocols

1 - Polling:

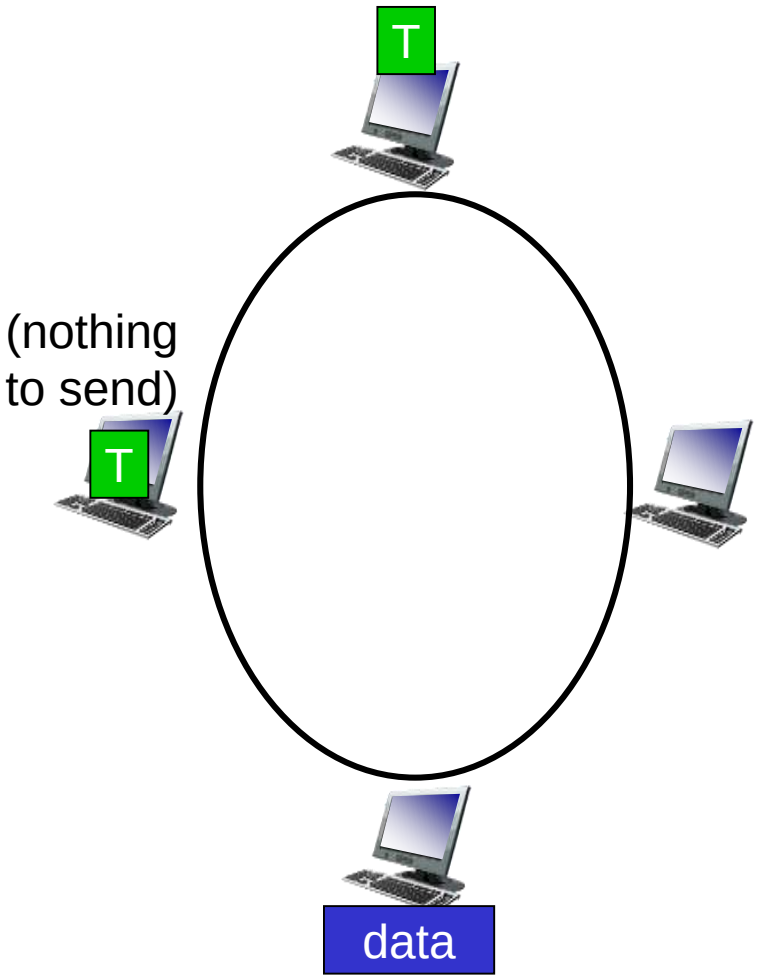
- Master node sends “invite” message to other nodes to transmit in turn (round robin). Invitation: you can transmit up to n frames
- If a node has nothing to send or finished, then master immediately poll next node
- No collision, No empty slot, Fair protocol
- Concerns:
  - Delay induced by polling
  - Single point of failure (master)



Taking-Turns Protocols

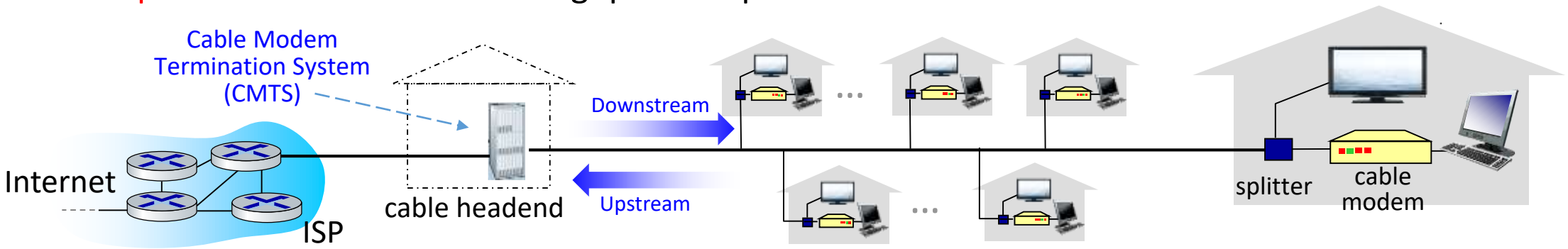
2- Token passing:

- A small, special-purpose frame known as a **token** is exchanged among nodes in some fixed order
- Node receives a token:
  - Sends maximum number of frames and forward token to next node
  - If has no frame, immediately forwards token to next node
- Fiber Distributed Data Interface (FDDI) protocol, and IEEE 802.5 Token Ring protocol
- Concerns:
  - Delay induced by token passing
  - Single point of failure (failure of one node can crash entire channel)



6.3.4 DOCSIS: The Link-Layer Protocol for Cable Internet Access

- Cable access network connects several thousand residential cable modems to a cable modem termination system (CMTS)
- Data-Over-Cable Service Interface Specifications (DOCSIS) specifies cable data network architecture and its protocols
- DOCSIS uses FDM to divide downstream (CMTS to modem) and upstream (modem to CMTS) network segments into multiple frequency channels
- Each downstream channel: max throughput 1.6Gbps
- Each upstream channel: max throughput 1Gbps



6.3.4 DOCSIS: The Link-Layer Protocol for Cable Internet Access

- CMTS grants permission to individual cable modems to transmit during specific mini-slots
- CMTS accomplishes this by sending a control frame known as MAP on a downstream channel to specify
  - which cable modem can transmit during which mini-slot for interval of time specified in MAP
- Since mini-slots are explicitly allocated to cable modems, there are no collision during a mini-slot

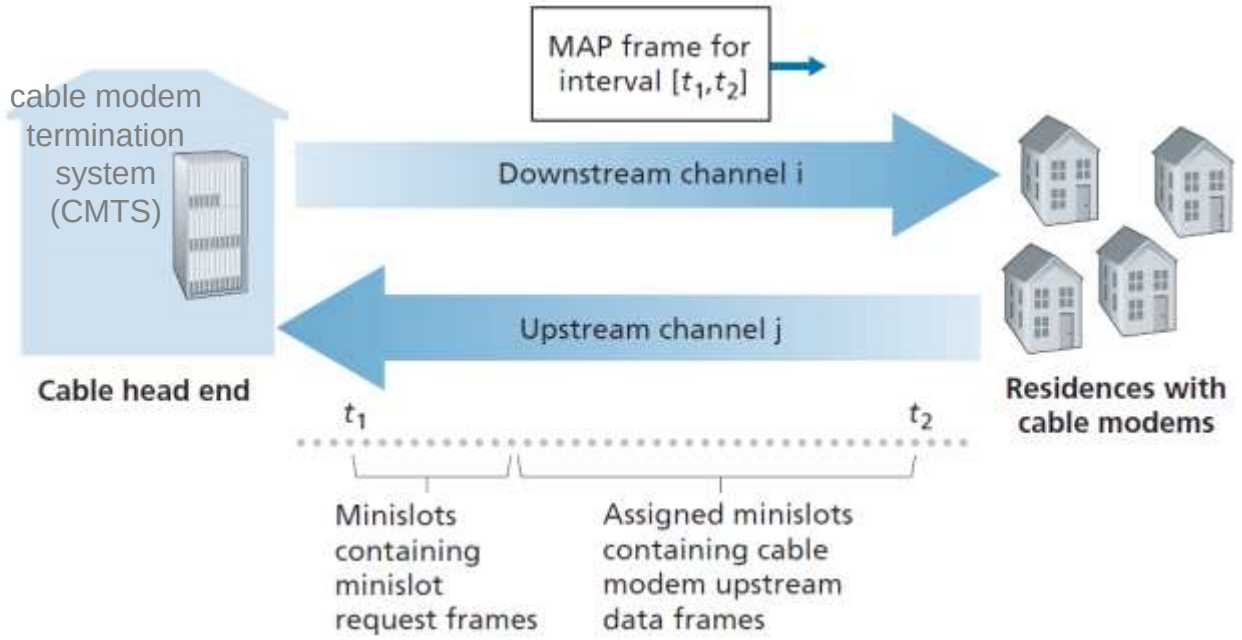


Figure 6.14 Upstream and downstream channels between CMTS and cable modems

Contents

6.4 Switched Local Area Networks

- Figure 6.15 shows a **Switched LAN** using four switches

Switches:

- Operate at link layer, they switch link-layer frames
- Don't recognize IP addresses
- Don't use routing algorithms to determine paths through network of layer-2 switches
- Use link-layer addresses to forward link-layer frames through network of switches

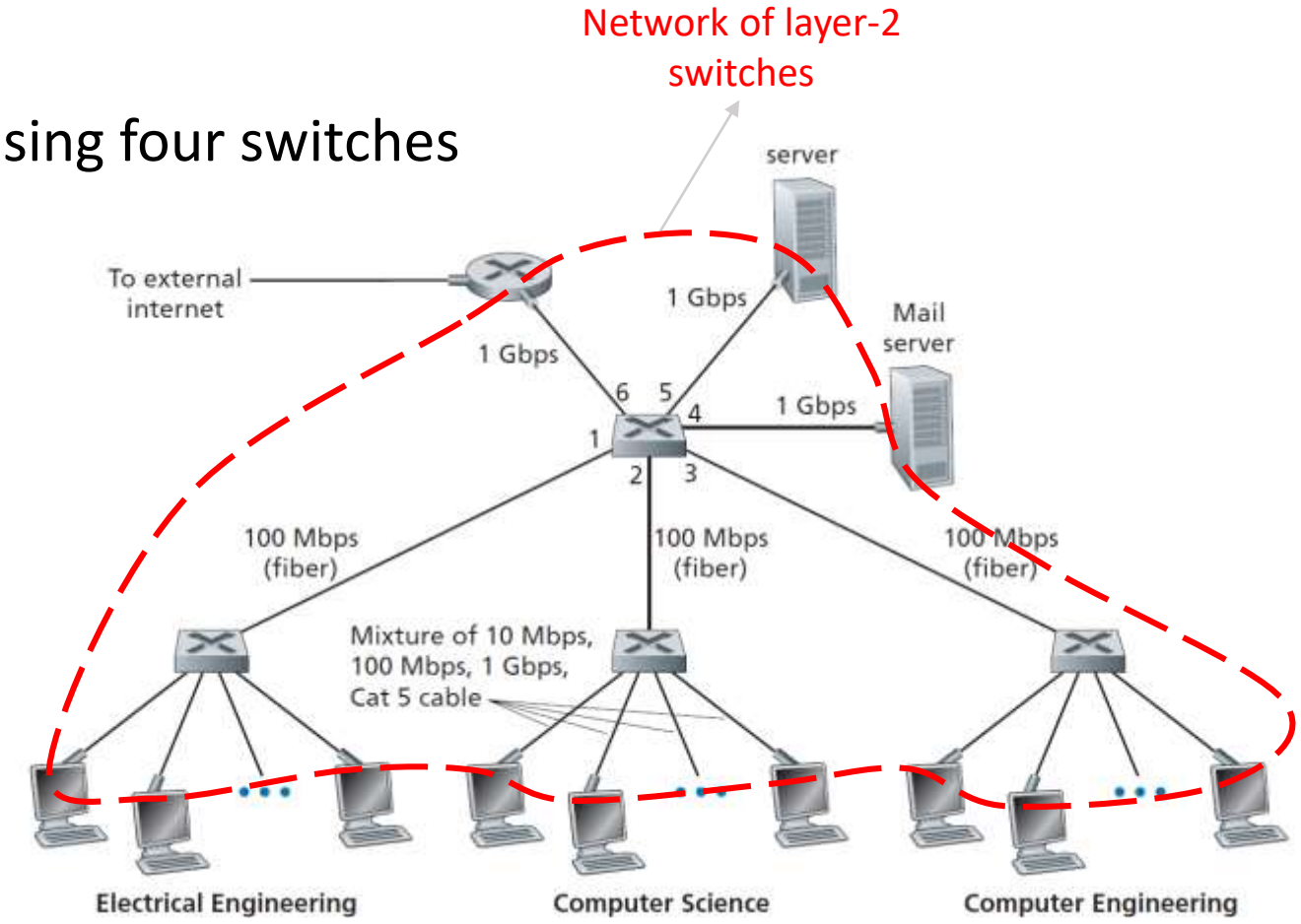


Figure 6.15 An institutional network connected together by four switches

6.4.1 Link-Layer Addressing and ARP

- Interface(s) in hosts and routers have link-layer and IP addresses
- Why do we need to have addresses at both network and link layers?
- Why two layers of addresses are useful and, in fact, necessary?

MAC Addresses

- Link layer address: LAN address, physical address, **MAC address**
- Adapters (NIC) in hosts and routers have MAC addresses
- A host or router with multiple interfaces (NICs) will have multiple MAC addresses, just as it would also have multiple IP addresses
- Switches do not have MAC and IP addresses
- Switches are transparent (host or router are not aware of their existence)
- MAC address: 6 bytes (expressed in hexadecimal)

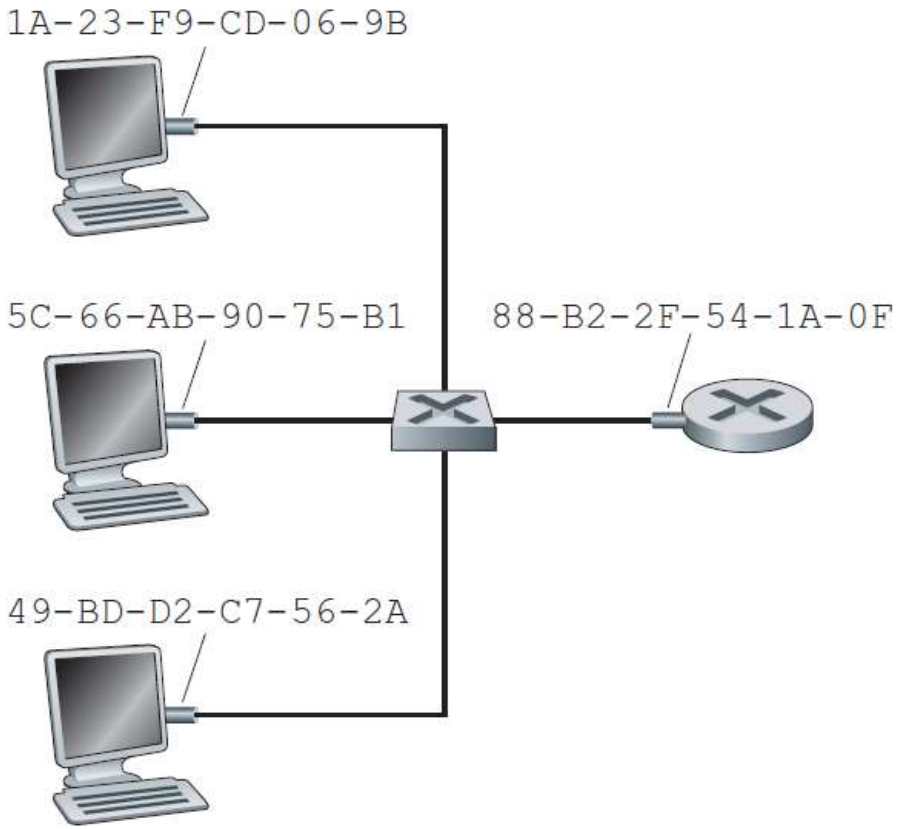


Figure 6.16 Each interface connected to a LAN has a unique MAC address

MAC Addresses

- IEEE manages MAC address space (6-Byte, 48-bit)
- When a company wants to manufacture NICs, it purchases a chunk of address space consisting of 2^{24} addresses for a nominal fee
- IEEE allocates chunk of 2^{24} addresses by fixing first 24 bits of a MAC address and letting company create unique combinations of last 24 bits for each adapter
- MAC address is burned in NIC ROM by manufacturer
- An adapter's MAC address has a flat structure
  - Portable address: a MAC address works in any network, any where

MAC Addresses

- Sending a frame to a destination adapter: Sending adapter inserts destination adapter's MAC address into frame and then sends frame into LAN
  - When an adapter receives frame, it will check to see whether destination MAC address in frame matches its own MAC address
- MAC broadcasting: Sending a frame to all NICs on LAN to receive and process
  - **Destination MAC broadcast address:** FF-FF-FF-FF-FF-FF

Address Resolution Protocol (ARP)

- Both network-layer addresses (IP addresses) and link-layer addresses (MAC addresses) are needed to have a packet moving into network
- **Q:** How to determine interface's MAC address, knowing its IP address?
- For Internet, this is job of **Address Resolution Protocol (ARP)** [RFC826]. ARP resolves an IP address to a MAC address
- ARP is analogous to DNS (resolves host names to IP addresses), but
  - DNS resolves host names for hosts anywhere in Internet
  - **ARP resolves IP addresses only for hosts and router interfaces on same subnet**

ARP-A uses ARP to find B's MAC address

A broadcasts ARP query, containing B's IP addr

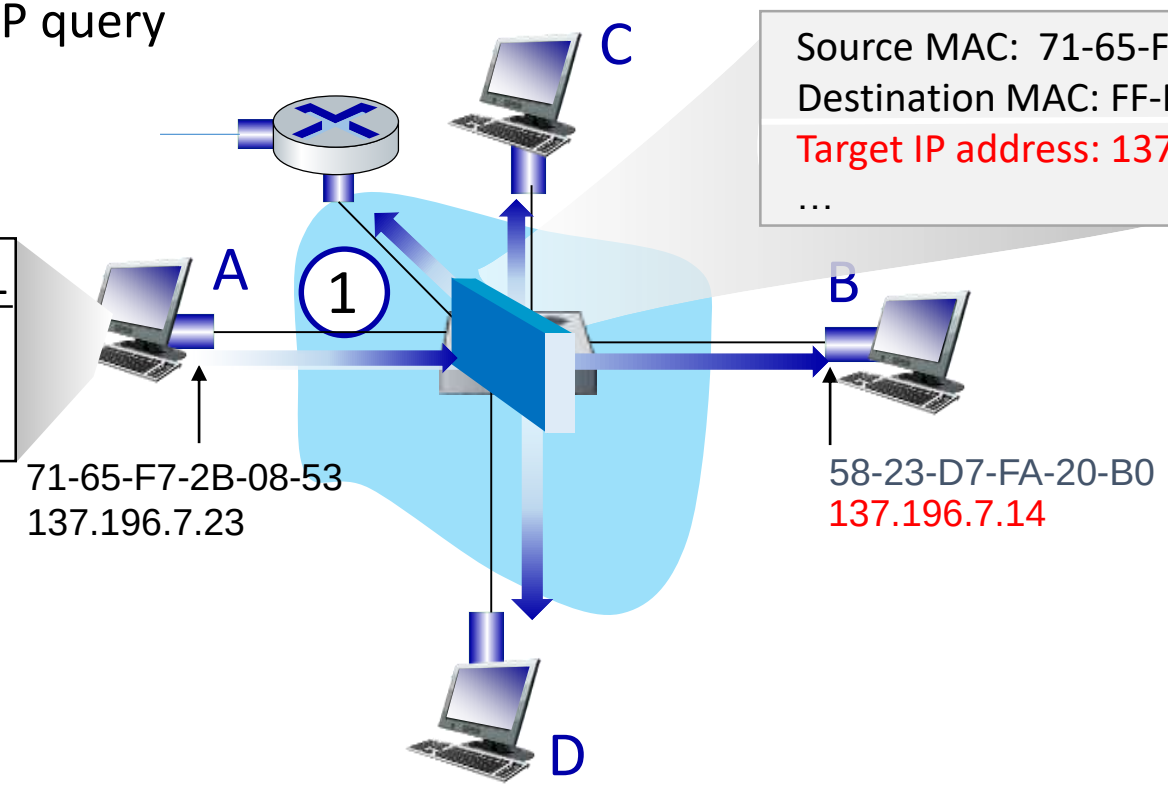
1

- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

IP addr	MAC addr	TTL

Each host and router has an **ARP table** in its memory

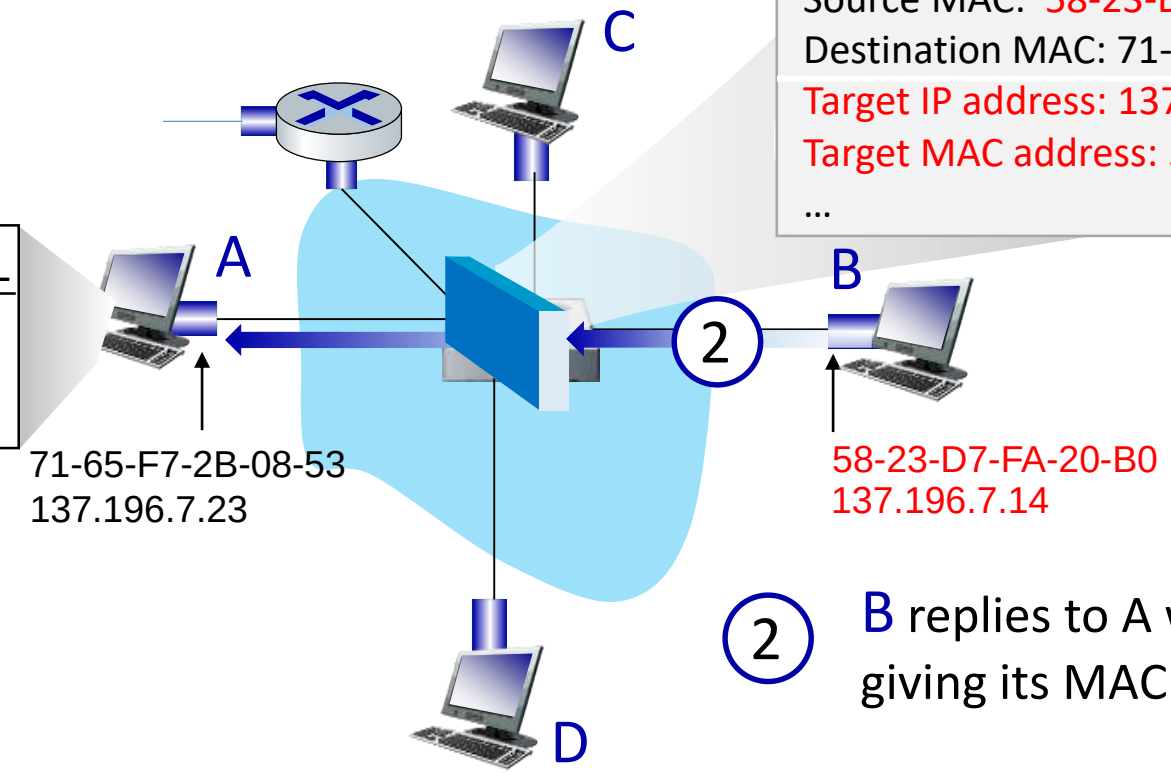


ARP-A uses ARP to find B's MAC address

What happen if B is not inside the subnet?

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	1hr



ARP message into Ethernet frame
(sent to 71-65-F7-2B-08-53)

Source MAC: 58-23-D7-FA-20-B0
Destination MAC: 71-65-F7-2B-08-53
Target IP address: 137.196.7.14
Target MAC address: 58-23-D7-FA-20-B0
…

2 B replies to A with ARP response, giving its MAC address

Some notes on ARP protocol

- ARP query message is sent within a broadcast frame, whereas ARP response message is sent within a unicast frame
- ARP is plug-and-play; that is, an ARP table gets built automatically, it doesn't have to be configured by a system administrator
- If a host becomes disconnected from subnet, its entry is eventually deleted from other ARP tables in subnet
- ARP packet is encapsulated within a link-layer frame and thus lies architecturally above the link layer

Sending a Datagram off the Subnet

- Sending a datagram from **A** to **B** via **R**
- Assume that:
  - A knows B's IP address
  - A knows IP address of first hop router, R (DHCP)
  - A knows R's MAC address (ARP)

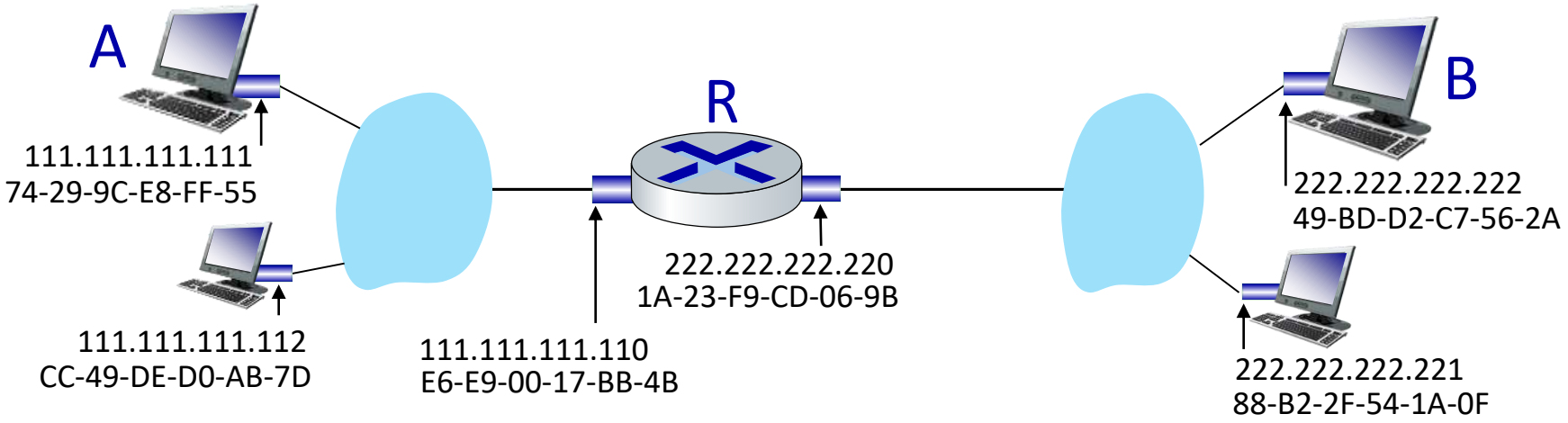
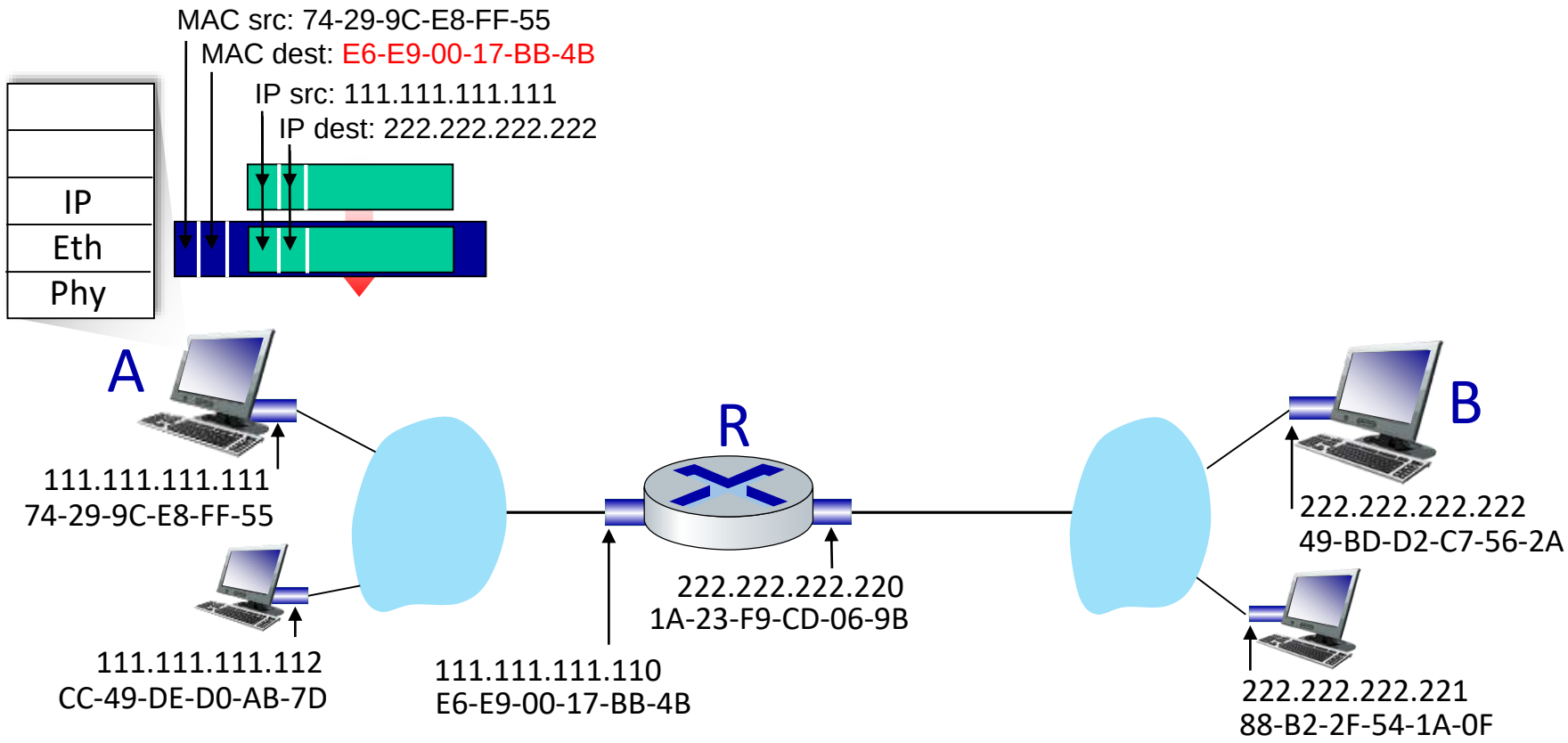
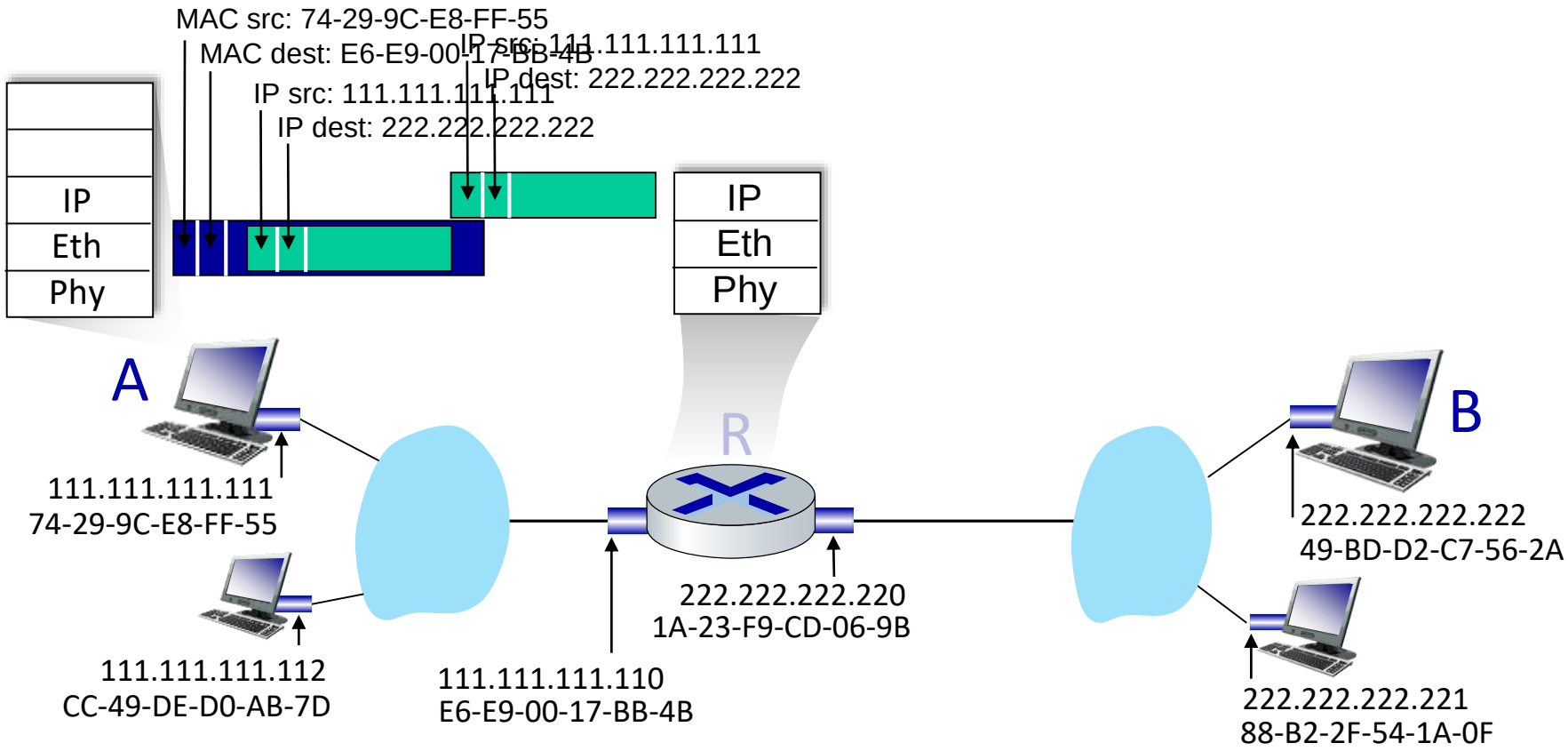


Figure 6.19 Two subnets interconnected by a router

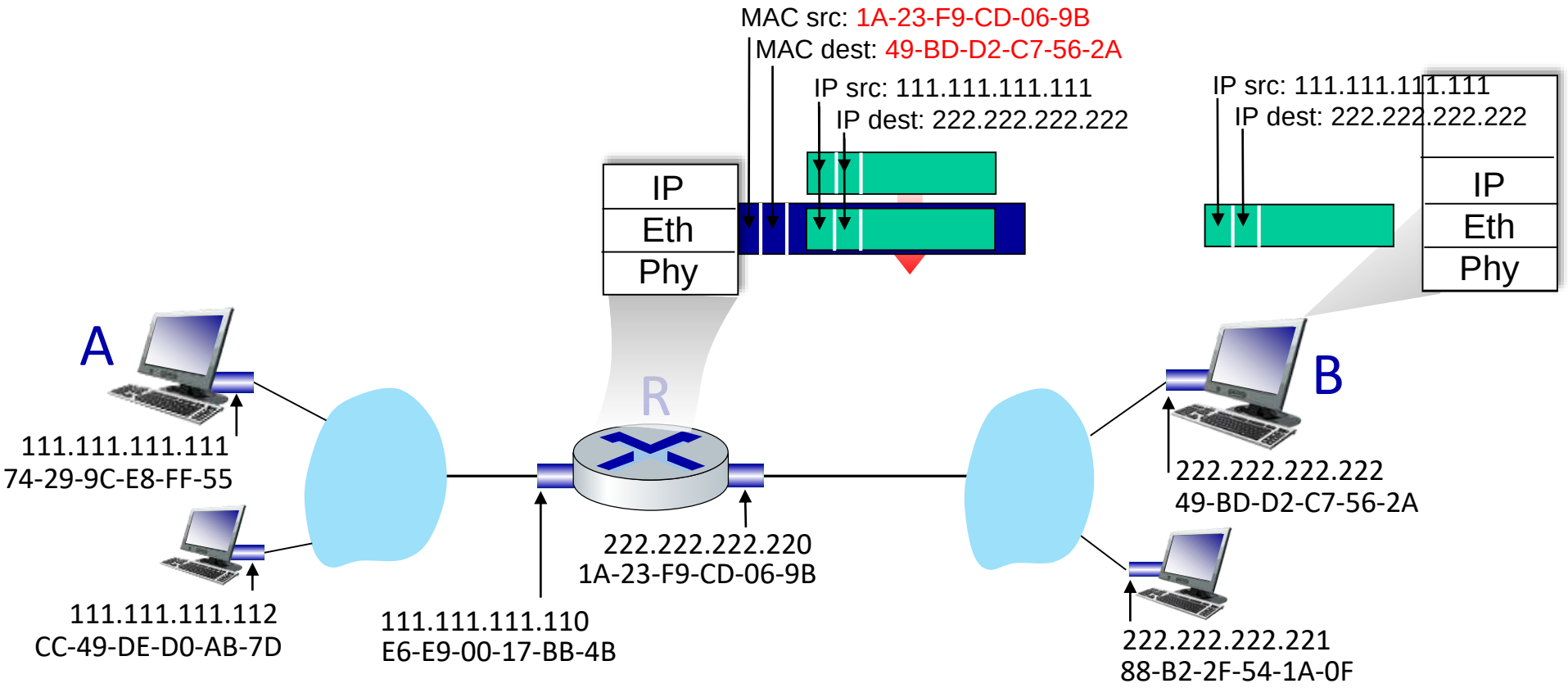
Sending a Datagram off the Subnet



Sending a Datagram off the Subnet

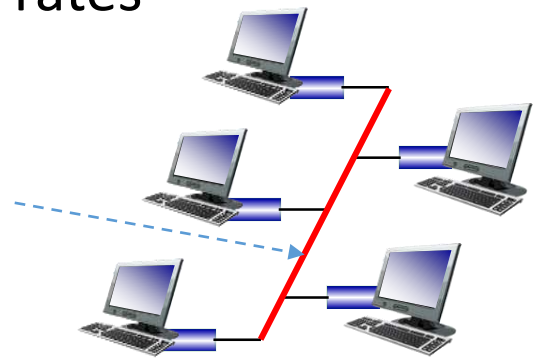


Sending a Datagram off the Subnet



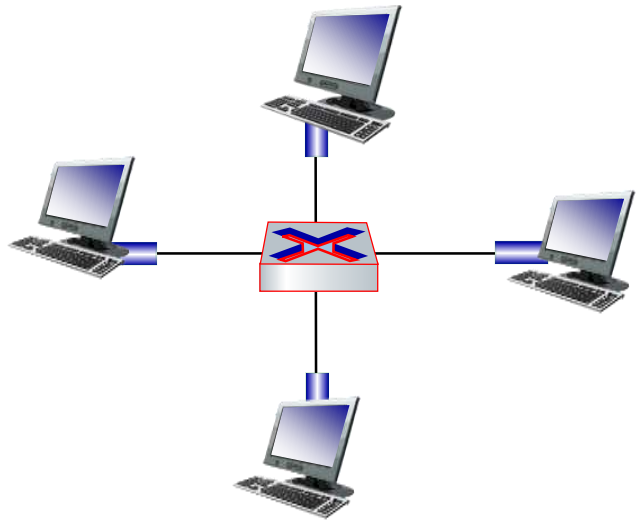
6.4.2 Ethernet

- Ethernet is by far most prevalent wired LAN technology (compare to token ring, FDDI, and ATM)
- Why?
  - Token ring, FDDI, and ATM are more complex and more expensive
  - Ethernet operates at equal data rates or higher than others
  - **Switched Ethernet** (early 1990s), increased effective data rates
- Ethernet LAN was invented in mid-1970s by Bob Metcalfe and David Boggs
  - It was based on a coaxial cable (bus) to interconnect nodes (1980s to mid-1990s), broadcast LAN, CSMA/CD



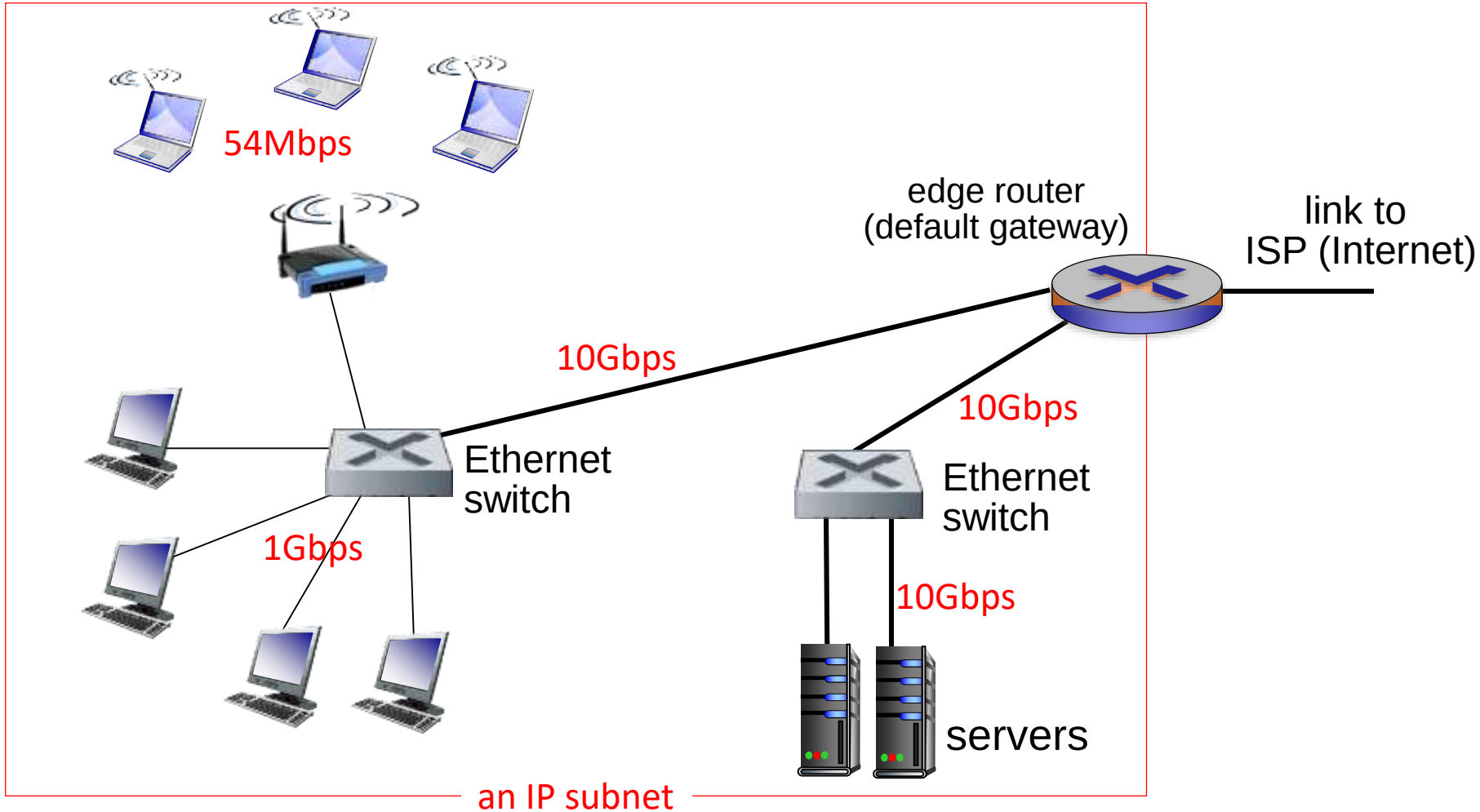
Ethernet switch

- **Ethernet switch** is a star topology local network
- Ethernet switch is not only "collision-less" but is also a store-and-forward packet switch, operates up through layer 2
- Data rate up to 400Gbps
- Single chip, multiple rates Ethernet interface (e.g., Broadcom BCM5761)

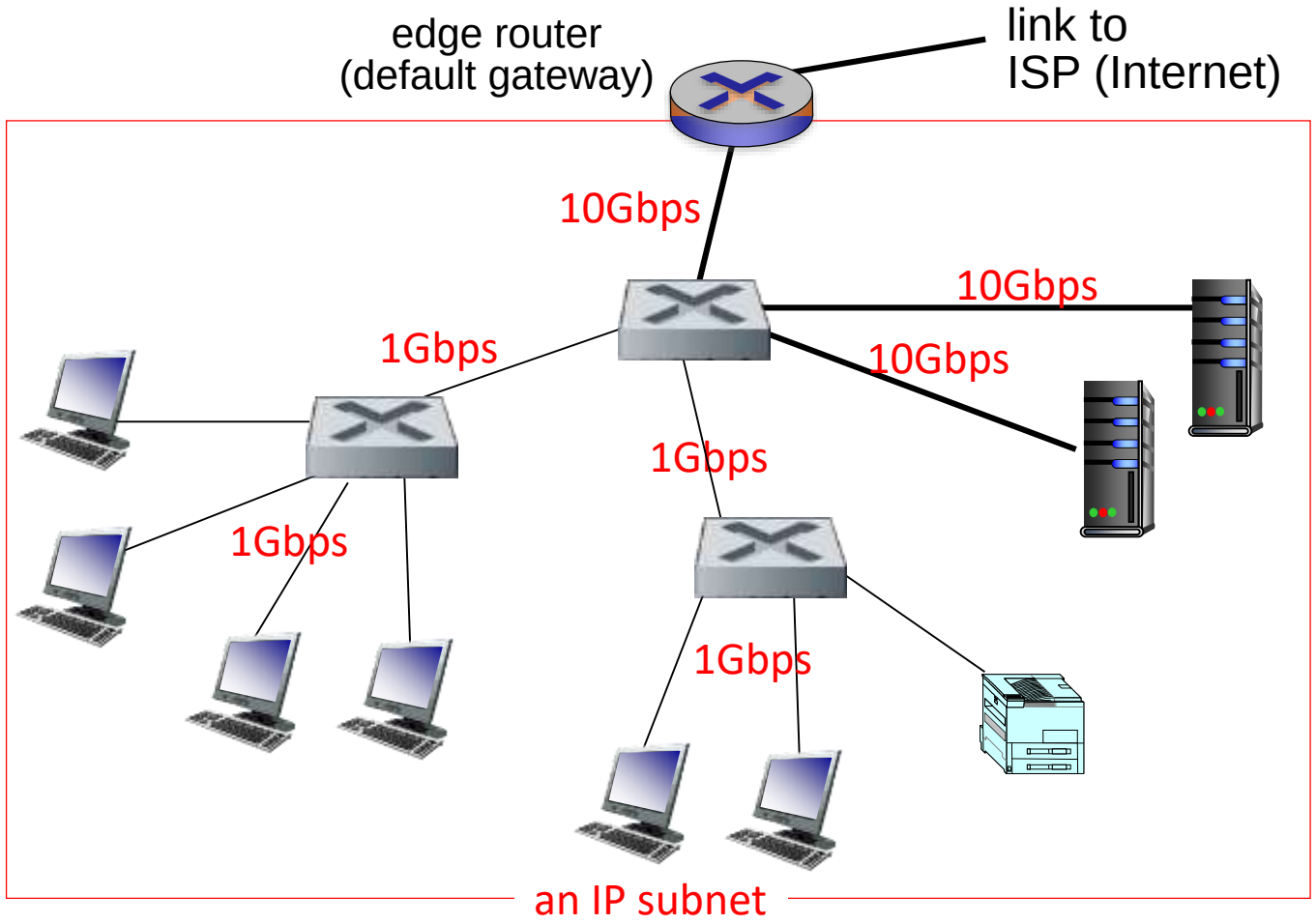


Star topology Switched Ethernet

A Switched LAN using 2 Ethernet switches



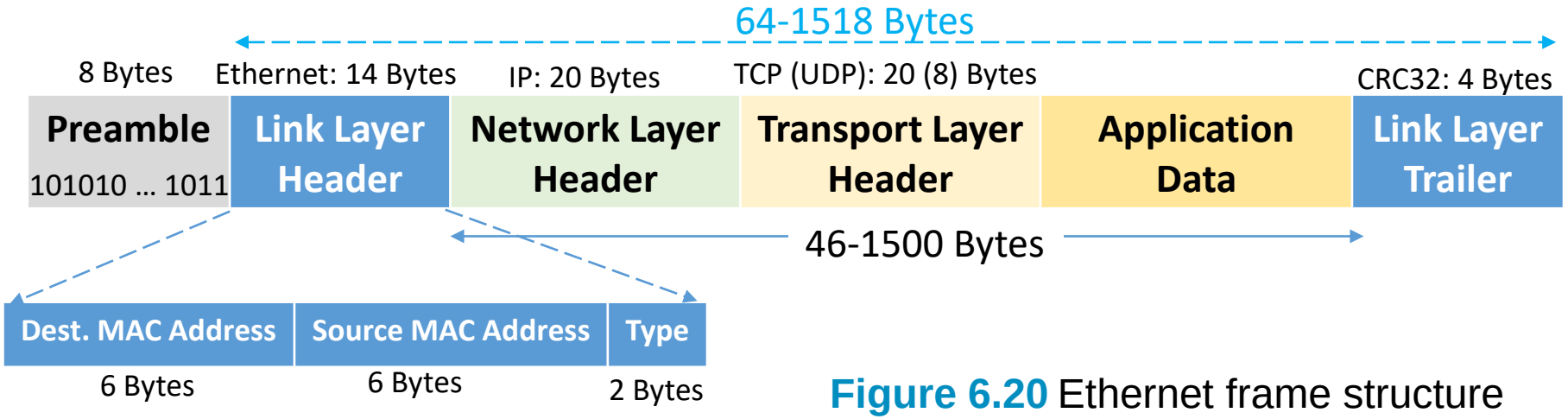
A Switched LAN using 3 Ethernet switches



Ethernet Frame Structure

Payload (46 to 1,500 bytes):

- Maximum transmission unit (MTU) of Ethernet is 1,500 bytes. Minimum size is 46 bytes
- If exceeds 1,500 bytes, host has to fragment datagram, (Section 4.3.2)
- If **less than 46 bytes**, payload has to be **"stuffed".** Data passed to receiving host network layer contains stuffing and an IP datagram. Network layer uses length in IP header to remove stuffing



Adapter uses **Type** to know to which network-layer protocol it should pass (demultiplex) payload: IP, IPX, AppleTalk, …

Figure 6.20 Ethernet frame structure

MAC (LAN, Physical) Addresses

Destination address (6 bytes)

- MAC address of Destination adapter
- When an adapter receives an Ethernet frame whose destination address is either its own MAC address or MAC broadcast address, it passes payload to network layer; if it receives a frame with any other MAC address, it discards the frame

Source address (6 bytes)

- MAC address of adapter that transmits frame onto LAN, in this example

Type field (2 bytes)

- It permits Ethernet to **multiplex network-layer protocols**
- A hosts can use other network-layer protocols besides IP
- In fact, a given host may support multiple network-layer protocols using different protocols for different applications
- Adapter uses **Type** to know to which network-layer protocol it should pass (demultiplex) payload. IP and other network-layer protocols (Novell IPX AppleTalk, …) each have their own type number
- **ARP protocol has its own type number**: **0806** hexadecimal
- Type is analogous to protocol field in network-layer datagram and port-number fields in transport-layer segment; all of these fields serve to glue a protocol at one layer to a protocol at layer above

Preamble

Preamble (8 bytes): 101010…..101011

- Each of first 7 bytes of preamble has a value of 10101010; last byte is 10101011
- **First 7 bytes** serve to "**wake up**" receiving adapters and to "**synchronize**" their clocks to that of sender's clock
  - **Synchronize:** An adapter aims to transmit frame at **10Mbps**, **100Mbps**, or **1Gbps**, depending on NIC capabilities
- Last 2 bits of eighth byte **alert receiving adapter that "frame" is about to come**

Ethernet: connectionless, unreliable

- Connectionless: Sending adapter sends frame into LAN, without first handshaking with receiving adapter (like IP and UDP)
- Unreliable: Ethernet provide an unreliable service to network layer
- Receiving NIC runs frame through a **CRC check**, but neither sends an **ACK** when a frame passes CRC check nor sends a **NAC** when a frame fails CRC check
- When a frame fails CRC check, it simply **discards frame**
  - **If application is using UDP, application in receiving host will see gaps in data**
  - If application is using TCP, TCP will take care of discarded frames

Ethernet Technologies

- These Ethernet technologies have been standardized over years by IEEE 802.3 CSMA/CD (Ethernet) working group
- Ethernet comes in many different flavors, such as
  - 10BASE-T, 10BASE-T2, 100BASE-T, 1000BASE-LX, 10GBASE-T, 40GBASE-T, …
- First part refers to speed of standard: 10, 100, 1000, or 10G (bps)
- "**BASE**" refers to baseband Ethernet, meaning that physical media only carries Ethernet traffic
- Final part refers to physical **media** itself

Ethernet Technologies

- Ethernet is both a link-layer and a physical-layer and is carried over a variety of physical media including coaxial cable, copper wire, and fiber

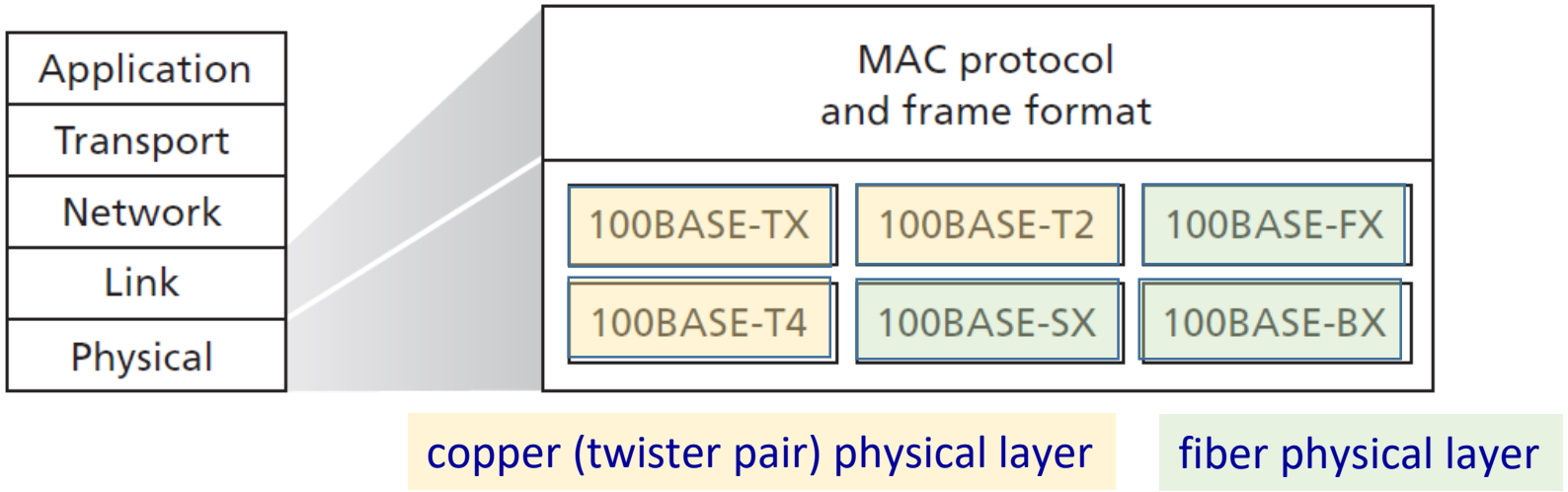


Figure 6.21 100 Mbps Ethernet standards: A common link layer, different physical layers

Gigabit Ethernet

- Offering a raw data rate of **40,000 Mbps**, **40 Gigabit**. Standard for Gigabit Ethernet, referred to as **IEEE 802.3z**, does the following:
- Uses standard Ethernet frame format (Figure 6.20) and is backward compatible with 10BASE-T and **100BASE-T** technologies. This allows for easy integration of Gigabit Ethernet with the existing installed base of Ethernet equipment.
- Allows for point-to-point links as well as shared broadcast channels. Point-to-point links use **switches** while broadcast channels use hubs (called *buffered distributors)*
- Uses CSMA/CD for shared broadcast channels. In order to have acceptable efficiency, maximum distance between nodes must be severely restricted
- **Allows for full-duplex operation at 40Gbps in both directions for point-to-point channels**
- Initially operating over **optical fiber**, Gigabit Ethernet is now able to run over **category 5 UTP cabling** (Unshielded Twisted Pair)

Today's Ethernet

- Ethernet includes **CSMA/CD** MAC protocol, to deal with **collisions** in a wired broadcast LAN spanning a small geographical region
- Ethernet today is used in a **switch-based star topology**, using store-and-forward packet switching
- Modern switches are **full-duplex**, so that a switch and a node can each send frames to each other at same time without interference or collision
- In other words, in a switch-based Ethernet LAN there are no collisions and, therefore, there is no need for CSMA/CD MAC protocol
- Today's Ethernets are very different from original Ethernet, but Ethernet's frame format has remained unchanged

6.4.3 Link-Layer Switches

- Role of a switch is to receive incoming link-layer frames and forward them onto outgoing links
- Switch is **transparent** to hosts and routers in subnet
- A host/router addresses a frame to another host/router (rather than addressing frame to switch) and sends frame into LAN, unaware that a switch will be receiving frame and forwarding it
- Rate at which frames arrive to any one of **switch's output interfaces** may temporarily **exceed link capacity** of that interface, so, switch output interfaces have **buffers**, in much same way that router output interfaces have buffers for datagrams

Forwarding and Filtering

- Filtering: A frame should be forwarded or should just be dropped
- Forwarding: To determine interfaces to which a frame should be directed, and then moves frame to those interfaces
- **Filtering** and **forwarding** are done using a **switch table**
- **Switch table** contains entries for some, but not necessarily all of hosts and routers on a LAN
- An entry in switch table contains

1. a MAC address
2. switch interface that leads toward that MAC address
3. time at which entry was placed in table

Switch Table Example

- Switch table for switch in Figure 6.15 is shown in Figure 6.22

Address	Interface	Time
88-B2-2F-54-1A-0F	1	9:32
1A-23-F9-CD-06-9B	2	9:36
5C-66-AB-90-75-B1	3	9:37
49-BD-D2-C7-56-2A	4	9:40

Figure 6.22 Switch table for a non-SDN switch
Interface that leads toward a MAC address

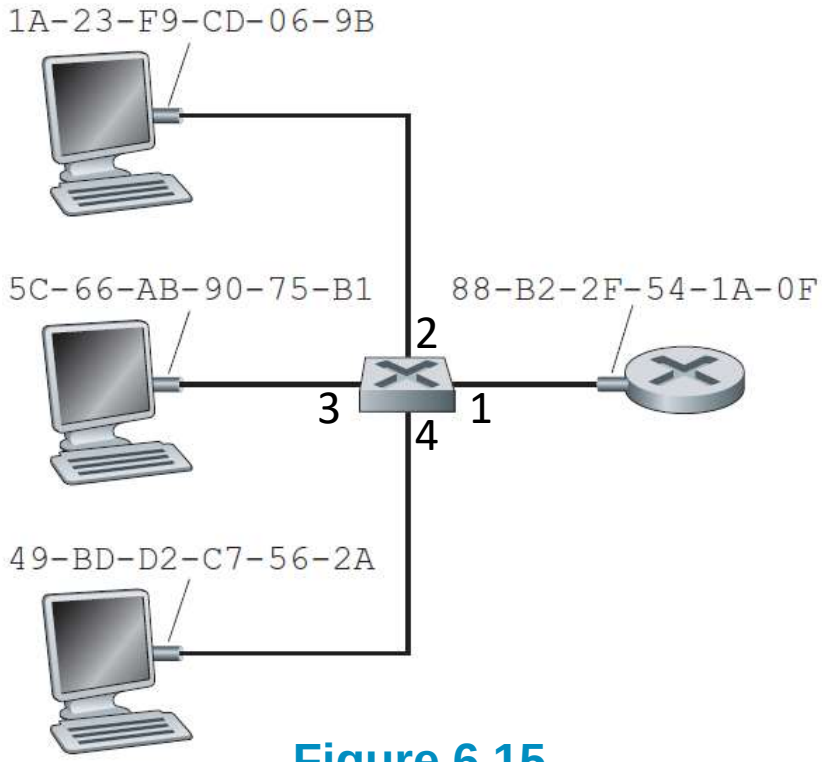
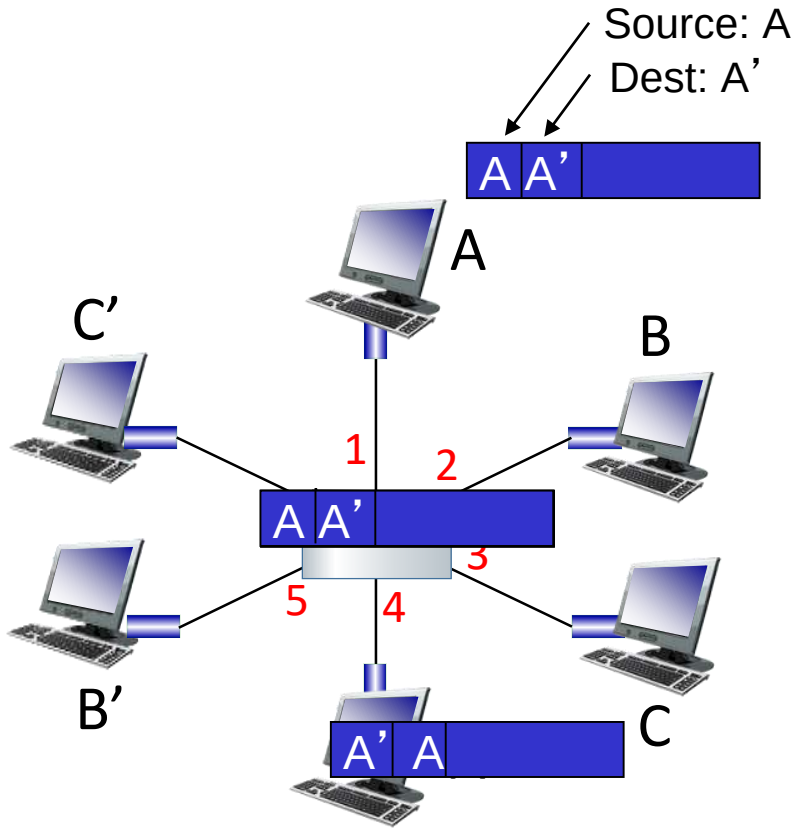


Figure 6.15

Self-Learning

- Switch learns which hosts can be reached through which interfaces
- When frame received, switch "learns" location of sender: incoming LAN segment
- Records sender/location pair in switch table
- Frame destination is A', location unknown: flood



Address	Interface	Time
A	1	13:40
A'	4	13:41

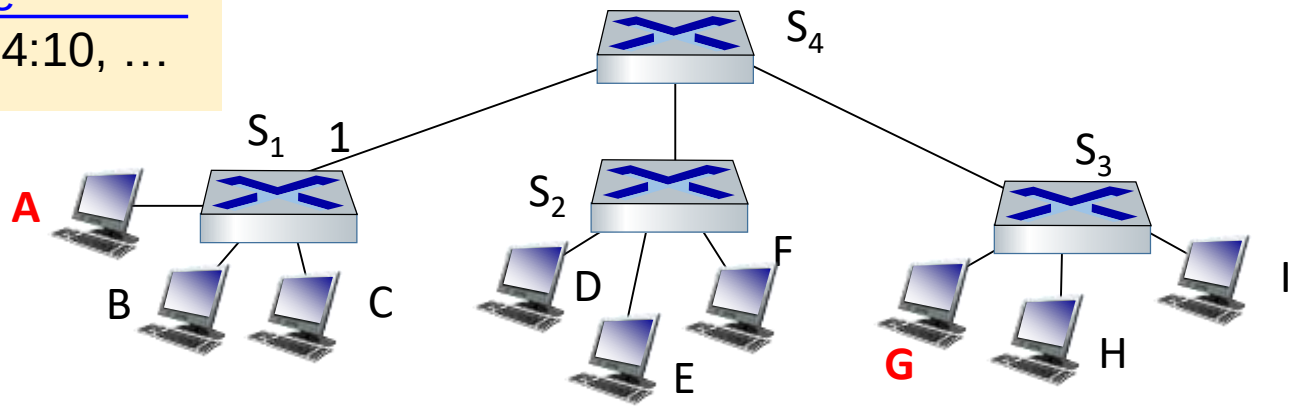
switch table (initially empty)

Interconnecting switches

- Self-learning switches can be connected together:
- Q: Sending from **A** to **G**, how does S_1 know to forward frame destined to **G** via S_4 and S_3?
- A: Self learning (works exactly same as in single-switch case)

Address	Interface	Time
D, E, F, G, H, I	1	13:30, 14:10, …

Part of switch table of S_1



Properties of Link-Layer Switching

Advantages of using switches, rather than broadcast links:

- Elimination of collisions: Switch buffers frames and never transmit more than one frame on a interface at any one time. Switches provide a significant performance improvement over LANs with broadcast links
- Heterogeneous links: Because a switch isolates one link from another:
  - Different links in LAN can operate at different speeds and can run over different media
  - For example, uppermost switch in Figure 6.15 might have three 1Gbps 1000BASE-T copper links, two 100Mbps 100BASE-FX fiber links, and one 100BASE-T copper link

Management in Switched LAN

- Management: A switch eases network management
- For example, if an adapter malfunctions and continually sends Ethernet frames (called a **jabbering adapter**), a switch can detect problem and internally disconnect malfunctioning adapter

- Switches gather **statistics** on
  - **Bandwidth usage**
  - **Traffic types**

  (make this information available to **network manager)**

Switches Versus Routers

Both are store-and-forward:

- Routers: forwards packets using IP addresses
- Switches: forwards packets using MAC addresses
- SDN switches uses MAC, IP, …, to forward a packet

Both have forwarding tables:

- Routers: compute tables using routing algorithms for **end-to-end** optimal path
- Switches: learn forwarding table using sender MAC addresses, flooding

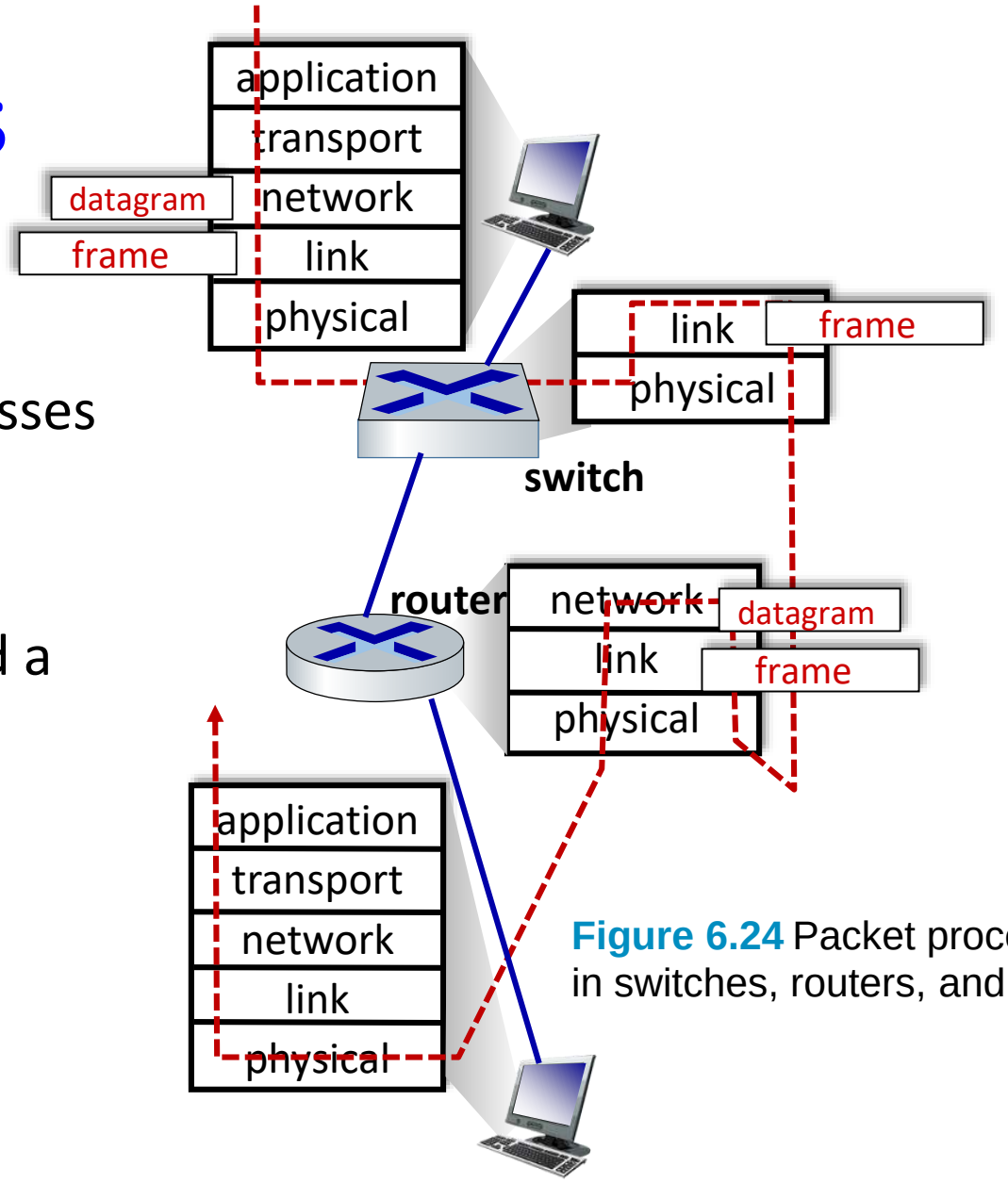


Figure 6.24 Packet processing in switches, routers, and hosts

Pros and Cons of Switches

Pros:

- Switches are plug-and-play
- Switches can also have relatively **high** filtering and forwarding **rates** (process frames only up through layer 2)

Cons:

- To prevent cycling of broadcast frames, in a switched network, topology should be **restricted to a spanning tree**
- **Large switched network** would require **large ARP tables** in hosts and routers and would generate substantial ARP traffic and processing
- Broadcast storms: If one host transmits **endless stream broadcast frames**, switches will forward all of these frames, causing entire network to collapse

Pros and Cons of Routers

Pros:

- Because network addressing is hierarchical, packets do not normally cycle through routers even when network has redundant paths
- Packets are not restricted to a spanning tree and can use best path between source and destination, so, they have allowed Internet to be built with a **rich topology** that includes, **multiple active paths between any two locations**
- Routers **provide firewall** protection **against layer-2 broadcast storms**

Cons:

- Routers are **not plug-and-play**, and need IP addresses to be configured
- Routers have a **larger per-packet processing time** than switches (process up through layer-3)

Switches or Switches + Routers

- **Small networks** consisting of a few hundred hosts:
  - **Switches suffice for these small networks**, as they localize traffic and increase aggregate throughput without requiring any configuration of IP addresses
  - Are consist of several LANs

- **Larger networks** consisting of thousands of hosts:
  - Typically include routers within network (in addition to switches)
  - Routers provide a more robust isolation of traffic, control broadcast storms, and use more "intelligent" routes among hosts in the network

6.4.4 Virtual Local Area Networks (VLANs)

- Figure 6.15: 3 switched LANs connected via a switch hierarchy
- Three drawbacks can be identified
  - Lack of traffic isolation
  - Inefficient use of switches
  - Difficult managing of users

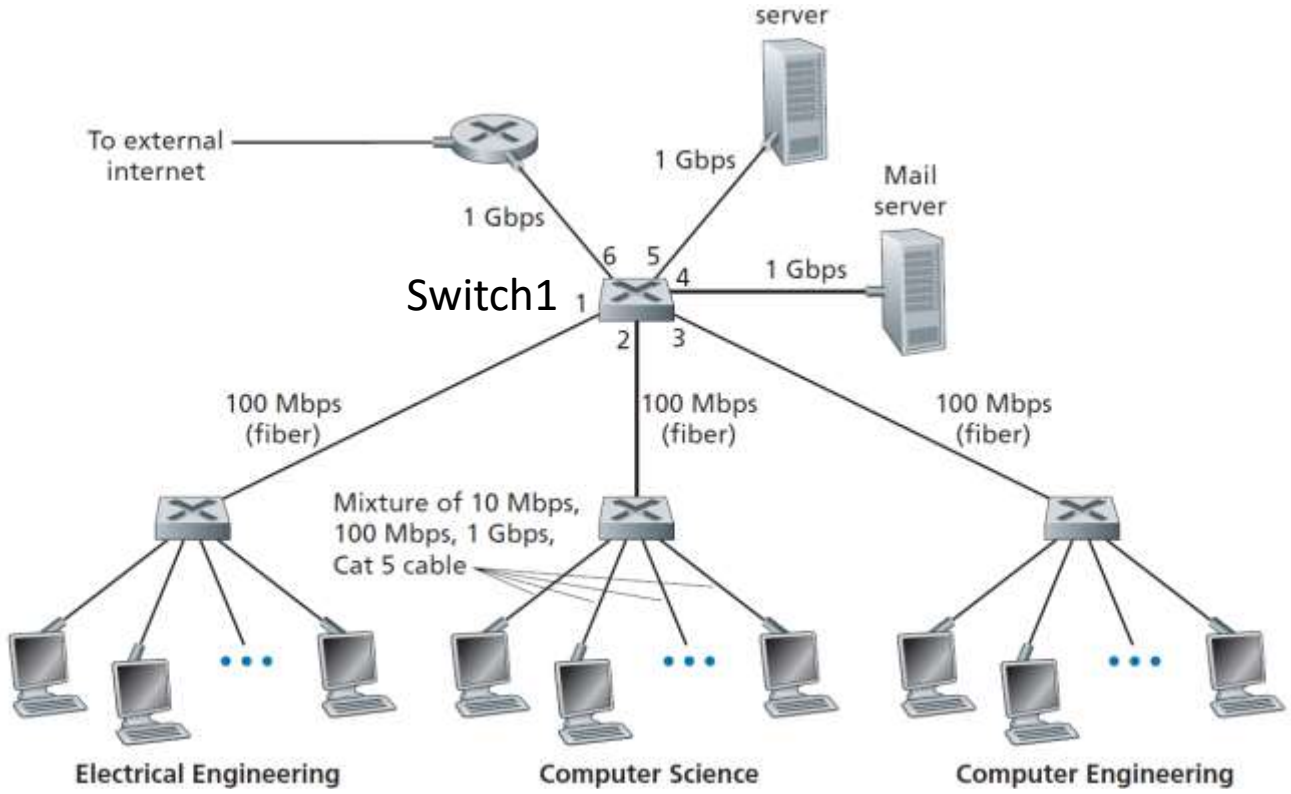


Figure 6.15 An institutional network connected together by four switches.

Three drawbacks

1- Lack of traffic isolation:

- **Broadcast traffic** (e.g., ARP and DHCP frames or frames whose destination has not yet been learned by a self-learning switch) must still **traverse entire institutional network**
- Limiting scope of such broadcast traffic would improve LAN performance
- It also is desirable to limit LAN broadcast traffic for security/privacy reasons
  - If one group contains **company's executive management** team and another group contains **disgruntled employees** running Wireshark packet sniffers, network manager may well prefer that executives' traffic never even reaches employee hosts
- This type of isolation could be provided by **replacing Switch1 in Figure 6.15 with a router**. This isolation also can be achieved via a virtual switch

Three drawbacks

2- Inefficient use of switches:

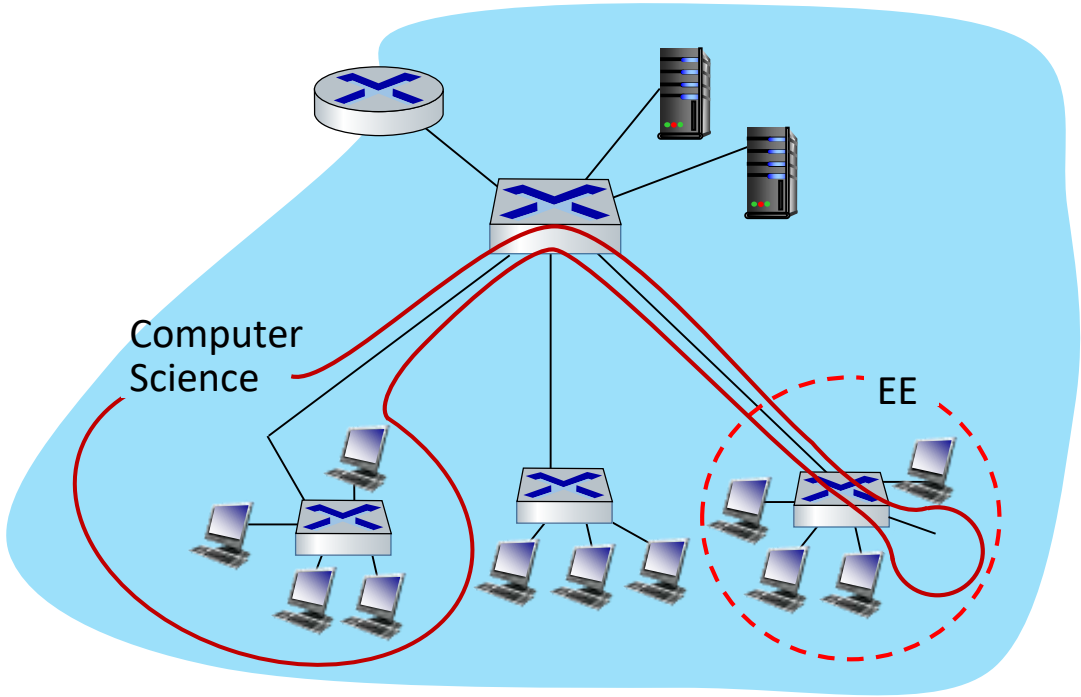
- If instead of three groups, institution had **10 groups**, then 10 first-level switches would be required
- If each group were small, say less than 10 people, then **a single 96-port** switch would likely be large enough to accommodate everyone, but this single switch would not provide traffic isolation. **Can we use a 96-port switch and have 10 separate groups?**

3- Managing users:

- If an employee moves between groups, physical cabling must be changed to connect employee to his/her own original switch. Employees belonging to two groups make problem even harder

Managing users

- CS user moves office to EE and physically connects LAN cable to EE switch (not original switch), but wants to remain logically attached to CS switch



Virtual Local Area Networks (VLANs)

- A switch that supports **VLANs** allows multiple virtual local area networks to be defined over a single physical local area network infrastructure
- Hosts within a VLAN communicate with each other as if they were connected to a switch

Port-Based VLAN switch

- In a **port-based VLAN**, switch's ports (interfaces) are divided into groups by network manager
- Each group constitutes a VLAN, with ports in **each VLAN forming a broadcast domain** (broadcast traffic from one port can only reach other ports in group)

VLAN Switch

- **VLAN switch solves all three drawbacks**
- CS and EE switches in Figure 6.15 can been replaced by a single VLAN switch
- If EE user at port 8 connects to port 11 of CS, network manager reconfigures VLAN software so that port 11 is now associated with EE VLAN (ports 1 and 16 are unassigned)
- Switch management software: a table of port-to-VLAN mappings is maintained within switch
- Switch supporting VLAN can be configured to define multiple virtual LANS over single physical LAN infrastructure

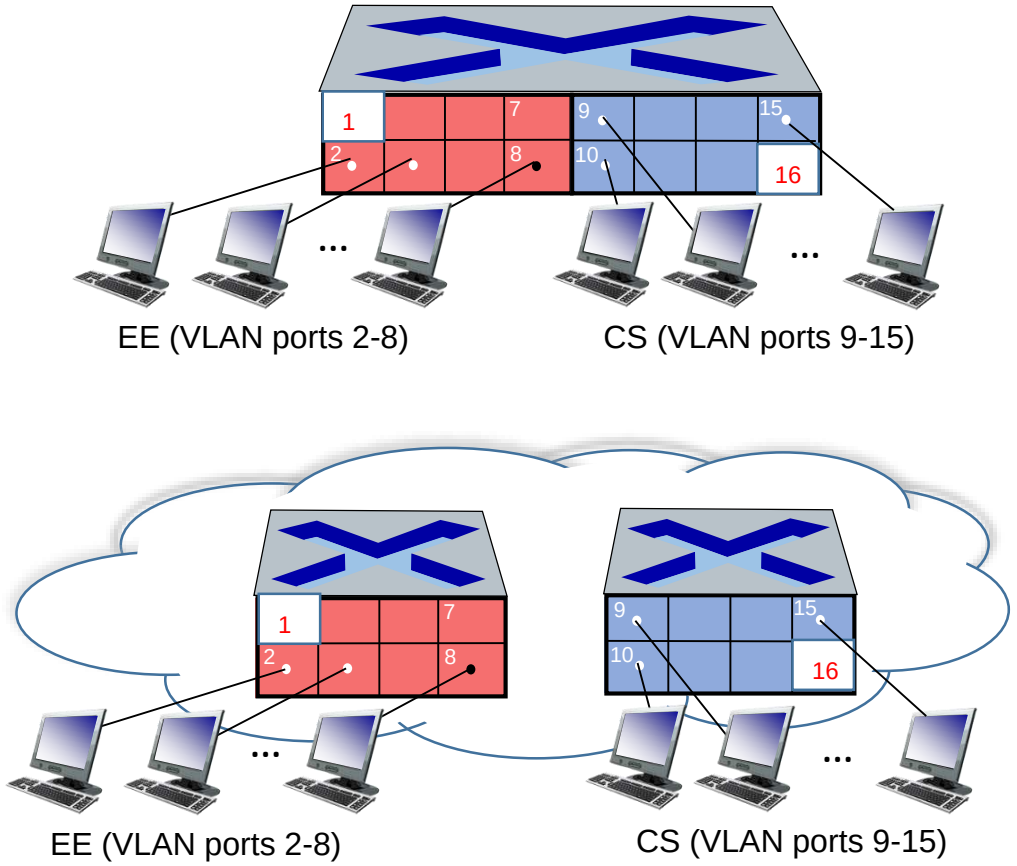
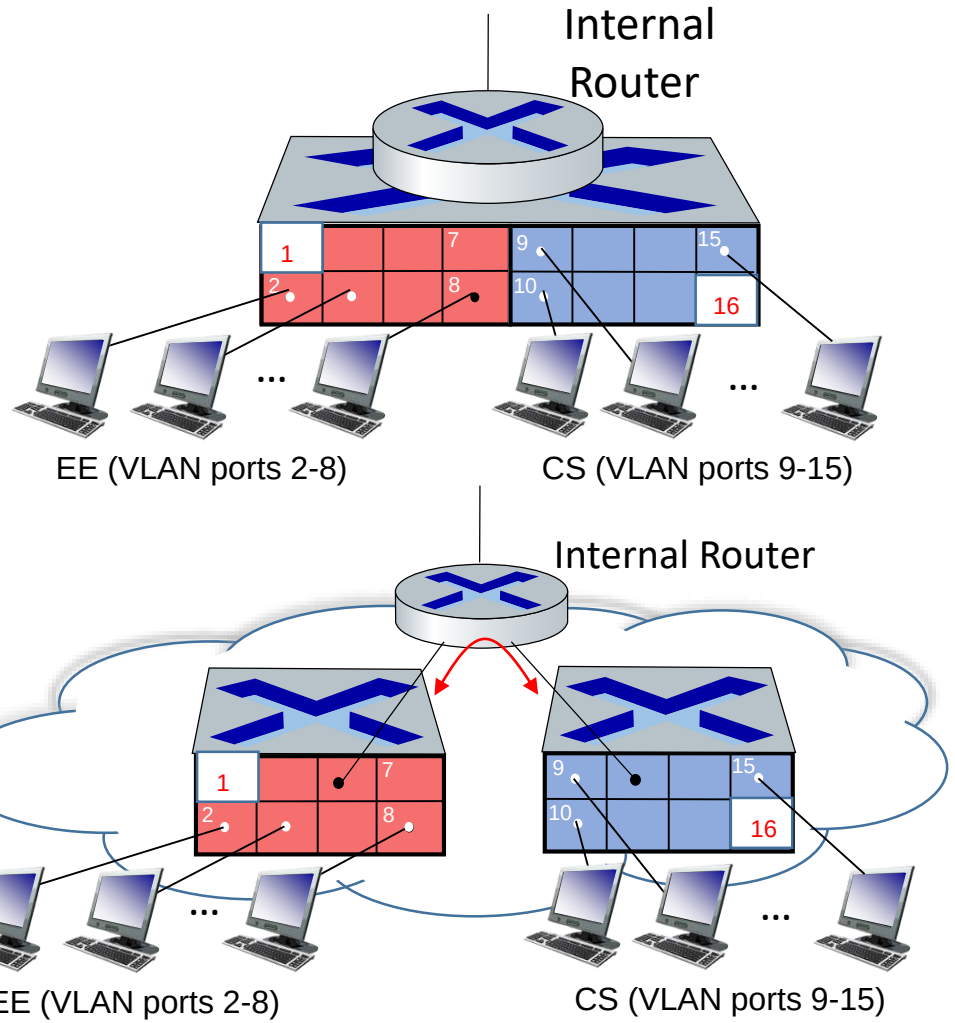


Figure 6.25 A single switch with two isolated configured VLANs

Traffic between VLANs in a VLAN switch

- How can traffic **from EE** Department be sent **to CS** Department?
- Switch vendors make a single device that contains a VLAN switch and a router
- An IP datagram going from EE to CS first cross EE VLAN to reach router and then be forwarded by router back over CS VLAN to CS host
- Internal router does Inter-VLAN routing



Trunking

- Suppose EE and CS faculty are housed in a separate building
- Figure 6.26 shows a two VLAN switches, one in each EE or CS building
- VLAN Trunking: **one or more ports** on each switch (e.g., **port 1** and **16** on left switch and port 1 and 8 on right switch) is/are **configured as trunk port(s)** to interconnect VLAN switches
- Ports with **highest bandwidth** are usually configured as Trunk port
- In a switch, trunk port(s) is/are belong to all VLANs in that switch
- Frames sent to any VLAN are forwarded over trunk link to other switch

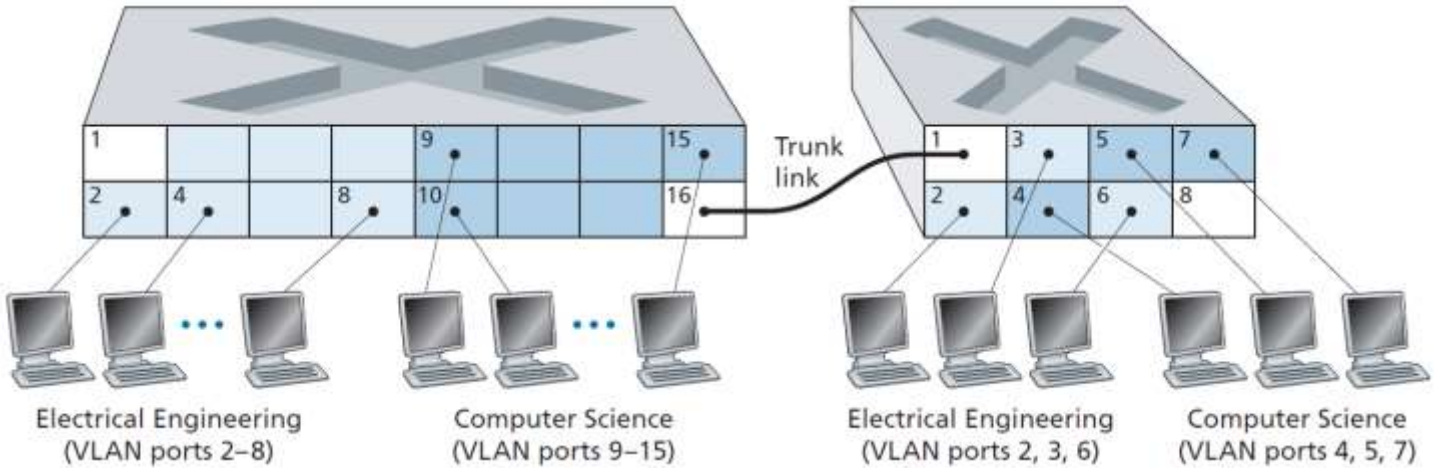


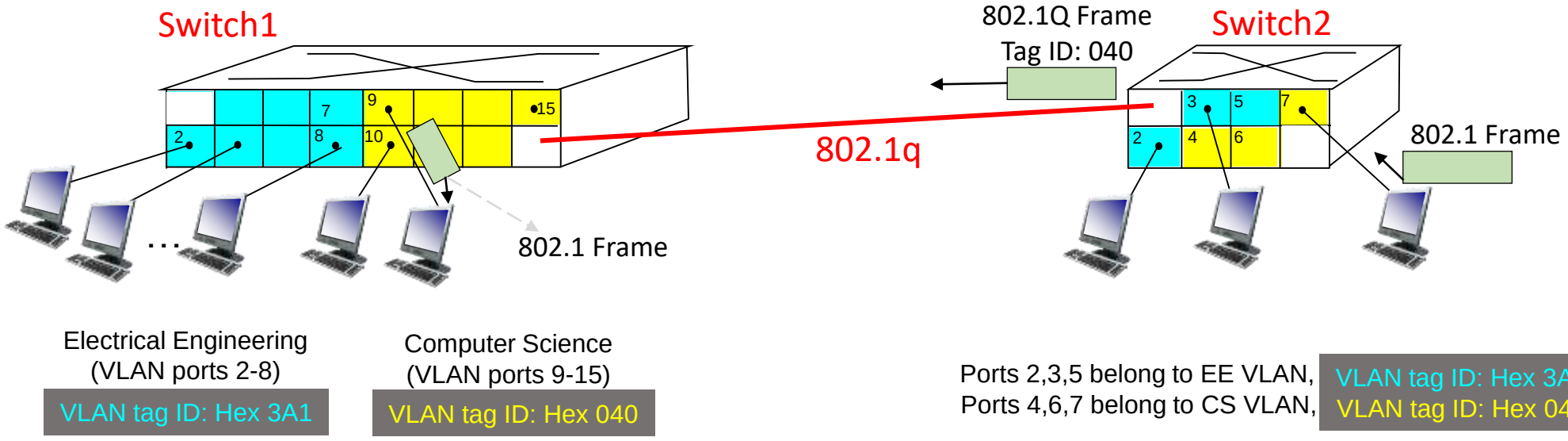
Figure 6.26 Connecting two VLAN switches with two VLANs: trunked

VLAN Identifier

- How does a switch know that a frame arriving on a trunk port belongs to a particular VLAN?
- IEEE has defined an extended Ethernet frame format, **802.1Q**, for frames crossing a VLAN trunk
- Frames between trunk ports carry destination **VLAN tag**
- VLAN tag consists of a 12bits destination VLAN Identifier
- 802.1Q protocol **adds/removed VLAN tag** for frames forwarded between trunk ports

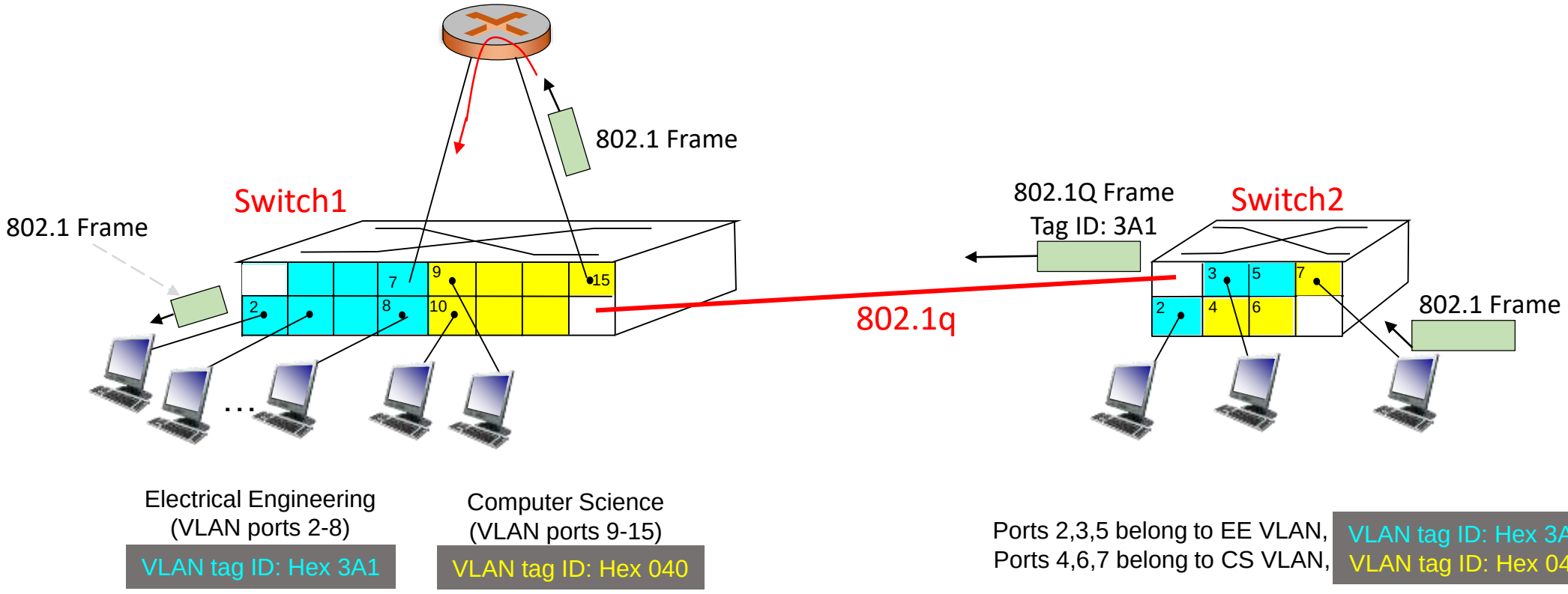
Example: Same VLANs

- A frame from a host (in VLAN:040 on Switch2) destined to host in VLAN:040 on Switch1



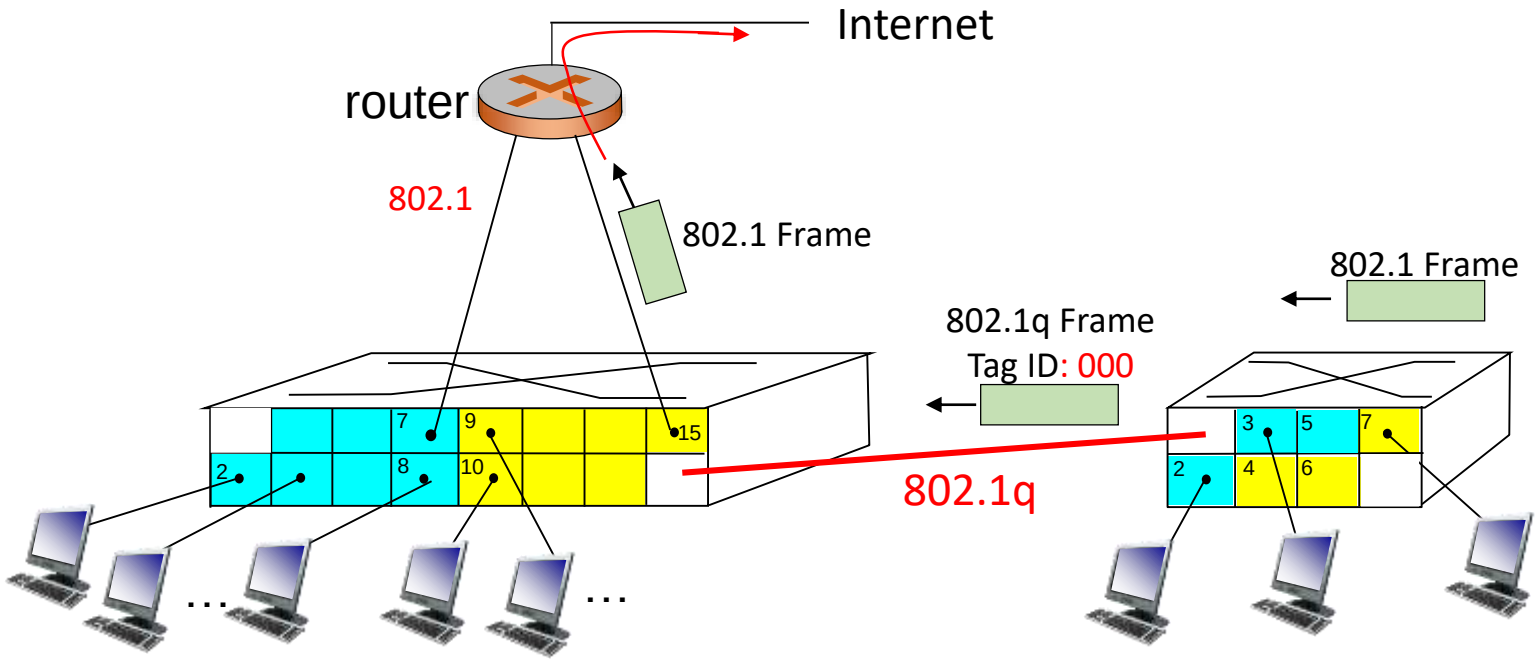
Example: Different VLANs

- A frame from a host (in VLAN:040 on Switch2) destined to host in VLAN:3A1 on Switch1



Example: To outside VLANs

- Tag ID value of 000 and FFF are reserved
- 000 indicates frame does not carry a VLAN ID, destination is not a VLAN
- FFF can be used in management operations



802.1Q frame

Min frame size unchanged at 64 byte
Max frame size extended from 1,518 bytes to 1,522 bytes

- 802.1Q frame consists of standard Ethernet frame with a four-byte **VLAN tag** added into header
- VLAN tag:
  - 2-byte Tag Protocol Identifier (TPID) field (set to a value of 81-00)
  - 2-byte Tag Control Information field contains of
    - 12-bit VLAN identifier field
    - 3-bit frame priority level field, similar to IP datagram TOS field
    - 1-bit CFI (Canonical Format Indicator)
- VLAN tag is added into a frame by switch at sending side of a VLAN trunk, parsed, and removed by switch at receiving side of trunk

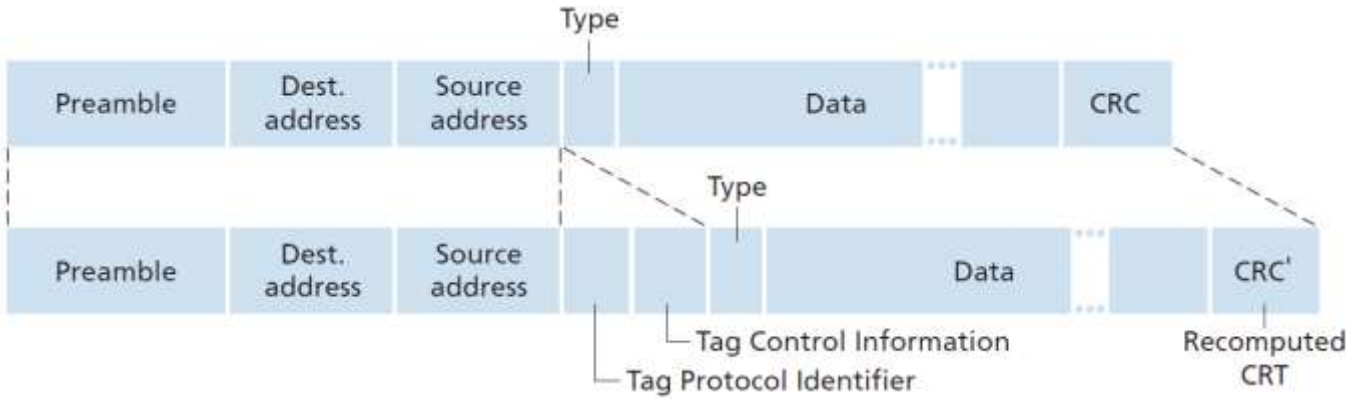


Figure 6.27 Original Ethernet frame (top), 802.1Q-tagged Ethernet VLAN frame (below)

Other VLAN concepts

- In this discussion was a brief introduction of VLANs **port-based VLAN**
- VLANs can be defined in several other ways:
  - MAC-based VLANs: network manager specifies set of MAC addresses that belong to each VLAN; whenever a device attaches to a port, port is connected into appropriate VLAN based on MAC address of device
  - Network layer protocol-based: IPv4, IPv6, or Appletalk, …
  - …
- It is also possible for VLANs to be extended across IP routers, allowing islands of LANs to be connected together to form a single VLAN that could span globe

Contents

6.5 Link Virtualization: A Network as a Link Layer

- Multiprotocol Label Switching (MPLS) is a **packet-switched**, **virtual-circuit** network
- It has its own packet formats and forwarding behaviors
- **Discussion of MPLS fits well into a study of either network layer or link layer**
- From an Internet viewpoint, we can consider an MPLS network as a link-layer technology that serves to interconnect IP routers and devices
- Frame-relay and ATM networks can also be used to interconnect IP devices, though they represent slightly older (but still deployed) technologies

6.5.1 Multiprotocol Label Switching (MPLS)

- MPLS uses a **fixed-length label** in datagram header
- MPLS capable IP routers forward datagrams **either based on labels or destination IP addresses**
- IETF unified these efforts in MPLS protocol [RFC 3031, RFC 3032], effectively **blending VC techniques into a routed datagram network**

MPLS frame format

- A link-layer frame transmitted between MPLS-capable devices has a small MPLS header added between layer-2 header and layer-3 header (RFC 3032)
- An standard IP router would be quite confused when it found an MPLS header where it had expected to find IP header

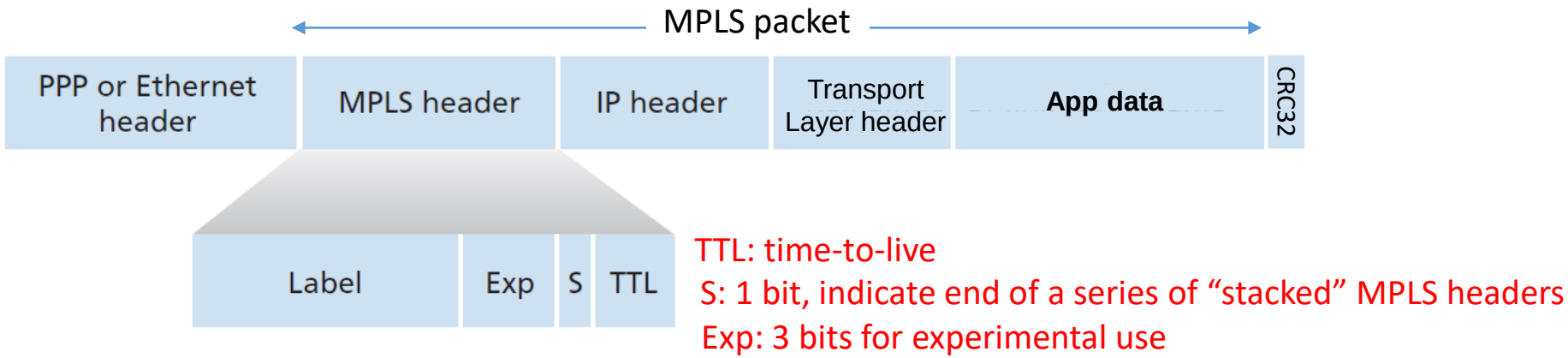


Figure 6.28 MPLS header: Located between link- and network-layer headers

Ethernet TYPE: 8847_{HEX} for MPLS unicast, and 8848_{HEX} for MPLS multicast

Label switching

When an MPLS frame inters into a MPLS-capable router:

- MPLS-capable router forwards an MPLS frame by looking up MPLS label in its forwarding table and then passing datagram to appropriate output interface Label of frame will be switched to a new one before outputting (**Labels witching**)
- MPLS-capable router need not extract destination IP address and perform a lookup of longest prefix match in forwarding table

But:

- How does a router know if its neighbor is indeed MPLS capable, and how does a router know what label to associate with given IP destination?
- To answer these questions look at next Example

Example

- R1 → R2, R3: I deliver label=6 to A
- R2 → R4: I deliver label=8 to A
- R3 → R4: I deliver label=10 to A
- R4: two paths to A (interface0 with outbound label=10 and interface1 with outbound label=8)
- Like a switched LAN, MPLS capable routers R1 through R4 connect together IP devices without touching IP header of a datagram

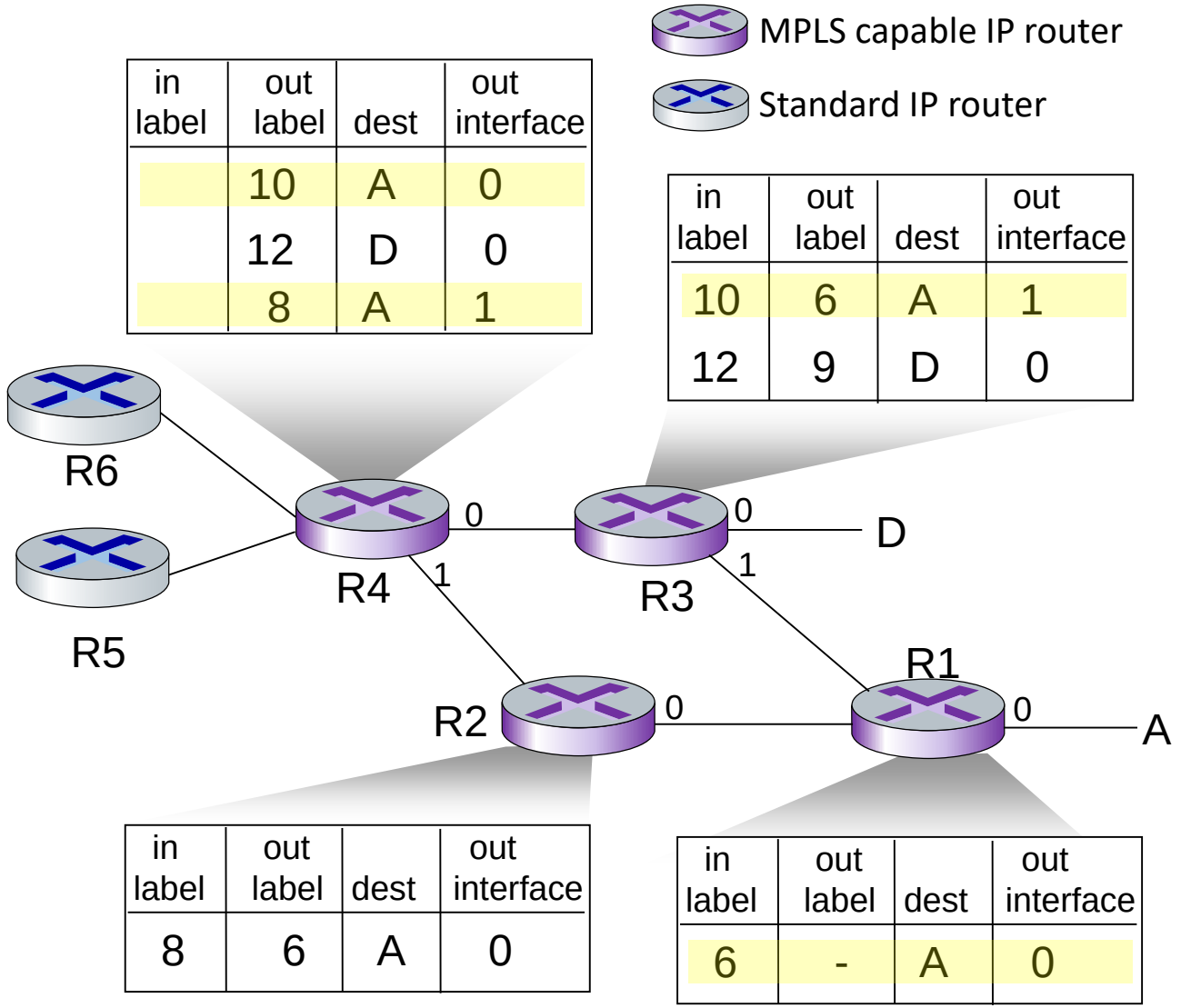
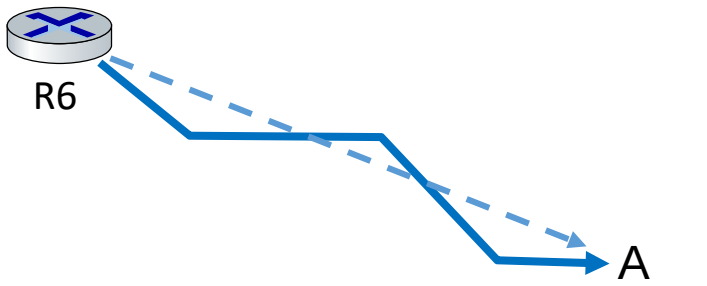


Figure 6.29 MPLS-enhanced forwarding
IP devices R5, R6, A, and D are connected together via an MPLS infrastructure

Example

- A packet from R6 to A
- Packet's label switched from 10 to 6 in interface1 of R3



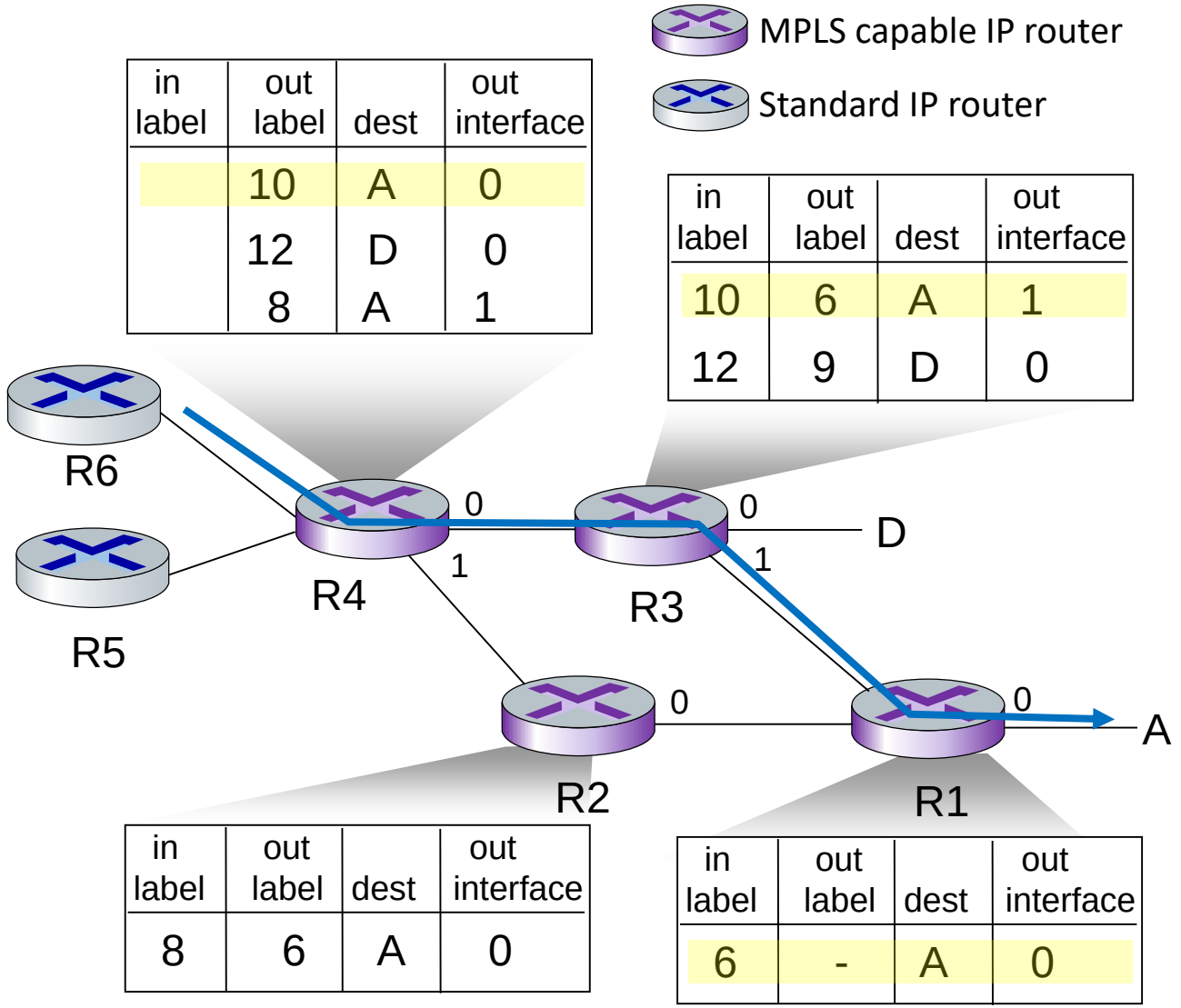
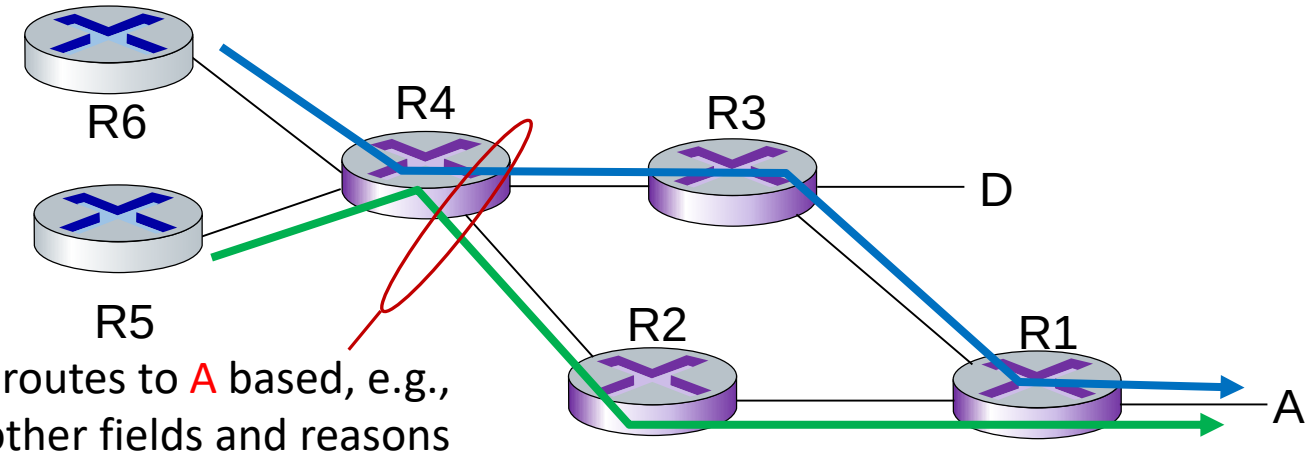


Figure 6.29 MPLS-enhanced forwarding
IP devices R5, R6, A, and D are connected together via an MPLS infrastructure

Advantages of MPLS

- Increases in switching speeds
- Traffic management (engineering) capabilities that MPLS enables
- A network operator can override normal IP routing and force some of traffic headed toward a given destination along one path, and other traffic destined toward same destination along another path (whether for policy, performance, or some other reason)



R4 can use different MPLS routes to A based, e.g., on IP source address or other fields and reasons

MPLS may be used for

- To perform fast restoration of MPLS forwarding paths, e.g., to reroute traffic over a precomputed failover path in response to link failure
- To implement VPN (Virtual private networks)
  - An ISP uses its MPLS-enabled network to connect together customer's various networks

MPLS and SDN

- MPLS rose before development of SDN
- Many of MPLS' traffic engineering capabilities can also be achieved via SDN
- Only future will tell whether MPLS and SDN will continue to co-exist, or whether newer technologies (such as SDN) will eventually replace MPLS

We have not discussed

- How MPLS computes paths for packets among MPLS capable routers (path computation algorithms are not standardized, and are currently vendor-specific)
- How MPLS gathers link-state information to use in these path computations
- What specific protocol used to distribute labels among MPLS-capable routers

Contents

Data Center (DC)

Google data center in Denmark

6.6 Data Center Networking

- Internet companies such as **Google**, **Microsoft**, **Amazon**, and **Alibaba** have built massive data centers, each housing **tens to hundreds of thousands of hosts**
- Data centers are connected to Internet
- Data centers serve three purposes:
  1. **Provide content** such as Web pages, search results, e-mail, or streaming video to users
  2. Serve as **massively-parallel computing infrastructures** for specific data processing tasks, such as distributed index computations for search engines
  3. Provide **cloud computing** to other companies
     - **A major trend for companies** is to use a cloud provider such as Amazon Web Services, Microsoft Azure, and Alibaba Cloud to handle essentially all of their IT needs
- Data Centers have **internally a complex computer networks**, called data center networks, which interconnect their **internal hosts**

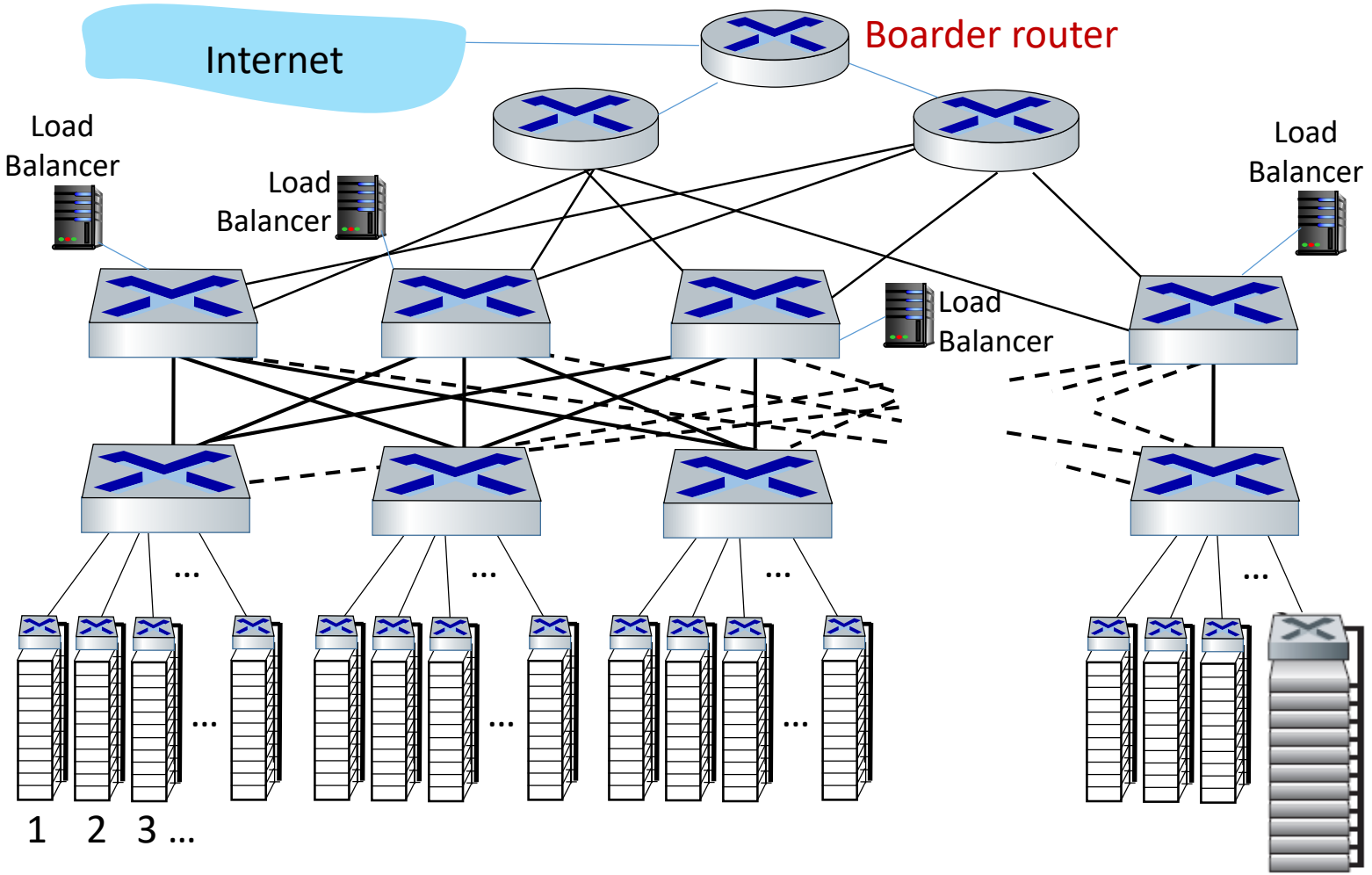
6.6.1 Data Center Architectures

- Data center designs are carefully kept **company secrets**, as they often provide critical competitive advantages to leading cloud computing companies
- Cost of a large data center is huge, exceeding **$12 million per month for a 100,000 host** data center in 2009
  - 45% can be attributed to hosts themselves (which need to be replaced every 3–4 years)
  - 25% to infrastructure, including **transformers**, uninterruptable power supplies (**UPS**) systems, **generators** for long-term outages, and **cooling** systems
  - 15% for **electric utility costs**
  - 15% for **networking**, including network (**switches, routers, and load balancers**), **external links**, and **transit traffic costs**
- While networking is not largest cost, networking innovation is key to reducing overall cost and maximizing performance

Hierarchical Architecture

- Network of a DC with few thousand hosts in few tens of racks:
  - A border router
  - A load balancer
  - **A single Ethernet switch to interconnect all TOR switches**
- Network of a DC with tens to hundreds of thousands of hosts:
  - A hierarchy of routers and switches (Figure 6.30)
  - Border router connects to access routers
  - Below each access router, there are three tiers of switches
  - Each access router connects to a top-tier switch, and each top-tier switch connects to multiple second-tier switches and a load balancer
  - Each second-tier switch in turn connects to multiple racks via the racks' TOR switches (third-tier switches)
  - All links typically use Ethernet for their link-layer and physical-layer protocols, with a mix of copper and fiber cabling
- With such a hierarchical design, it is possible to scale a data center to hundreds of thousands of hosts

Figure 6.30 A data center network with a hierarchical topology



Boarder router (to another AS)

Access routers
Connections outside datacenter

Tier-1 switches
Connecting to ~16 T-2s below

Tier-2 switches
Connecting to ~16 TORs below

TOR switch, Tier-3 switches
One per rack
40-100Gbps Ethernet to blades

Server racks
20- 40 server blades: hosts

Hosts in Data Centers

- Hosts in data centers, called **blades**, that include CPU, memory, and disk storage
- Hosts are stacked in racks, with **each rack typically having** 20 to 40 blades
- At top of each rack, there is a switch, named **Top of Rack (TOR) switch**, that interconnects hosts in rack with each other and with other switches in data center
- **Each host in rack has a network interface that connects to its TOR switch**, and each TOR switch has additional ports that can be connected to other switches
- Today, hosts typically have 40Gbps or 100Gbps Ethernet connections to their TOR switches
- Each host is also assigned its own data-center internal IP address

Data center Network

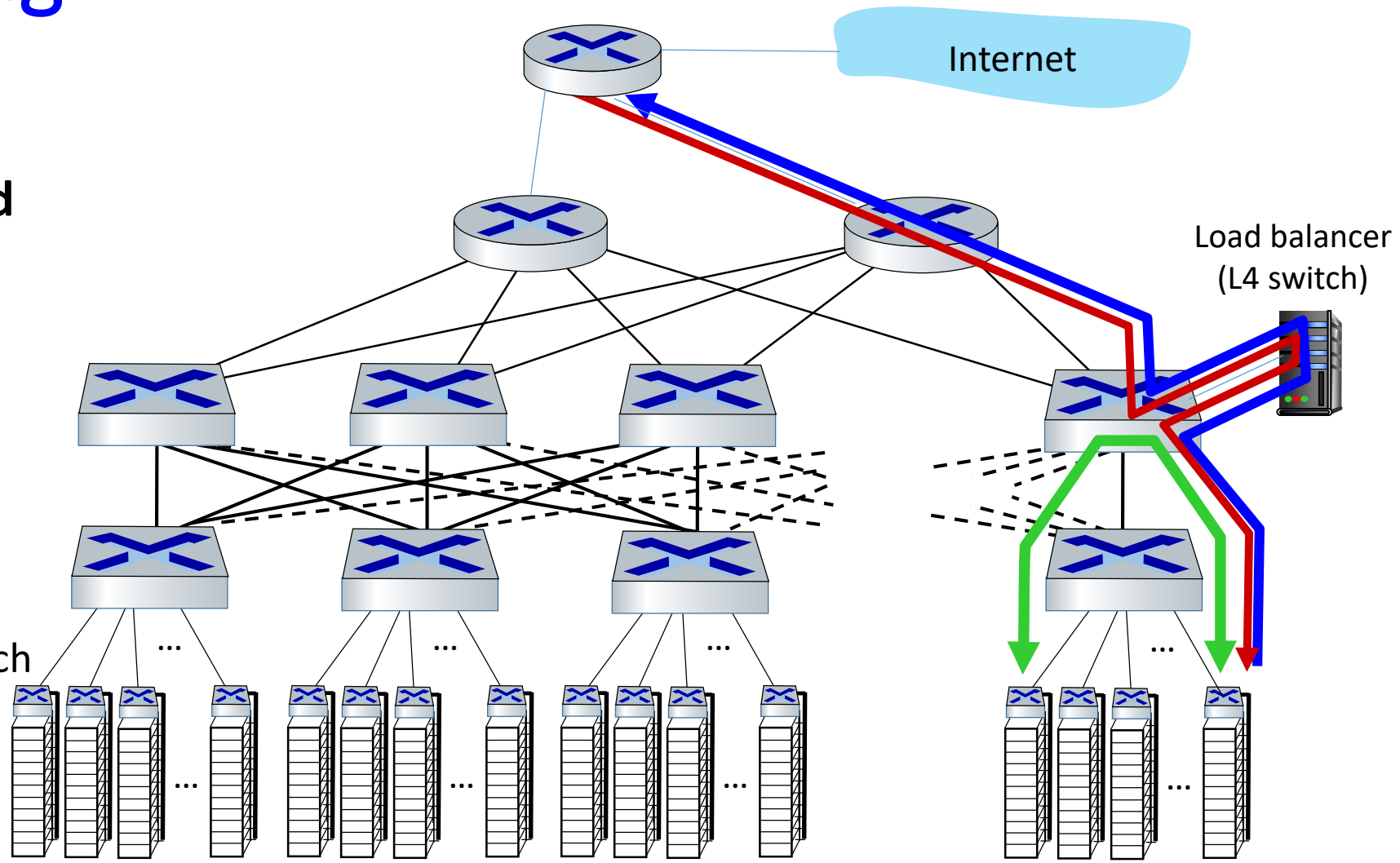
- DC network supports **two types of traffic**:
  1. Traffic flowing between external clients and internal hosts
  2. Traffic flowing between internal hosts
- To handle flows between external clients and internal hosts, DC network includes one or more **border routers**, **connecting DC network to public Internet**
- DC network interconnects racks with each other and connects racks to border routers
- DC network design: Art of designing interconnection network and protocols that connect racks with each other and with border routers, has become an **important branch of computer networking research in recent years**

Load Balancing

- **A cloud DC provides many applications concurrently**, such as search, e-mail, and video applications
- **Each APP is associated with a publicly visible IP address** to which clients send their requests and from which they receive responses
- Upon receiving a request for a particular application, **load balancer forwards it to one of hosts that handles application**
- Such a load balancer is sometimes referred to as a "layer-4 switch" since it makes decisions based on **destination port number** (layer 4) **as well as destination IP address**
- **Load balancer also provides a NAT-like function**, translating public external IP address to internal IP address, and then translating back for packets traveling in reverse direction back to clients
  - Clients can not contacting to hosts directly, and hiding internal network structure

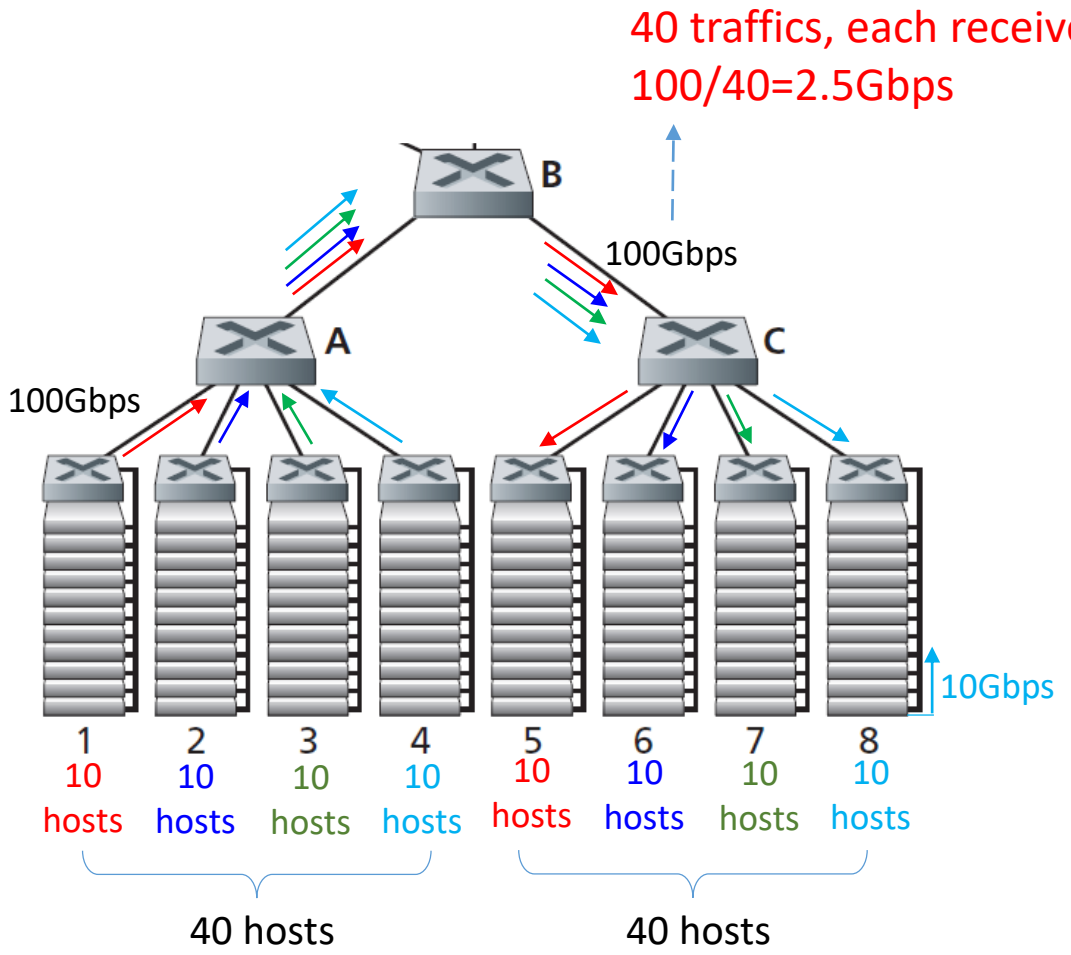
Load Balancing

- External requests are **first directed to a load balancer** whose job is to distribute requests to hosts, balancing load across hosts as a function of their current load
- A large DC will often have several load balancers, each one devoted to a set of specific cloud applications



Host-to-host capacity

- Suppose host to TOR is a **10Gbps link**, links between switches are **100Gbps**
- Two hosts in same rack can always communicate at a full 10Gbps
- Consider a traffic of **40 simultaneous** flows between **40 pairs of hosts** in **different racks**
- Each traffic only receive 2.5Gbps<<10Gbps
- Problem becomes even more acute for flows between hosts that **need to travel higher up hierarchy**

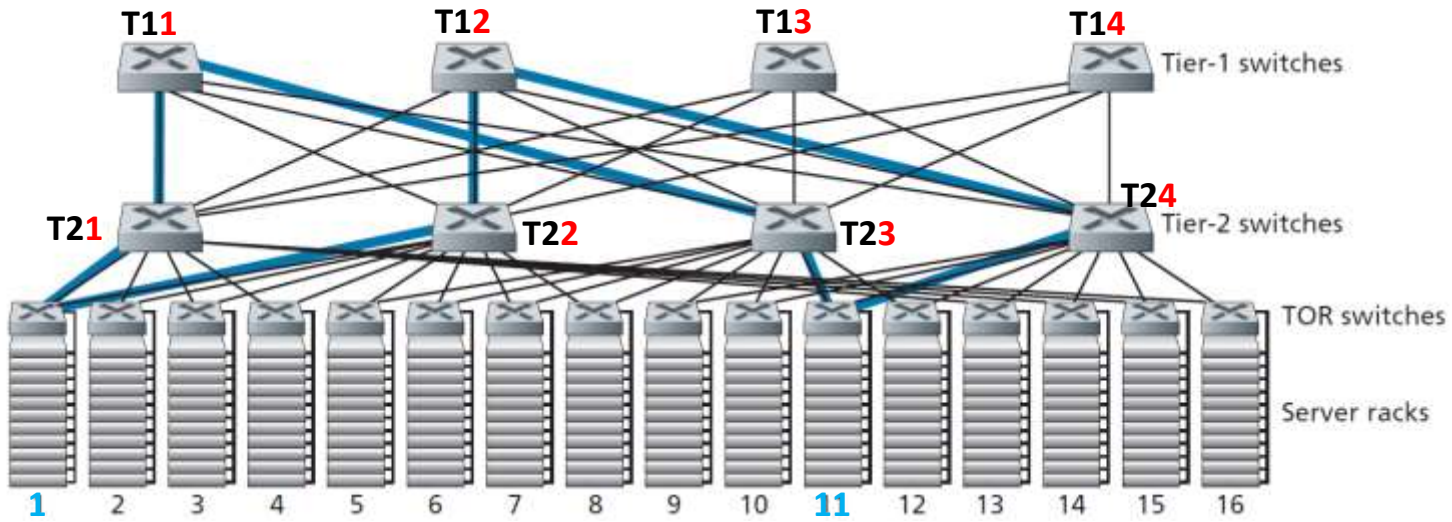


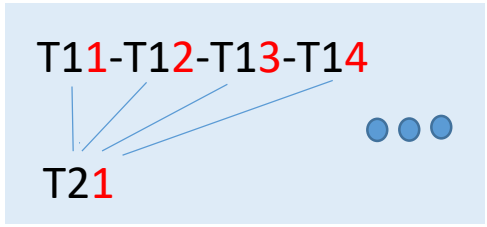
Possible solutions

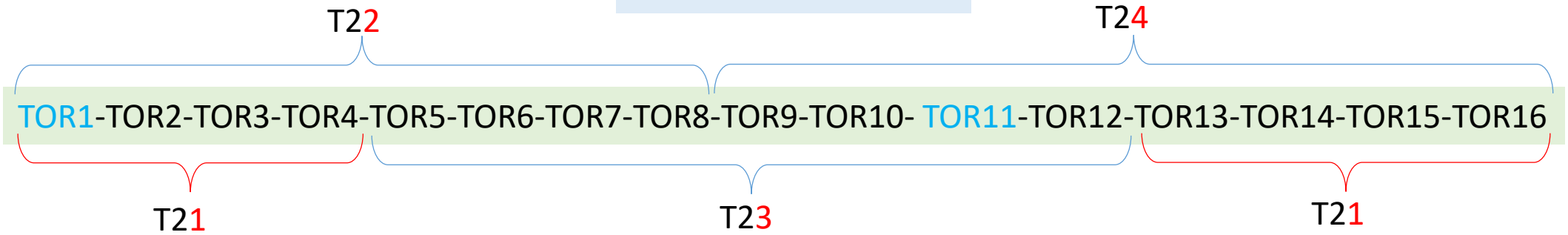
1. **Deploy higher-rate switches and routers,** very expensive solution!
2. **Co-locate** related services and data as close to one another as possible (e.g., in **same rack or in a nearby rack**) to **minimize inter-rack traffic**, reduces flexibility in placement of computation and services!
   - Large-scale Internet search engine may run on thousands of hosts spread across multiple racks with significant bandwidth requirements between all pairs of hosts
   - Cloud computing service: **multiple virtual machines** for a customer on placed on physical hosts with most capacity. If physical hosts are spread across multiple racks, network bottlenecks as described above may result in poor performance

Figure 6.31 Highly interconnected data network topology

- Each TOR switch has links to Tier2 switches
- TOR1 has links to Tier2-1 and Tier2-2 switches
- TOR11 has links to Tier2-3 and Tier2-4 switches



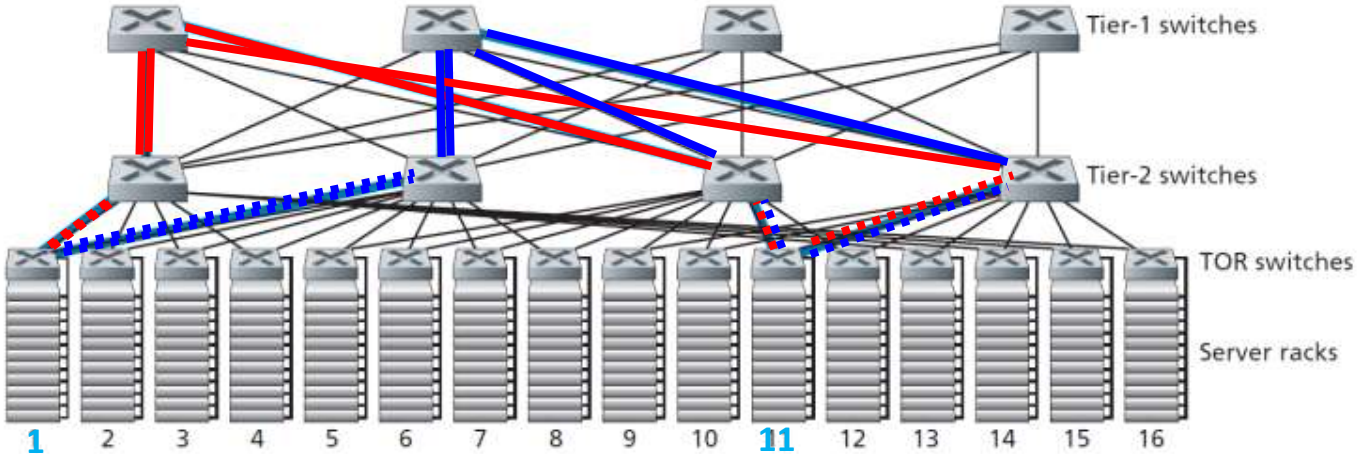




Possible solutions

3. Highly interconnected data network topology

- Figure 6.31: Two possible paths between TOR1 and TOR2, aggregate capacity if each path: 200Gbps
- Benefits:
  1. Increased reliability (**redundancy between Tier2 and Tier1**)
  2. Increased capacity
- In Facebook's DC, each TOR is connected to four different tier-2 switches, and each tier-2 switch is connected to four different tier-1 switches



Redundant network equipment

- Cloud application provider: continually providing applications with high availability using redundancy
  - Each TOR switch can connect to two **tier-2 switches**
  - Each **access router**, **tier-1 switch**, and **tier-2 switch** can be **duplicated**
  - Each access router form a single subnet. In order to localize ARP broadcast traffic, each of these **subnets** is further partitioned into **smaller VLAN subnets**

6.6.2 Trends in Data Center Networking

- DC networking is evolving rapidly, trends are driven by **cost reduction**, **virtualization**, **physical constraints**, **modularity**, and **customization**

Cost Reduction

- New DC network designs: To reduce the cost (also improve delay and throughput, ease of expansion and deployment)
  - Some designs are proprietary
  - Some designs are open (e.g., Facebook's Data Center Network)
- Design concept: hierarchical, tiered network interconnecting

Examples: GOOGLE DC, Facebook DC networks

Google's Jupiter DC:

- 48 links between TOR switch and its servers below, up to 8 tier-2 switches; a tier-2 switch has links to 256 TOR switches and links up to 16 tier-1 switches

Facebook DC:

- Each TOR switch connects up to four different tier-2 switches, and each tier-2 switch connects up to 4 of 48 tier-1 switches
- Tier-1 and tier-2 switches connect to a larger, scalable number of tier-2 or TOR switches, respectively
- Some of largest DC operators, built Tier-1 and tier-2 switches in-house
- A multi-switch layered (tiered, multistage) interconnection network are known as Clos networks, named after Charles Clos. A rich theory of Clos networks has been developed for **data center networking** and in **multiprocessor interconnection networks**

Centralized SDN Control and Management

- Because a DC is managed by a single organization, it is perhaps natural that a number of largest data center operators, including Google, Microsoft, and Facebook, are using SDN-like logically centralized controllers
- Their architectures also reflect a clear separation of a data plane (comprised of relatively simple, commodity switches) and a software-based control plane, as we saw in Section 5.5
- Due to immense-scale of their data centers, automated configuration and operational state management, as we encountered in Section 5.7, are also crucial

Virtualization

- Virtual Machines (VMs) **decouple software running applications** from **physical hardware**
- It allows **seamless migration** of VMs **between physical servers**, which might be located on different racks
- **Standard Ethernet and IP protocols have limitations** in enabling movement of VMs while maintaining active network connections across servers
- A solution is to treat entire data center network as a single, flat, layer-2 network
- To emulate effect of having all hosts connect to a "single" switch, ARP mechanism is modified to use a DNS style query system instead of a broadcast, and directory maintains a mapping of IP address assigned to a VM and which physical TOR switch VM is currently connected to in data center network
- Scalable schemes that implement this basic design have been proposed and have been successfully deployed in modern data centers

Physical Constraints

- DC networks operate in environments that not only have very high capacity (40Gbps and 100Gbps) but also have extremely low RTT delays (microseconds)
- Consequently, default TCP receive buffer sizes **are small** and **congestion control** protocols such as TCP and its variants **do not scale well** in data centers
- In DCs, congestion control protocols have to react fast and operate in extremely low loss regimes, as loss recovery and timeouts can lead to extreme inefficiency
- Several approaches have been proposed and deployed, ranging from DC-specific TCP variants to implementing Remote Direct Memory Access (RDMA) technologies on standard Ethernet
- Scheduling theory has also been applied to develop mechanisms that decouple flow scheduling from rate control, enabling very simple congestion control protocols while maintaining high utilization of the links

Hardware Modularity and Customization

- Shipping container–based modular DCs (MDCs)
- In an MDC, a factory builds, within a standard 12-meter shipping **container**, a "mini DC" and ships container to DC location
- Each container has up to a few thousand hosts, stacked in tens of racks, which are packed closely together
- At DC location, multiple **containers** are interconnected with each other and also with Internet
- Each prefabricated container is designed for performance degradation: as components (servers and switches) fail over time, container continues to operate but with degraded performance
- **When many components have failed and performance has dropped below a threshold, entire container is removed and replaced with a fresh one**
- Building a DC out of containers creates new networking challenges
- **Two types of networks**: container-internal networks within each of containers and core network connecting each container
  - **Within each container**, at scale of up to a **few thousand hosts**, it is possible to build a **fully connected network** using inexpensive commodity Gigabit Ethernet switches

Hardware Modularity and Customization

- However, design of core network, interconnecting hundreds to thousands of containers while providing high host-to-host bandwidth across containers for typical workloads, **remains a challenging problem**
- Another important trend is that large cloud providers are increasingly **building or customizing just about** everything that is in their DCs, including **network adapters**, **switches routers**, **TORs**, **software**, and **networking protocols**
- Another trend, pioneered by Amazon, is to improve reliability with "availability zones," which essentially replicate distinct DCs in different nearby buildings. By having buildings nearby (a few kilometers apart), transactional data can be synchronized across DCs in same availability zone while providing fault tolerance
- Many more innovations in data center design are likely to continue to come.

Contents

6.7 Retrospective: A Day in the Life of a Web Page Request

- Big picture view in a extremely simple case: downloading a Web page
- User connects a laptop to Ethernet switch, opens a Web Browser and types www.google.com into browser
- School is customer of Comcast, connected to Comcast network, and IP addresses used within school network are within Comcast's address block

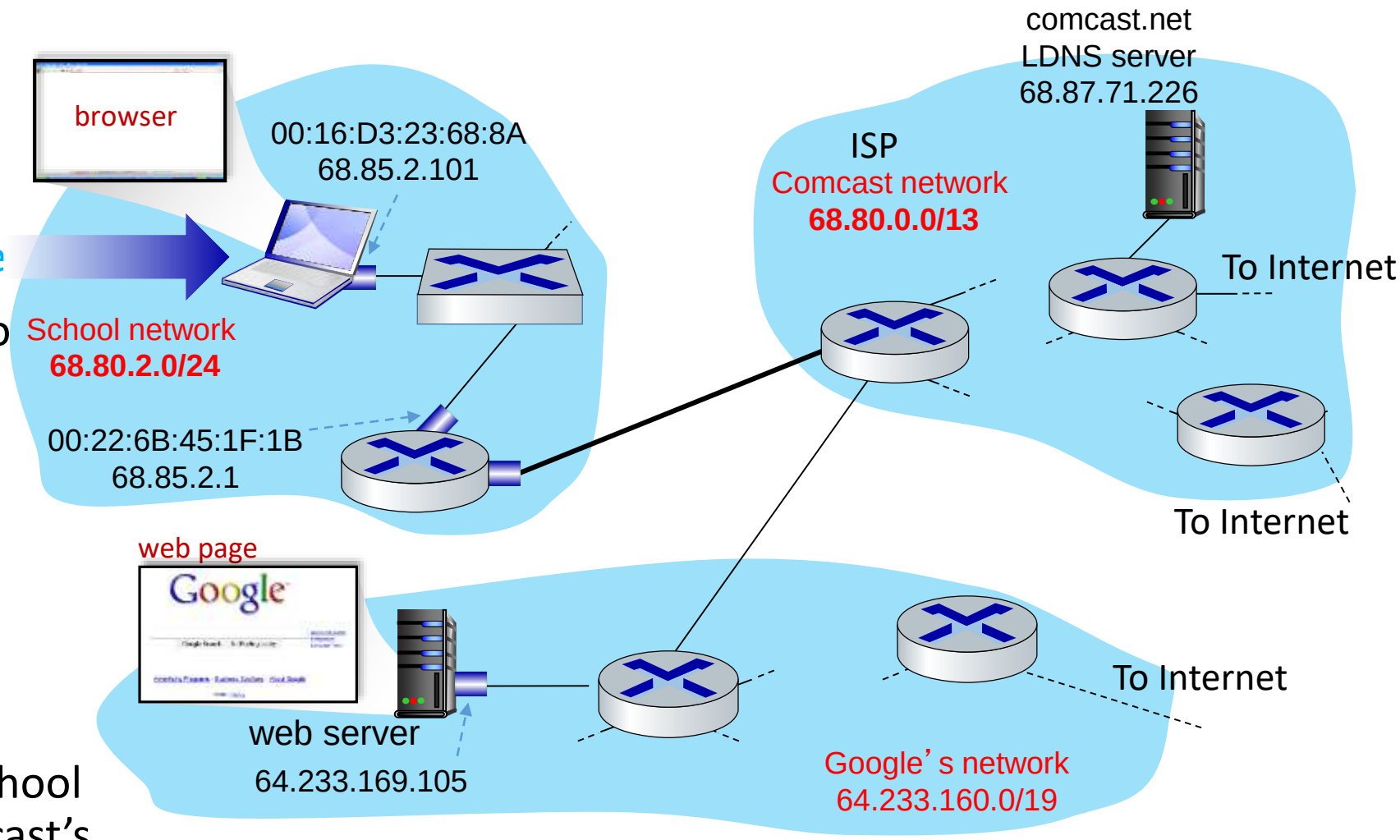
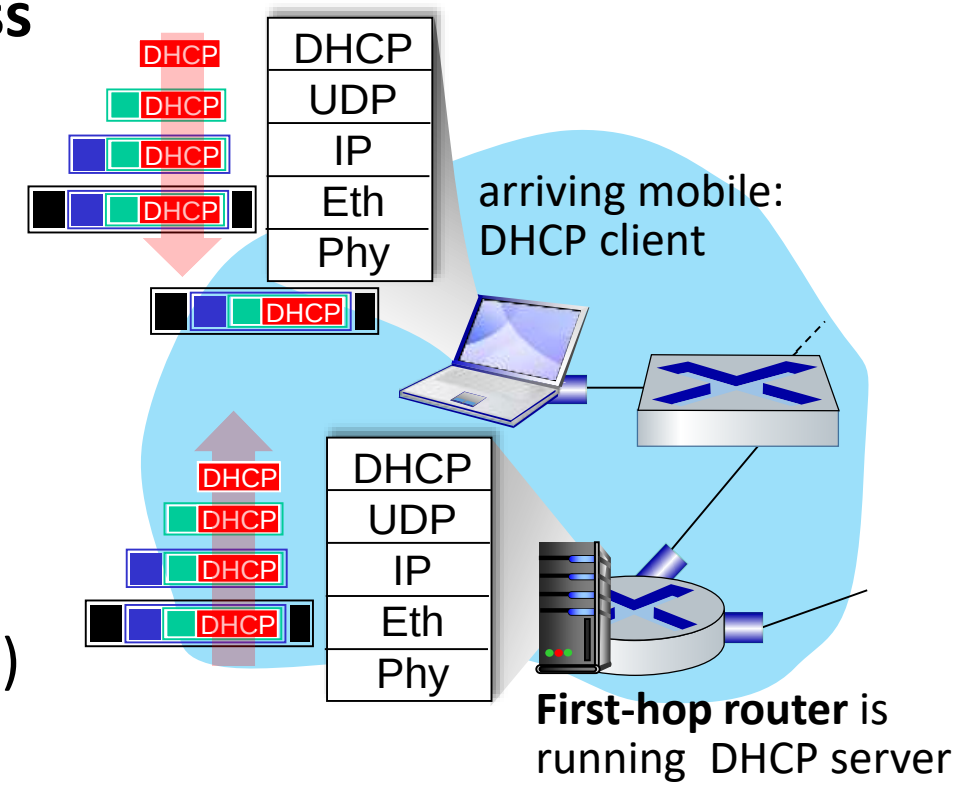


Figure 6.32 A day in the life of a Web page request: Network setting and actions

6.7.1 Getting Started: DHCP, UDP, IP, and Ethernet

- DHCP: Laptop needs to get its own **IP address** and its **Mask**, **address of first-hop router** (**default gateway**), address of **LDNS serves**
- DHCP request encapsulated in UDP destination port 67, encapsulated in IP, encapsulated in 802.3 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN (first frame sent by laptop), received at router running DHCP server (listening on port 67)
- In router, Ethernet demuxed to IP, IP demuxed to UDP, UDP demuxed to DHCP

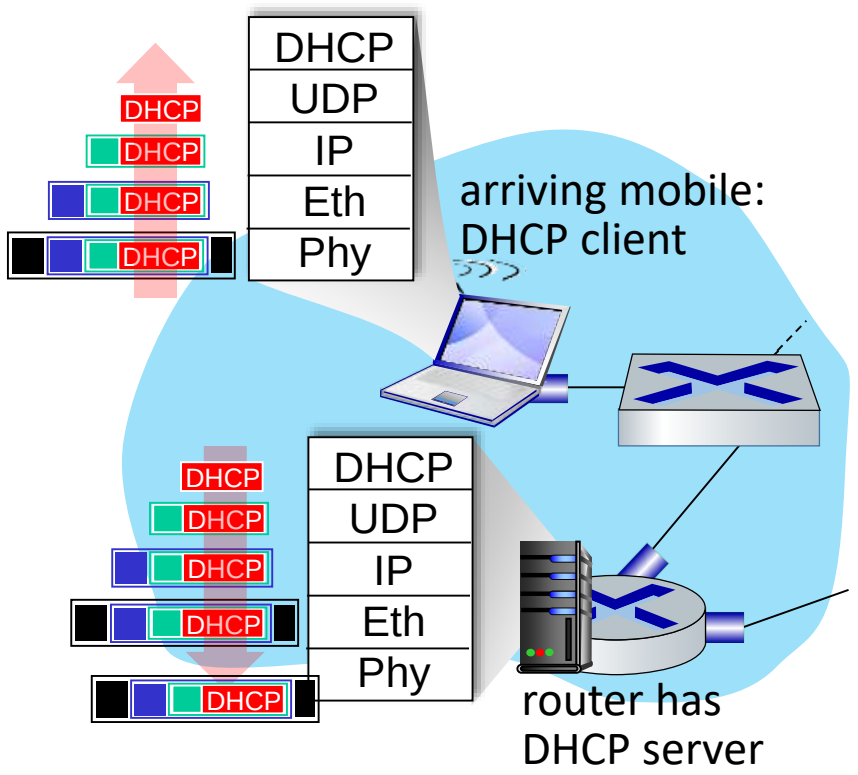


DHCP in client host

- Operating system of laptop creates a **DHCP request message** and puts this message within a **UDP segment** (destination port 67 and source port 68)
- UDP segment is placed within an **IP datagram** (destination IP address 255.255.255.255, IP broadcast, and source IP address of 0.0.0.0)
- IP datagram is placed within an **Ethernet frame** (destination MAC addresses FF:FF:FF:FF:FF:FF, MAC broadcast, and source MAC address is 00:16:D3:23:68:8A). This is a DHCP request (no DHCP discover)
- LAN switch broadcasts DHCP request frame on all outgoing ports, including port connected to router

DHCP ACK from server to client

- DHCP server formulates DHCP ACK containing client's IP address, subnet mask, IP address of first-hop router for client, name & IP address of LDNS servers
- Encapsulation at DHCP server, frame forwarded through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply
- Client now has **IP address**, **subnet mask**, name & addr of **LDNS** servers, IP address of its **first-hop router**



DHCP ACK in more details

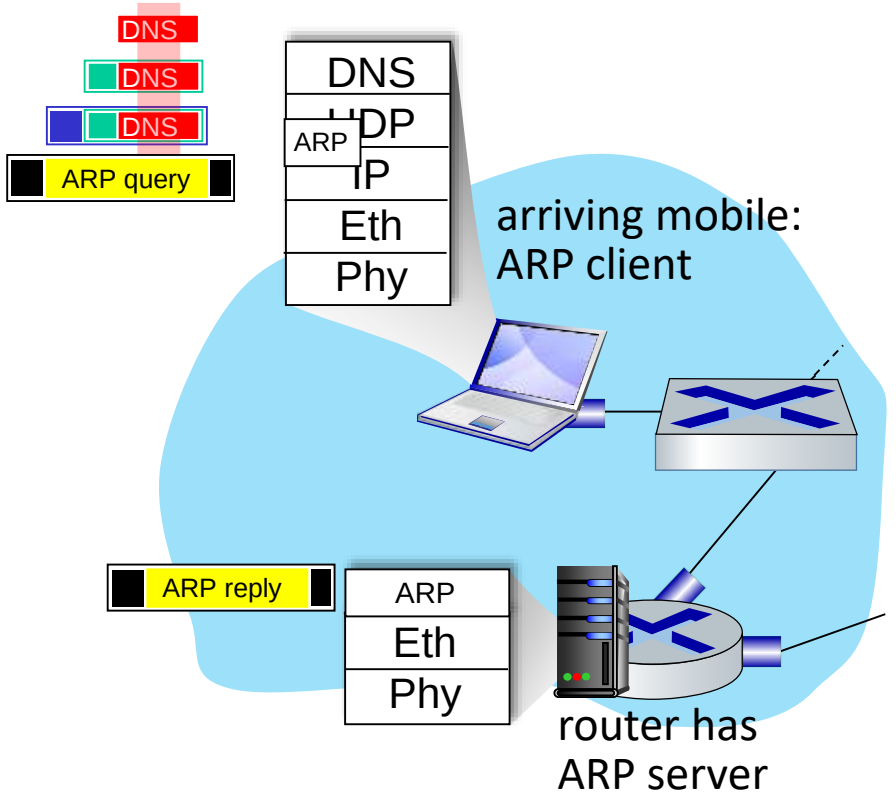
- DHCP server allocates IP addresses 68.85.2.101 in block 68.85.2.0/24
- DHCP server creates a DHCP ACK message containing 68.85.2.101, IP address of LDNS server (68.87.71.226), IP address for default gateway router (68.85.2.1), and subnet mask (68.85.2.0/24)
- DHCP message is put inside a UDP segment, which is put inside an IP datagram, which is put inside an Ethernet frame (DHCP ACK)
- DHCP ACK has a source MAC address 00:22:6B:45:1F:1B (router's interface to school network and a destination MAC address of Laptop: 00:16:D3:23:68:8A
- DHCP ACK **unicasts** by router to switch. Switch is **self-learning** and previously received an Ethernet frame from laptop, switch forwards DHCP ACK only to output port leading to laptop

6.7.2 Still Getting Started: DNS and ARP

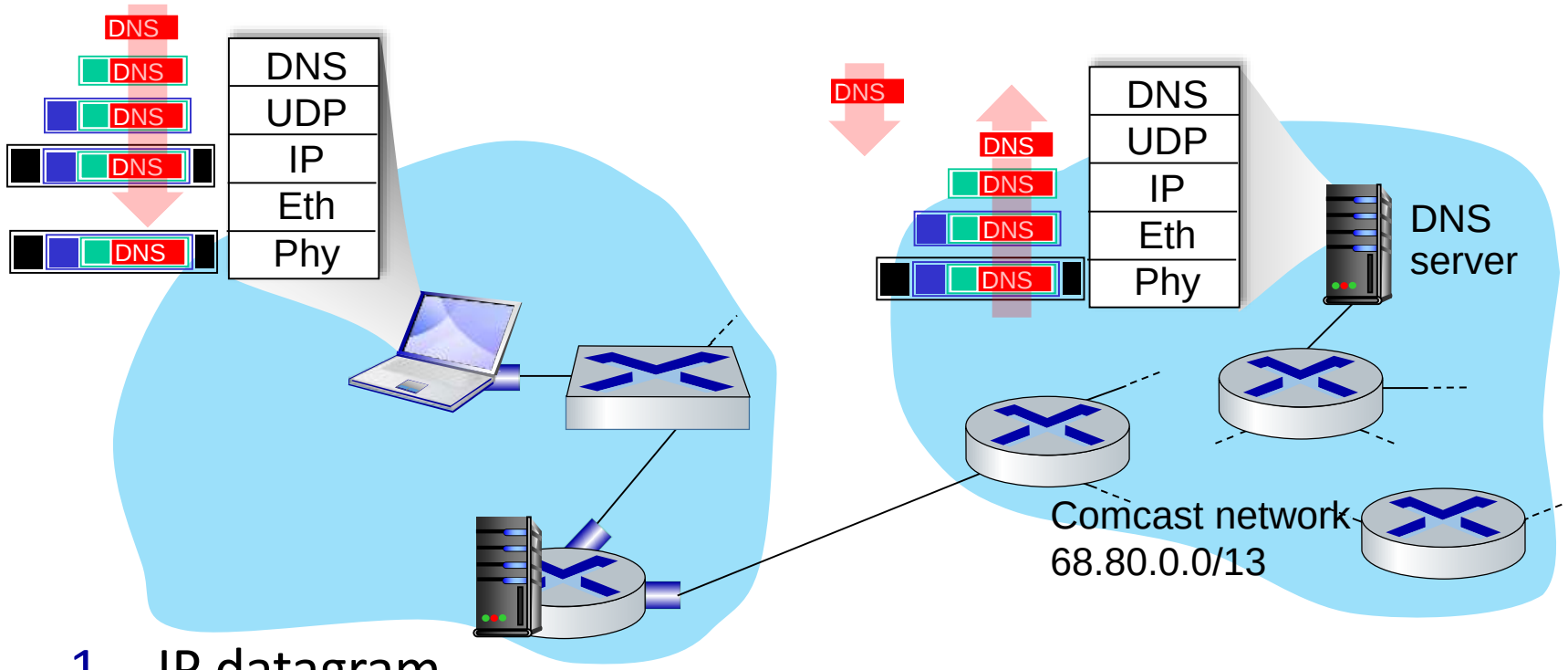
- User types: **www.google.com** into Web browser
- User's Web browser begins process by creating a **TCP socket** that will be used to send **HTTP request** to **www.google.com**
- In order to create socket, user's laptop uses **DNS protocol** to find IP address of **www.google.com**

DNS, ARP

- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- To send frame to router, need MAC address of router interface: ARP
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
- Client now knows MAC address of first hop router, so can now send frame containing DNS query



DNS



1. IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
2. IP datagram forwarded from school network into Comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
3. demuxed to DNS
4. DNS replies to client with IP address of www.google.com

DNS query from laptop

- Operating system on laptop creates a **DNS query message**, putting string "www.google.com" in question section of DNS message
- DNS query is placed within a UDP segment with a destination port of 53
- UDP segment is then placed within an IP datagram with an IP destination address of 68.87.71.226 (address of LDNS returned in DHCP ACK) and a source IP address of 68.85.2.101. 9
- IP datagram then placed in an Ethernet frame
- Frame will be sent (addressed, at link layer) to gateway router in user's school's network
- However, Bob's laptop doesn't know gateway router's MAC address
- So, laptop will need to use **ARP protocol**

ARP operation

- Laptop creates an **ARP query** message with a target IP address of 68.85.2.1 (default gateway), places ARP message within an Ethernet frame with a broadcast destination address (FF:FF:FF:FF:FF:FF) and sends to switch, which delivers frame to all connected devices, including the gateway router
- Gateway router receives frame containing ARP query message, and finds that target IP address of 68.85.2.1 in ARP message matches IP address of its interface
- Gateway router thus prepares an **ARP reply**, indicating that its MAC address of 00:22:6B:45:1F:1B corresponds to IP address 68.85.2.1
- ARP reply is put in an Ethernet frame, with a destination address of 00:16:D3:23:68:8A (laptop) and sends to switch, which delivers frame to laptop
- Now, laptop can complete a frame containing DNS query and send it out

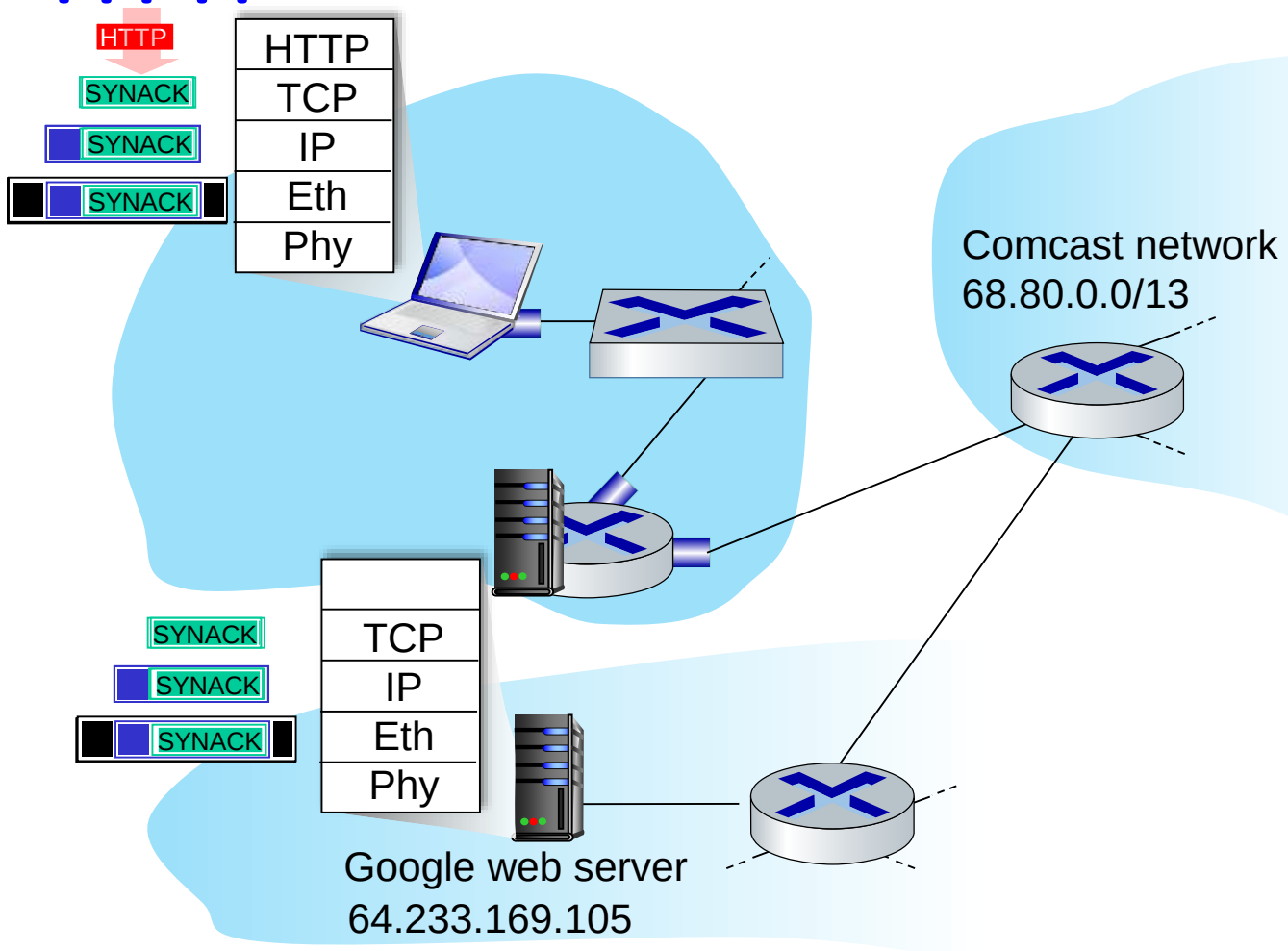
6.7.3 Still Getting Started: Intra-Domain Routing to the DNS Server

- Gateway router receives DNS query
- Router looks up destination address (68.87.71.226) and determines from its forwarding table that it should be sent to Comcast network
- IP datagram is placed inside a link-layer frame appropriate for link connecting **school's router to Comcast router**
- Router in Comcast forwards datagram toward DNS server using **forwarding table** entry that comes from Comcast's intra-domain protocol (such as **RIP**, **OSPF** or **IS-IS**)

LDNS resolves name

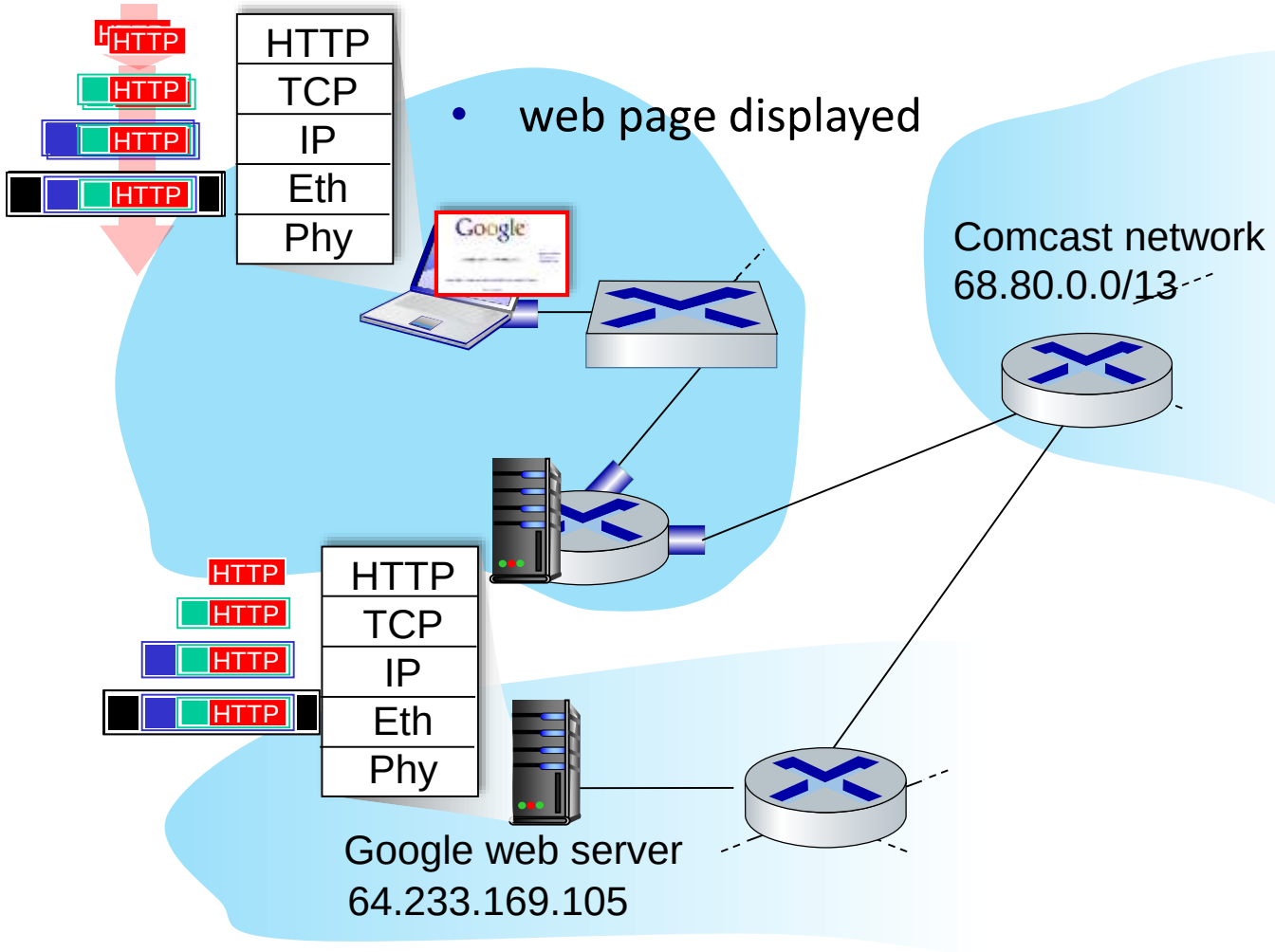
- LDNS server extracts DNS query, looks up name www.google.com in its DNS database, and finds cached **DNS resource record** that contains IP address for www.google.com
- LDNS server forms a **DNS reply message** in a UDP segment, in IP datagram addressed to laptop (68.85.2.101) in a frame, …
- Laptop receives and extracts IP address of server www.google.com
- Now laptop is ready to TCP contact to www.google.com server

6.7.4 Web Client-Server Interaction: TCP and HTTP



1. To send HTTP request, client first opens TCP socket to connect to web server socket
2. TCP SYN segment (step 1 in TCP 3-way handshake) inter-domain routed to web server
3. Web server responds with TCP SYNACK (step 2 in TCP 3-way handshake)
4. TCP connection established

HTTP



5. HTTP request sent into TCP via client socket in Laptop

6. IP datagram containing HTTP request **routed to** www.google.com

7. web server responds with HTTP reply (containing web page)

8. IP datagram containing **HTTP reply routed back to client**

TCP and HTTP

- Laptop has IP address of www.google.com, it can create **TCP socket** that will be used to send **HTTP GET**
- When User creates TCP socket, TCP in laptop perform a **three-way handshake** with TCP in www.google.com
- **TCP SYN** (destination port 80 for HTTP), sent to 64.233.169.105, using destination MAC address of 00:22:6B:45:1F:1B (default gateway router)
- Eventually, TCP SYN arrives at www.google.com
- A connection socket is created for TCP connection between Google HTTP server and Web browser in laptop
- TCP SYNACK is generated and sent for laptop

TCP and HTTP

- TCP SYNACK arrives at TCP socket in laptop, then socket enters connected state
- TCP socket in laptop now ready to send bytes to www.google.com,
- Browser creates HTTP GET message containing URL to be fetched
- HTTP GET message is then written into socket, GET message becomes payload of a TCP segment…
- HTTP server at www.google.com reads HTTP GET from TCP socket, creates an **HTTP response message**, places Web page HTML in body of HTTP response message, and sends into TCP socket
- Web browser reads HTTP response from socket, extracts HTML for Web page, and finally displays the Web page

Contents

6.8 Summary

- Principles behind data link layer services: Error detection, correction
  - Sharing a broadcast channel: multiple access
  - Link layer addressing
- Basic service of link layer: moving a network-layer datagram from one node (host, switch, router, WiFi access point) to an adjacent node
- Instantiation, implementation of various link layer technologies
  - Ethernet
  - Switched LANS, VLANs
  - virtualized networks as a link layer: MPLS
- Synthesis: a day in life of a web request

6.8 Summary

- Links
  - Point-to-point link: single sender and receiver communicating over a single “wire.”
  - A multiple access link: a link shared among many senders and receivers
  - MPLS “link”: connecting two adjacent nodes (for example, two IP routers that are adjacent in an IP sense, that they are next-hop IP routers toward some destination) (MPLS is actually a network by itself)
- Next 2 chapters cover wireless networking and network security
- These 2 topics do not fit conveniently into any one layer; indeed, each topic crosscuts many layers
- Understanding these topics requires a firm foundation in all layers of protocol stack

Contents

NORM ABRAMSON AND **ALOHANET**

- Norm Abramson, a PhD engineer, had a passion for surfing and an interest in packet switching. This combination of interests brought him to the University of
- Hawaii in 1969. Hawaii consists of many mountainous islands, making it difficult to install and operate land-based networks
- Norm Abramson, a PhD engineer, thought about how to design a network that does packet switching over radio
- Network he designed had one central host and several secondary nodes scattered over Hawaiian
- The network had two channels, each using a different frequency band
  - Downlink channel broadcasted packets from central host to secondary hosts
  - Uplink channel sent packets from secondary hosts to central host

NORM ABRAMSON AND ALOHANET

- In addition to sending informational packets, central host also sent on downlink channel an acknowledgment for each packet successfully received from secondary hosts
- Because secondary hosts transmitted packets in a decentralized fashion, collisions on uplink channel inevitably occurred
- This observation led Abramson to devise ALOHA protocol
- In 1970, with funding from ARPA, Abramson connected his ALOHAnet to ARPAnet
- Abramson's work is important not only because it was first example of a radio packet network, but also because it inspired Bob Metcalfe
- A few years later, Metcalfe modified ALOHA protocol to create CSMA/CD protocol and Ethernet LAN

KEEPING THE LAYERS INDEPENDENT

- Why hosts and router interfaces have MAC addresses in addition to network-layer addresses?
- In order for layers to be largely independent building blocks in a network architecture, different layers need to have their own addressing scheme
- Three types of addresses: host names for application layer, IP addresses for network layer, and MAC addresses for link layer

So:

1. LANs are designed for arbitrary network-layer protocols, not just for IP and Internet
2. If adapters were assigned IP addresses rather than “neutral” MAC addresses, then adapters would not easily be able to support other network-layer protocols (for example, IPX or DECnet)
3. If adapters were to use network-layer addresses instead of MAC addresses, network-layer address would have to be stored in adapter RAM and reconfigured every time the adapter was moved (or powered up)
4. If adapters not use any addresses, and have each adapter pass data (typically, an IP datagram) of each frame it receives up protocol stack Network layer could then check for a matching network-layer address. One problem with this option is that host would be interrupted by every frame sent on LAN, including by frames that were destined for other hosts on same broadcast LAN

BOB METCALFE AND ETHERNET

- Bob Metcalfe: As a PhD student at Harvard University in early 1970s, worked on ARPAnet at MIT, he also became exposed to Abramson's work on ALOHA and random access protocols
- After PhD, he got a job at Xerox Palo Alto Research Center (Xerox PARC), he became exposed to Alto computers, which in many ways were forerunners of personal computers of 1980s
- He saw need to network these computers in an inexpensive manner
- So armed with his knowledge about ARPAnet, ALOHAnet, and random access protocols, Metcalfe, along with colleague David Boggs, invented Ethernet

BOB METCALFE AND ETHERNET

- Metcalfe and Boggs's original Ethernet ran at 2.94 Mbps and linked up to 256 hosts separated by up to one mile
- Metcalfe and Boggs succeeded at getting most of researchers at Xerox PARC to communicate through their Alto computers
- Metcalfe forged an alliance to establish Ethernet as a 10 Mbps Ethernet standard, ratified by IEEE
- Xerox did not show much interest in commercializing Ethernet
- In 1979, Metcalfe formed 3Com company which developed and commercialized networking technology, including Ethernet technology
- 3Com developed and marketed Ethernet cards in early 1980s for immensely popular IBM PCs

SNIFFING A SWITCHED LAN: SWITCH POISONING

- Consider a switched LAN ir
  there is an entry for host B
  host B
- If host C happens to be rur
  frame However, switches b
  C can still sniff some frame
- Sniffer will be able sniff all
  address FF–FF–FF–FF–FF–F
- **Switch poisoning** attack: A
  different bogus source MA
  for MAC addresses of legiti
  - This causes switch to broad