Lecture 1

Tuesday, May 9, 2023

Quizzes

- At the end of chapters
- In class, closed book
- Best 4 out of 5
- No make-up quiz for missed quizzes

Midterm

- Date: June 16
- Coverage: Chapter 1 and 2, half of Chapter 3

TCP/IP protocol

- Application - L5 (HTTP, FTP)
- Transport - L4 (TCP)
- Network - L3 (IP)
- Datalink - L2 (Ethernet)
- Physical - L1
    - Not covered in this course

Order of layers

- Sender: L5 to L1
- Receiver: L1 to L5

1-3 The Internet: a “nuts and bolts” view

- Devices
    - Host distributed applications at the Internet's "edge"
    - Called hosts or end systems
- Communication links
    - Transfer data from sources to sinks (destinations)
    - Can be guided (e.g. wires) or unguided (e.g. access point)
    - There can be different types of links within a path from a source to a destination
    - Can have different transmission rates (bandwidth)
- Packet switches
    - Divide data into small chunks (packets)
    - Packets have headers and data
    - The protocol in each layer define the format of headers
    - Packet names in different layers
        - L5 - message
        - L4 - segment
        - L3 - datagram
        - L2 - frame
    - Packet switches forward packets
    - Include routers and switches
        - Routers operate at L3 use IP address
        - Switches operate at L2 use physical addresses

1-4 The Internet: a "services" view

- Provides a programming interface (e.g. socket) to distributed applications
- Allows them to "connect to" (use) Internet transport services

1-6 Internet standards

- RFC (Request for Comments): specifications of protocols
    - Managed by IETF (Internet Engineering Task Force)

1-7: What's a protocol

- Protocols define the format, order of messages sent and received among network entities, and actions taken on message transmission, receipt

1-8 Protocol example

- A TCP connection must be established before using a GET request to fetch a webpage
- The TCP connection request and response follow the formats defined in the protocols

1-11 Access networks

- Allow devices (at network edges) to connect to the Internet
- Can be wired, wireless or hybrid

1-12 Network core

- Interconnected routers

1-14 Access networks: cable-based access

- Frequency division multiplexing (FDM) - different channels transmitted in different frequency bands
- Time division multiplexing (TDM) - different channels transmitted in different time slots
- Shared medium => bandwidth is also shared

1-16 Access networks: digital subscriber line (DSL)

- Use existing telephone lines to central DSLAM
    - Data over DSL phone line goes to Internet
    - Voice over DSL phone lien goes to telephone net

Lecture 2

Thursday, May 11, 2023

1-27 Two key network-core functions

- A router can have multiple output links
- The packet's header contains the destination IP address
- Each router has a forwarding table that uses part of the destination address as index
- The lookup process is called **forwarding** or **switching** (data plane)
    - Local action - move arriving packets from router's input link to appropriate router output link
- **Routing** is the populating of the forwarding table (control plane)
    - Global action - determine source-destination paths taken by packets
    - Routing algorithms

1-30 Packet-switching: store-and-forward

- Entire packet must arrive at router before it can be transmitted on next link
    - To ensure integrity of the packet
    - Cannot send a corrupted packet
- Packet transmission delay = L/R
- If N is the # of links, the total transmission delay is NL/R seconds
- If N is the # of links and P is the # of packets, the total transmission delay is (N + P - 1)L/R

1-33 Alternative to packet switching: circuit switching

- Packet switching is on-demand
- Circuit switching reserves resources
- Packet transmission delay = L/R

1-35 Packet switching vs. circuit switching

- Suppose there are 35 users
- Circuit switching: supports (1Gb/s) / (100 Mb/s) = 10 users
- Packet switching: supports 35 users

1-49 Packet delay

- **Transmission delay**: time needed for the router to put the packet on the link
- **Propagation delay**: time needed for the packet to be transferred through the link
- These two are very different

Lecture 3

Tuesday, May 16, 2023

Chapter 1 Quiz

- Thursday, May 25

1-48 Packet delay

- Processing delay
  - Same for packets of the same size
- Queueing delay
  - Varies from packet to packet
- Transmission delay
  - Depends on the link
- Propagation delay
  - Depends on the physical link

1-52 Packet queueing delay

- Golden rule for traffic intensity
  $\frac{L \cdot a}{R} \leq 1$
- Having a traffic intensity > 1 can lead to packet loss
- Example 1: arrival rate = L/R
  - Average queueing delay = L/R
- Example 2: burst of N packets at arrive rate = NL/R
  - Queueing delay for 1st packet = 0
  - Queueing delay for 2nd packet = L/R
  - Queueing delay for 3rd packet = 2L/R
  - …
  - Queueing delay for Nth packet = (N-1)L/R
  - Total delay for one burst = 0 + L/R + 2L/R + … + (N-1)L/R
  - Average delay for one burst = total delay / N
  - By the time the second burst arrives, the first burst has been transmitted
  - Average delay for N bursts = average delay for one burst

1-53 "Real" Internet delays and routes

- Traceroute
  - Sends IP packets with a specific TTL and UDP protocol (in the transport layer) to a port that is unlikely to be an actual destination
  - Time to Live (TTL): when to drop the packet
  - If there is no TTL then the packet may live forever in the network
  - When the TTL becomes 0, the router drops the packet and sends an ICMP packet (in the network layer) to the source
    - Includes the IP address and the time it takes to get the response

1-56 Throughput

- Average throughput = # bits / # sec

1-57 Throughput

- $R_s < R_c$
  - Throughput is constrained by R_s (bottleneck link)
  - Throughput = R_s
- $R_s > R_c$
  - Throughput is constrained by R_c (bottleneck link)
  - Throughput = R_c
- Bottleneck link: link on end-end path that constrains end-end throughput
- Throughput = $\min\{R_s, R_c\}$
- If there are N links, then the throughput is $\min\{R_s, R_c, R_1, R_2, \dots, R_N\}$

1-58 Throughput: network scenario

- Example: $R_c = 2\ \text{Mbps}$, $R_s = 1\ \text{Mbps}$, $R = 5\ \text{Mbps}$, 10 links (fairly) share the link with rate R
  - Throughput = 5 Mbps / 10 = 500 Kbps

1-62 Packet interception

- Does not inject anything into the network, only listens for packets passing by
- Hard to detect

1-64 Denial of service (DoS)

- Creates bonus traffic so that the rate exceeds the bandwidth of a link

Chapter 1 Summary

Sunday, May 21, 2023

Devices
- Hosts (aka end systems)
- Running network apps at the Internet's edges

Packet switches
- Forward packets
- Routers and switches

A services view
- Infrastructure that provides services to applications
- Provides programming interface to distributed applications

Protocol
- Define the format, order of messages sent and received among network entities, and actions taken on message transmission, receipt

Network edge
- Clients and servers

Wireless access networks
- Shared wireless access network connects end system to router
- Wireless local area networks
    - Typically within or around building (~100 ft)
- Wide-area cellular access networks
    - Provided by mobile, cellular network operator (10's km)

Access networks: enterprise networks
- Companies, universities, etc.
- Mix of wired, wireless link technologies
    - Ethernet: wired access
    - Wi-Fi: wireless access points

Access networks: data center networks
- High-bandwidth links connect hundreds to thousands of servers together, and to Internet

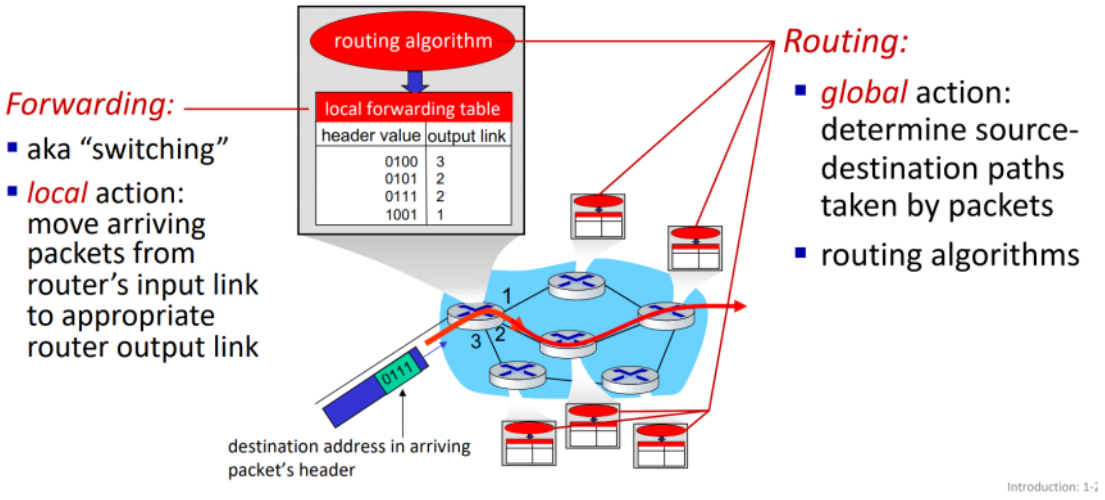
Host: sends packets of data
- Takes application message
- Breaks into smaller chunks called **packets** of length L bits
- Transmits packets into access network at transmission rate R
- Transmission rate R also called
    - Link capacity
    - Link bandwidth
- Packet transmission delay = L/R

The network core

- Mesh of interconnected routers
- **Packet-switching**: hosts break application-layer messages into packets
- Network **forwards** packets from one router to the next, across links on path from source to destination

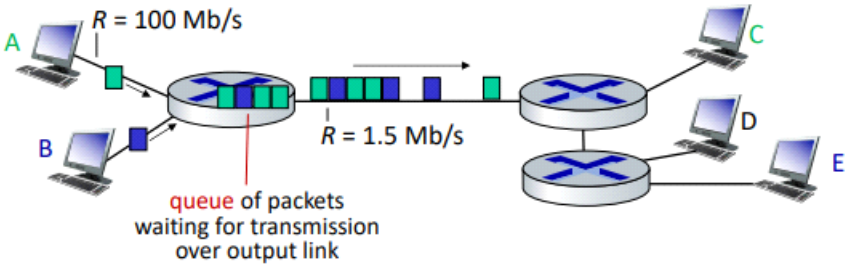
Two key network-core functions

- Forwarding
  - Aka switching
  - Local action: move arriving packets from router's input link to appropriate router output link
- Routing
  - Global action: determine source-destination paths taken by packets
  - Routing algorithms



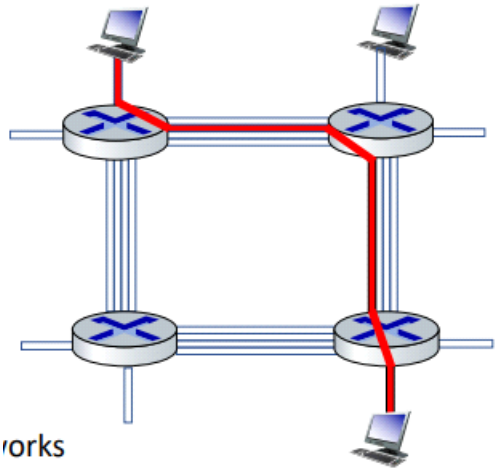
Packet-switching: queueing

- If arrival rate to link exceeds transmission rate of link for some period of time, packets will queue, waiting to be transmitted on output link
- Packets can be dropped (lost) if there is no space in the queue



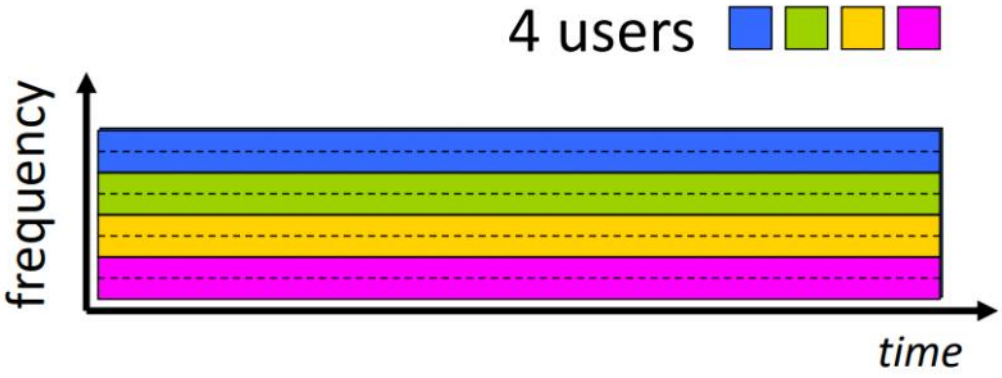
Alternative to packet switching: circuit switching

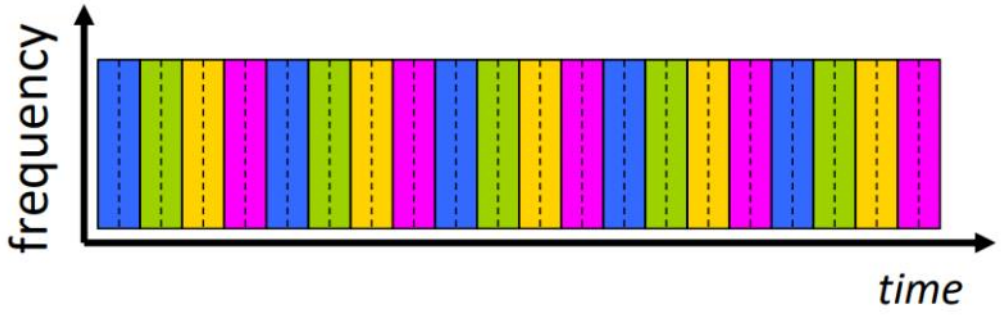
- End-end resources reserved for "call" between source and destination
- Dedicated resources: no sharing
- Circuit segment is idle if not used by call
- Commonly used in traditional telephone networks



Circuit switching: FDM and TDM

- Frequency Division Multiplexing (FDM)
  - Each call allocated its own band, can transmit at max rate of that band
- Time Division Multiplexing (TDM)
  - Each call allocated periodic slots
  - Can transmit at maximum rate of frequency band during its times slots





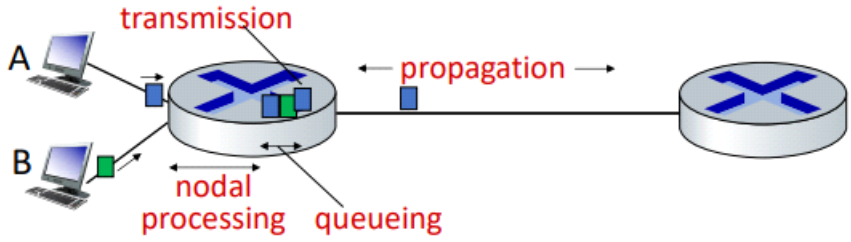
Packet switching compared to circuit switching

- Great for bursty data - simpler, no call setup
- Excessive congestion possible

Packet delay: four sources

- Nodal processing

    - Check bit errors
    - Determine output link
- Queueing delay
    - Time waiting at output link for transmission
    - Depends on congestion level of router
- Transmission delay
    - L: packet length (bits)
    - R: link transmission rates (bps)
    - L/R
- Propagation delay
    - d: length of physical link
    - s: propagation speed
    - d/s



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Packet queueing delay (revisited)

- a: average packet arrival rate
- L: packet length (bits)
- R: link bandwidth (bit transmission rate)
- Traffic intensity = La/R

Traceroute

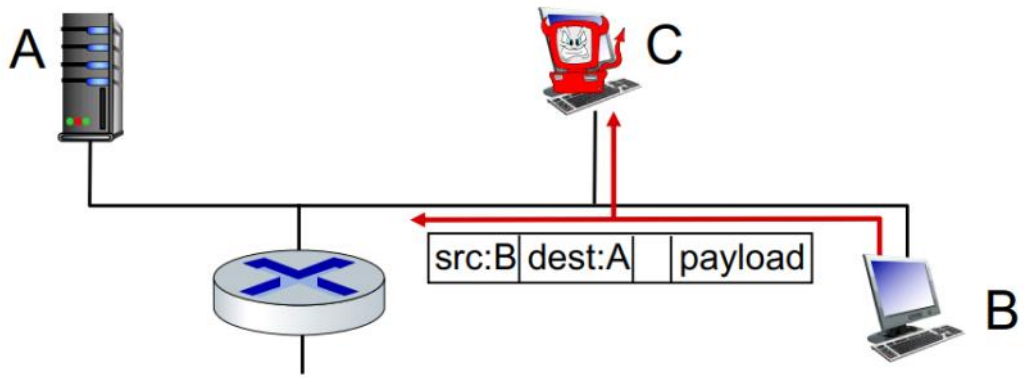
- Provides delay measurement from source to router along en-end Internet path towards destination
- For all i:
    - Sends three packets that will reach router i (with TTL field = i)
    - Router i will return packets to sender
    - Sender measures time interval between transmission and reply

Packet loss

- Queue (buffer) preceding link has finite capacity
- Packet arriving to full queue gets dropped

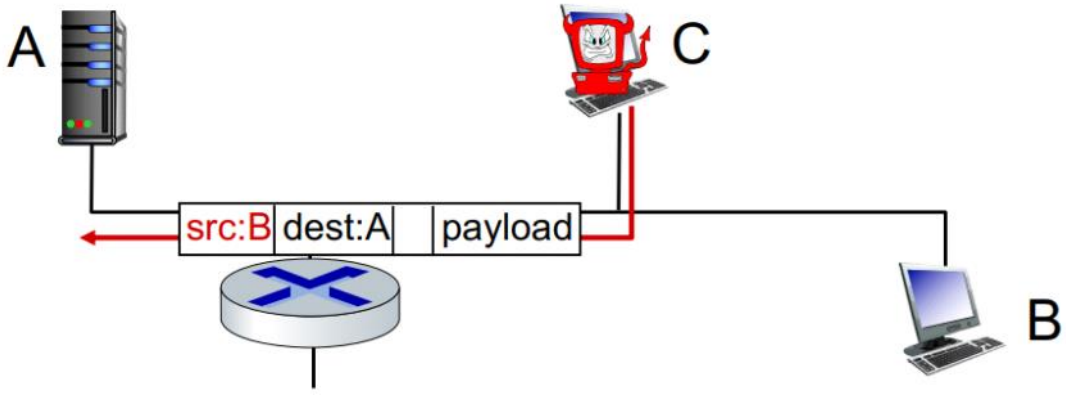
Packet interception

- Packet "sniffing"
- Broadcast media (e.g. shared ethernet, wireless)
- Reads/records all packets passing by



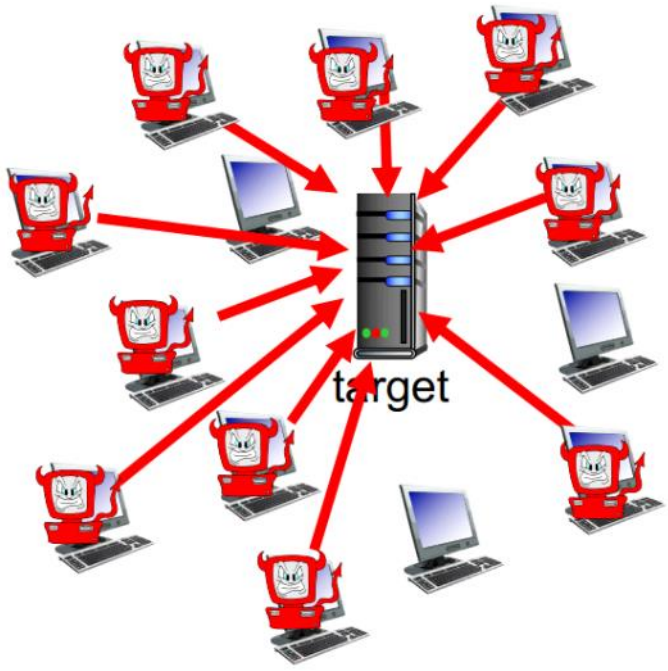
Fake identity

- IP Spoofing
- Injection of packet with false source addresses



Denial of service

- Denial of Service: attackers make resources unavailable to legitimate traffic by overwhelming resource with bogus traffic



Lines of defense

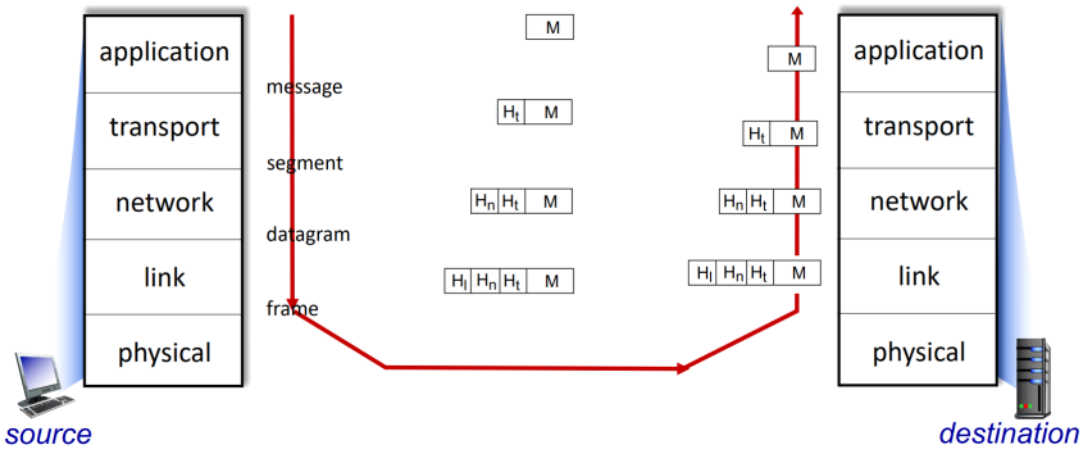
- Authentication
  - Proving you are who you say you are
- Confidentiality
  - Encryption
- Integrity checks
  - Digital signatures
- Access restrictions
  - Password-protected VPNs
- Firewalls

Protocol layers and reference layers

- Each layer implements a service
  - Via its own internal-layer actions
  - Relying on services provided by layers below
- Explicit structure allows identification, relationship of system's pieces
- Modularization eases maintenance, updating of system
  - Changes in layer's service implementation is transparent to the rest of the system

Layered Internet protocol stack

- Application
  - Supports network applications
  - HTTP, IMAP, SMTP, DNS
- Transport
  - Process-process data transfer
  - TCP, UDP
- Network
  - Routing of datagrams from source to destination
  - IP
- Link
  - Data transfer between neighbouring network elements
  - Ethernet, WiFi, PPP
- Physical
  - Bits "on the wire"



Packets in each layer

- Application: messages
- Transport: segments
- Network: datagram
- Link: frame

Chapter 1 Review Questions

Saturday, June 10, 2023 21:16

R1.

There is no difference between a host and an end system.

Several types of end systems:

- Phones
- PCs
- TVs
- Server

A web server is an end system.

R2.

R3.

We need standards so that everyone agree on what each protocol does to create systems and products that interoperate.

R4.

Home: DSL, Cable, FTTH, and 5G Fixed Wireless
Enterprise: WIFI and Ethernet
Wide area: 3G, 4G and 5G

R5.

HFC is shared among users.

On the downstream channel, all packets emanate from a single source, namely, the head end. Thus there are no collisions in the downstream channel.

R6.

4G? 5G?

Dial-up, DSL, cable mode, fibre-to-the-home

R7. Dial-up modems and DSL both use the telephone line (a twisted-pair copper cable) as their transmission medium. Why then is DSL much faster than dial-up access?

R8. What are some of the physical media that Ethernet can run over?

Twisted-pair copper wire.

It can also run over fibers optic links.

R9. HFC, DSL, and FTTH are all used for residential access. For each of these access technologies, provide a range of transmission rates and comment on whether the transmission rate is shared or dedicated.

R10. Describe the different wireless technologies you use during the day and their characteristics. If you have a choice between multiple technologies, why do you prefer one over another?

SECTION 1.3

R11.

$$\frac{L}{R1} + \frac{L}{R2}$$

R12.

Circuit-switched network

- No resource sharing => no queueing delays etc.
- Guarantee a certain amount of end-to-end bandwidth for the duration of a call

TDM

- You get the maximum bandwidth during your timeslot

FDM

- Requires sophisticated analog hardware to shift signal into appropriate frequency bands

R13.

a. 2 Mbps / 1 Mbps = 2 users
b. Because the link can supports the simultaneous transmission of two users (packets)
c. 0.2
d. $0.2^3 = 0.008 = 0.8\%$

R14.

If they do not peer with each other, then when they send traffic to each other they have to send the traffic through a provider ISP, to which they have to pay for carrying the traffic.

An Internet Exchange Points (IXP) is a meeting point where multiple ISPs can connect and/or peer together. An ISP earns its money by charging each of the ISPs that connect to the IXP a relatively small fee.

R15.

Content providers often bypass higher tier ISPs to connect privately with lower tier ISPs.

R16.

- Processing delay - constant for packets of the same length
- Queueing delay - depends on the number of packets in queue
- Transmission delay - depends on the packet length and transmission rate
- Propagation delay - constant for routes with the same physical distance

R17. Visit the Transmission Versus Propagation Delay interactive animation at the Companion Website. Among the rates, propagation delay, and packet sizes available, find a combination for which the sender finishes transmitting before the first bit of the packet reaches the receiver. Find another combination for which the first bit of the packet reaches the receiver before the sender finishes transmitting.

R18.

Wireless

$$\frac{1.5\ kb}{2000\ kbps} + \frac{1000\ m}{\left(3 \times\ 10^8 \frac{m}{s}\right)} =\ 0.000753s$$

Wired

$$\frac{1.5\ kb}{100000\ kbps} + \frac{1000\ m}{\left(2 \times\ 10^8 \frac{m}{s}\right)} = 2 \times 10^{-5} s$$

R19.

a. 500 Kbps

b.

$$\frac{4000\ kb}{500\ kb} =\ 8\ packets$$

$$8 \times \frac{500}{500} + \frac{500}{2000} + \frac{500}{1000} = 8.75\ s$$

64 seconds?

c.

The throughput would be 100 kbps.

$$\frac{4000\ kb}{100\ kb} = 40\ packets$$

$$\frac{100}{500} + 40 \times \frac{100}{100} + \frac{100}{1000} = 40.3\ s$$

320 seconds?

R20.

The end system creates a packet by adding a header at the transport layer and another header at the network layer. The information contained in the header includes an IP address and a port number.

The router uses the IP address to determine the link.

SECTION 1.5

R22. If two end-systems are connected through multiple routers and the data-link level between them ensures reliable data delivery, is a transport protocol offering reliable data delivery between these two end-systems necessary? Why?

R23.

L5 Application Layer - for developers to create distributed applications on ends
L4 Transport Layer - for communication between processes running on different hosts
L3 Network Layer - for packet forwarding based on the IP address
L2 Data-link Layer - for packet transfer based on MAC addresses
L1 Physical Layer - physical bits in wires

R24.

Encapsulation is taking packets from layers above and wrapping them in headers.

Decapsulation is taking packets from layers below and removing the headers.

They are needed because this gives the Internet a good way to structure. It also introduces abstraction, I,. developers working on distributed applications in L5 do not need to understand how the layers below work. Instead, they use programming interfaces (i.e. sockets) from the transport layers. Services are provided by layers below to layers above.

R25.

A router processes in the network layer.
A switch processes in the data-link layer.
A host processes in the application layer.

A router processes in the network, link and physical layers.
A switch processes in the link and physical layers.
A host processes in all five layers.

Chapter 1 Problems

Sunday, June 11, 2023

P3.

a.

A circuit-switched network would be more appropriate for this application, since it transmits data at a steady rate, instead of sending data at, so using a circuit-switched network can help the application reserve a resource to ensure throughput and timing.

b.

No, since if the traffic intensity is less than 1, there will be no queueing delay.

P5. Review the car-caravan analogy in Section 1.4. Assume a propagation speed of 100 km/hour.

a. Suppose the caravan travels 175 km, beginning in front of one tollbooth, passing through a second tollbooth, and finishing just after a third tollbooth. What is the end-to-end delay?

$$10 \times 0.2 \times 2 + \frac{175}{100} \times 60 = 109 \text{ minutes}$$

b. Repeat (a), now assuming that there are eight cars in the caravan instead of ten

$$8 \times 0.2 \times 2 + \frac{175}{100} \times 60 = 108.2 \text{ minutes}$$

P6. This elementary problem begins to explore propagation delay and transmis sion delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B.

a. Express the propagation delay, d_{prop}, in terms of m and s.

$$\frac{d}{s}$$

b. Determine the transmission time of the packet, dtrans, in terms of L and R.

$$\frac{L}{R}$$

c. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

$$\frac{L}{R} + \frac{d}{s}$$

d. Suppose Host A begins to transmit the packet at time t = 0. At time t = dtrans, where is the last bit of the packet?

The last bit of the packet has been pushed to the link.

e. Suppose dprop is greater than dtrans. At time t = dtrans, where is the first bit of the packet?

The first bit of the packet is in the link.

f. Suppose dprop is less than dtrans. At time t = dtrans, where is the first bit of the packet?

The first bit of the packet is in the link.

g. Suppose s = 2.5×10^8, L = 1500 bytes, and R = 10 Mbps. Find the distance d so that dprop equals dtrans.

$$d_{trans} = \frac{L}{R} = \frac{1500 \times 8}{10^6}$$

$$d_{trans} = d_{prop} \Leftrightarrow \frac{L}{R} = \frac{d}{s} \Leftrightarrow d = \frac{sL}{R} = \frac{2.5 \times 10^8 \times 1500 \times 8}{1 \times 10^7} = 0.0324\ m$$

P7. In this problem, we consider sending real-time voice from Host A to Host B over a packet-switched network (VoIP). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 56-byte packets. There is one link between Hosts A and B; its transmission rate is 10 Mbps and its propagation delay is 10 msec. As soon as Host A gathers a packet, it sends it to Host B. As soon as Host B receives an entire packet, it converts the packet's bits to an analog signal. How much time elapses from the time a bit is created (from the original analog signal at Host A) until the bit is decoded (as part of the analog signal at Host B)?

P8. Suppose users share a 3 Mbps link. Also suppose each user requires 150 kbps when transmitting, but each user transmits only 10 percent of the time. (See the discussion of packet switching versus circuit switching in Section 1.3.)

a. When circuit switching is used, how many users can be supported?

$$\frac{3000\ kbps}{150\ kbps} = 20$$

b. For the remainder of this problem, suppose packet switching is used. Find the probability that a given user is transmitting

0.1

c. Suppose there are 120 users. Find the probability that at any given time, exactly n users are transmitting simultaneously. (Hint: Use the binomial distribution.)

$$\binom{120}{n} 0.1^n \cdot 0.9^{120-n}$$

d. Find the probability that there are 51 or more users transmitting simultaneously.

$$1 - \sum_{i=0}^{50} \binom{120}{i} 0.1^i \cdot 0.9^{120-i}$$

P9. Consider the discussion in Section 1.3 of packet switching versus circuit switching in which an example is provided with a 1 Mbps link. Users are generating data at a rate of 100 kbps when busy, but are busy generating data only with probability p = 0.1. Suppose that the 1 Mbps link is replaced by a 1 Gbps link.

a. What is N, the maximum number of users that can be supported simultaneously under circuit switching?

10000

b. Now consider packet switching and a user population of M users. Give a formula (in terms of p, M, N) for the probability that more than N users are sending data.

$$1 - \sum_{i=0}^{N} \binom{M}{i} 0.1^N \cdot 0.9^{M-N}$$

P10. Consider the network illustrated in Figure 1.16. Assume the two hosts on the left of the figure start transmitting packets of 1500 bytes at the same time towards Router B. Suppose the link rates between the hosts and Router A is 4-Mbps. One link has a 6-ms propagation delay and the other has a 2-ms propagation delay. Will queuing delay occur at Router A?

No arrival rate?

P11. Consider the scenario in Problem P10 again, but now assume the links between the hosts and Router A have different rates R1 and R2 byte/s in addition to different propagation delays d1 and d2. Assume the packet lengths for the two hosts are of L bytes. For what values of the propagation delay will no queuing delay occur at Router A?

No arrival rate?

P12. Consider a client and a server connected through one router. Assume the router can start transmitting an incoming packet after receiving its first h bytes instead of the whole packet. Suppose that the link rates are R byte/s and that the client transmits one packet with a size of L bytes to the server. What is the end-to-end delay? Assume the propagation, processing, and queuing delays are negligible. Generalize the previous result to a scenario where the client and the server are interconnected by N routers.

P13. (a) Suppose N packets arrive simultaneously to a link at which no packets are currently being transmitted or queued. Each packet is of length L and the link has transmission rate R. What is the average queuing delay for the N packets?

$$\frac{L/R + 2L/R + 3L/R + \cdots + (N-1)L/R}{N}$$

(b) Now suppose that N such packets arrive to the link every LN/R seconds. What is the average queuing delay of a packet?

$$\frac{L/R + 2L/R + 3L/R + \cdots + (N-1)L/R}{N}$$

Lecture 4

Tuesday, May 16, 2023

2-5 Creating a network app

- Applications that reside on end systems need to implement all layers while routers and switches don't

2-6 Client-server paradigm

- The web consists of clients, servers, contents, protocol
- HTTP is the protocol used
- Server
  - Always on
  - Permanent IP address
  - Needs scalability
- Client
  - May not be always on
  - May have dynamic IP addresses
- The communication is always between a client and a server

2-7 Peer-peer architecture

- Examples: BitTorrent, Skype

2-8 Processes communicating

- Socket
  - The interface for the application layer to use the service provided by the layers below
  - The developers control the applications
  - The layers below are taken care of by the OS
  - The destination is specified by an IP address and a port (specific process within a host)

2-10 Addressing processes

- Identifiers includes both an IP addresses and a port number
- Ports 0-1023 are reserved
  - Defined by IANA (Internet Assigned Numbers Authority)
  - E.g. port 80 is for HTTP server
  - E.g. port 25 is for mail server
- Ports 1024-65535 can be used by other applications

2-11 An application-layer protocol defines

- Types of messages exchanged
- Message syntax
- Message semantics
- Rules

2-12 What transport service does an app need

- Transport services that could be offered
  - Reliable data transfer
  - Throughput
  - Timing

    - Security
- Reliability
    - Contents are delivered in order
    - For some applications (e.g. web and email), there should be no packet loss
        - Dropped packets must be retransmitted
- Throughput
    - Impacted by bottleneck link and traffic
    - Encoding can vary depending on the current throughput
- Timing
    - Low delay
- Security
    - Encryption
    - Data integrity

2-14 Internet transport protocols services

- TCP service
    - Process-process connection
    - Reduces the rate of a sender when there is a congestion
- UDP service
    - Has less overhead than TCP since it does not need to control the connections between processes like TCP
    - Usually a backup for TCP

2-16 Securing TCP

- TLS works in the application layer

2-20 HTTP overview

- A TCP connection must be established before the client and the server can exchange messages

2-21 HTTP connections: two types

- The oldest version of HTTP only supports non-persistent HTTP
- Even in a persistent HTTP connection, the transfer is still stateless

2-24 Non-persistent HTTP: response time

- RTT: round trip time

Lecture 5

Thursday, May 25, 2023

2-94 Socket programming
- Reliable service: TCP
- Unreliable service: UDP

2-96 Socket programming with UDP
- Every message sent by the UDP client must have an identifier (IP + port)
- UDP does not have connections, whereas TCP does

2-98 Example app: UDP client
- serverAddress = implicit serverName + serverPort added by the server

2-99 example app: UDP server
- Continuously listening on port 12000 to receive messages from (different) clients

2-102 Example app: TCP client
- The socket created on the serer side is called a **welcome socket**
- Since there is already a connection, there is no need to attach the identifier of the receiver in every message

2-103 Example app: TCP server
- connectionSocket is not on the same port as the serverSocket (welcome socket_
- Server blocks when listening for client messages

2-26 HTTP request message
- There are wo types of HTTP messages
  - **Request**: sent by a client to obtain a specific object
  - **Response**: sent by a server to the client

2-27 HTTP request message: general format
- Method - what should the server do
  - GET

E.g. GET	/somedir/index.html	HTTP1.1
  - POST

E.g. POST	uwaterloo.ca
  - PUT

E.g. PUT	User-agent	Mozilla
  - HEAD

E.g. HEAD	Accept-languages: fr
- In a POST request, the input values are in the entity body
- In a GET request, the specific parameters are in the URL
  E.g. /somedir/index.html?f1=v1&f2=v2

2-29 HTTP response message

- Instead of a request line, there is a status line
-

Version	Code	Phrase
HTTP/1.1	200	OK

- Header field names examples:
    - Connection close
    - Date
    - Last modified

2-31 Trying out HTTP (client side) for yourself

- For a GET request, hit carriage return twice to specify an empty message body

Lecture 6

Tuesday, May 30, 2023

2-36 HTTP/2 to HTTP/3

- TCP tries to provide fair bandwidth for each client
- But a client can open up many connections
- Original HTTP
  - L5: HTTP2 + TLS
  - L4: TCP
  - L3: IP
- QUIC
  - L5: HTTP2 + QUIC (provides reliability)
    - This is HTTP3
  - L4: UDP
  - L3: IP
- In HTTP3, reliability is not provided by UDP, so it is implemented in the application layer

2-37 Maintaining user/server state: cookies

- HTTP is **stateless**
  - New transactions do not remember objects transferred in old transactions
- Websites use cookies to track users and make recommendations

2-38 Maintaining user/server state: cookies

- Cookie header file for adding a cookie to the client's host
  - set-cookie: xxx
- Other cookie headers
  - max-age
  - secure
- Once a cookie is set, subsequent client request will include a cookie header
  - cookie: xxx
- The webserver uses the cookie to access the database and respond with cookie-specific information

2-39 HTTP cookies: comment

- Cookies can be thought of as a layer above the application layer

2-40 Web caches

- Aka: proxy servers, edge servers
- Example: Netflix has CDN (content delivery networks) that work like proxies

2-41 Web caches (aka proxy servers)

- Both client and server
  - Server for original requesting client
  - Client to origin server
- Cache control
  - Server tells cache about object's allowable caching in response header

2-42 Caching example

- Traffic intensity = La/R = 1.50 Mbps / 1.54 Mbps = 0.97
    - Very close to 1
    - Large queueing delays at high utilization

2-43 Option 1: buy a faster access link
- Traffic intensity = La/R = 1.50 Mbps / 154 Mbps = 0.0097
    - Very close to 0
    - Almost no queueing delay
- Faster access link is expensive

2-44,2-45 Option 2: install a web cache
- Assuming 40% cache hit rate, which means 60% traffic goes to origin servers
- Traffic intensity = 0.97 * 60% = 0.58
    - Low (msec) queueing delay at access link

2-46 Conditional GET
- Do not send object if cache has up-to-date cached version
- If the object if not modified after a specific date, the response will be "304 Not Modified"

2-48 E-mail
- User agent
    - Compose, edit and read mail messages
- Email server
    - Stores outgoing and incoming emails

2-50 SMTP RFC
- Vanilla SMTP: body in SCII
- MIME: extension for SMTP

2-51 Scenario: Alice sends e-mail to Bob
- SMTP is a push protocol
- User agents cannot pull messages from mail server using SMTP
- Instead, they use SMAP (Simple Mail Access Protocol)

2-52 Sample SMTP interaction
- No authentication in vanilla SMTP means impersonation can happen

Lecture 7

Thursday, June 1, 2023

2-57 DNS: Domain Name System

- Routers - use IP addresses (32 bit, fixed)
- Humans - use names (e.g. www.uwaterloo.ca)
- In order to establish UDP/TCP connections, we need to use services provided by DNS (hostname to IP mapping)
- Two types of messages in DNS
    - Query
    - Reply (aka response)
- DNS works in application layer
    - Minimize delay
    - Does not guarantee reliability
- DNS relies on UDP (port 53)
    - May use TCP as a backup

2-58 DNS: services, structure

- Host aliasing
    - Alias name (e.g. uwaterloo.ca)
    - Canonical name (e.g. main-campus.uwaterloo.ca)
- Load distribution, aka load balance
    - Replica servers - many IP addresses correspond to one name
    - When querying using a specific name, DNS might return multiple IP addresses

2-62 DNS: root name servers

- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain
- Root server - last resort by names servers that cannot resolve name

2-63 Top-Leve Domain, and authoritative servers

- There can be intermediate servers between TLD (Top-Level Domain) and authoritative DNS servers
    - E.g. dns.cs.uwaterloo.ca
    - E.g. dns.ece.uwaterloo.ca

2-64 Local DNS name servers

- Goals of local name servers
    - Act on behalf of higher level servers
    - Provides caching
- DHCP (Dynamic Host Configuration Protocol) assigns each host that connects to the network a local DNS server

2-68 DNS records

- type=A
    - IP address for a given hostname
    - E.g. (uwaterloo.ca, x.x.x.x, A)
- type=NS
    - E.g. (uwaterloo.ca, dns.uwaterloo.ca, NS)

    - Store in TLD servers
- type=CNAME
    - E.g. (www.ibm.com, servereast.backup2.ibm.com, CNAME)

2-74 Peer-to-peer (P2P) architecture

- Previously mentioned architecture is called **client-server architecture**
- P2P architecture has self scalability
    - As the number of users increase, both demand and supply increase

2-75,2-76,2-77,2-78 File distsribution: client-server vs P2P

- Client-server architecture, file size = F bits
    - The server must send NF bits
    - Min distribution time = $\frac{NF}{u_s}$
    - Min download time = $\frac{F}{\min\{d_1,d_2,\ldots d_N\}}$
    - Total time needed = $\max\{\frac{\mathrm{NF}}{\mathrm{u_s}},\frac{\mathrm{F}}{\min\{\mathrm{d_1},\mathrm{d_2},\ldots \mathrm{d_N}\}}\}$
    - Distribution time grows linearly with respect to number of clients
- P2P architecture
    - The server must send one copy
    - Min distribution time = $\frac{F}{u_s}$
    - Min download time for one client = $\frac{F}{\min\{d_1,d_2,\ldots d_N\}}$
    - Clients as aggregate must download NF bits
    - Max download rate is $u_s + \sum u_i$
    - Min distribution time = $\max\{\frac{\mathrm{F}}{\mathrm{u_s}},\frac{\mathrm{F}}{\min\{\mathrm{d_1},\mathrm{d_2},\ldots \mathrm{d_N}\}},\frac{NF}{u_s+\sum u_i}\}$
    - Distribution time with respect to number of clients grows slower than client-server

Lecture 8

Tuesday, June 6, 2023

2-79,2-80 P2P file distribution: BitTorrent

- P2P architecture has no reliance on an always-on server
- Torrent corresponds to a distribution, or a swarm (a group of peers exchanging chunks of a file)
- Example: Alice wants to download a file
    - Alice joins Torrent
    - Alice registers with the tracker and obtains list of peers from trackers
    - Alice requests to establish TCP connections with peers, some of which may reject
    - As time progresses, other peers may request to establish TCP connections with Alice, so that Alice changes peers with who it exchanges chunks
    - Once Alice has the entire file, it may (selfishly) leave or (altruistically) remain in swarm

2-81 BitTorrent: requesting, sending file chunks

- Alice needs to make two decisions
    - Which chunks to request
    - Which requests to respond to
- Which chunks to request
    - Rarest first - if the peers with the rarest chunks leave, then no one in the swarm would be able to download the chunk
- Which request to respond to
    - Top 4 peers that are sending chunks to Alice at the highest rate
    - Updates top 4 every 10 seconds
    - Every 30 seconds, randomly select another peer and starts sending chunks
    - The 5th peer is called opportunistically unchoked peer
    - Why? Alice hopes that by randomly choosing an unchoked peer, that peer will start reciprocating and (might) eventually join the top 4.

2-82 BitTorrent: tit-for-tat

- If Bob only downloads chunks from peers but does not reciprocate, he is called a free rider

2-84 Video Streaming and CDNs: context

- Different users have different capabilities (wired vs wireless, bandwidth…)
- A user can experience fluctuations of bandwidths
- DASH: dynamic adaptive streaming over HTTP
- CDN: content distribution networks

2-85 Streaming stored video

- Videos are made of frames (images), which are consisted of pixels
- Videos can be compressed to different qualities
- The measure is bit rate
    - The higher the bit rate, the lower the compression and vice versa

2-86 Streaming multimedia: DASH

- Server
    - Divides video file into multiple chunks

    - Each chunk encoded at multiple different rates
    - Different rate encoding stored in different files
    - Files replicated in various CDN nodes
    - **Manifest files**: provides URLs for different chunks
- Client
    - Periodically estimates server-to-client bandwidth
    - Chooses maximum coding rate given current bandwidth
    - Can choose different coding rates at different points in time (depending on available bandwidth at time), and from different servers

2-87 Streaming multimedia: DASH

- Client determines
    - When to request chunk (to avoid starvation and overflow)
    - What encoding rate to request (higher quality when more bandwidth available)
    - Where to request chunk (from a CDN node that is closest or has highest bandwidth)

2-88 Content distribution networks (CDNs)

- Single, large "mega-server" does not scale
- We need a distributed infrastructure

2-89 Content distribution networks (CDNs)

- Where should we put the surrogate servers?
    - Enter deep - push CDN servers into many access networks, close to the network edge
    - Bring home - small number of larger clusters in POPs (point of presence) near access nets
- Example: NetCinema www.netcinema.com
    - Suppose it employs a 3rd party CDN called KingCDN
    - User wants to watch a video by clicking the video URL video.netcinema.com/ABC123
    - The fetch request is sent to NetCinema authoritative DNS server, which returns an address for KingCDN's authoritative DNS server
    - This is called **DNS redirection**
    - The KingCDN's authoritative DNS server then returns the IP addresses of the surrogate servers that are close to the client's local DNS server
- Some problems
    - Geographical closeness does not guarantee low packet congestion

2-91 Case study: Netflix

- Relies on Amazon cloud for recommendations, user registrations, content processing, etc.
- Netflix does not need DNS redirection to redirect to a 3rd party CDN since its CDNs are private

Chapter 2 Summary

Wednesday, June 7, 2023

Principles of network applications

Creating a network app

- There is no need to write software for network-core devices
- Network-core devices do not run user applications
- Applications on end systems allows for rapid app development, propagation

Client-server paradigm

- Server
  - Always on
  - Permanent IP address
  - Often in data centers, for scaling
- Client
  - Contact, communicate with server
  - May be intermittently connected
  - May have dynamic IP address
  - Do not communicate directly with each other

P2P architecture

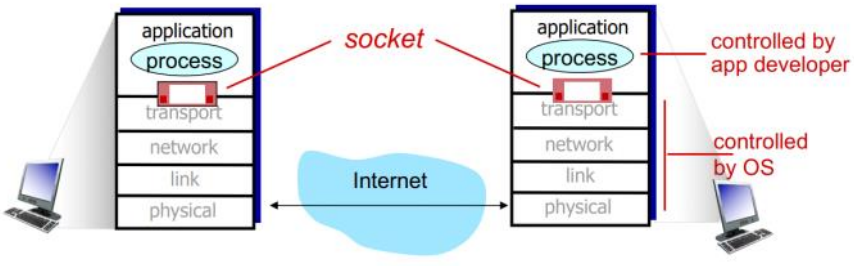
- No always-on server
- Peers request/provide service from/to each other
- Self-scalability: new peers bring new service demands and capacity
- Peers are intermittently connected and change IP addresses

Process communicating

- Processes in the same host communicate using IPC (inter-process communication)
- Processes in different hosts communicate by exchanging messages
- Client process - initiates communication
- Server process - waits for connection request

Sockets

- Process sends/receives message to/from its socket
- Two sockets involved: one on each side
- Relies on transport infrastructure to deliver message from one socket to the other



Addressing processes

- Many processes can be running on the same host
- Identifier includes
    - IP address
    - Port number

Application-layer protocol

- Types of messages exchanged
    - Request
    - Response
- Message syntax
    - Fields in message
- Message semantics
    - Meaning of fields
- Rules for when and how processes send and respond to messages

Transport service an app need

- Data integrity
- Timing
- Throughput
- Security

Internet transport protocols services

- TCP
    - Reliable transport
    - Flow control
    - Congestion control
    - Connection-oriented
    - No timing, throughput guarantee, security
- UDP
    - Unreliable
    - Does not provide anything…

Securing TCP

- Vanilla TCP & UDP sockets have no encryption
- Messages are transferred in cleartext
- Transport Layer Security (TLS)
    - Encrypted TCP connections
    - Data integrity
    - End-point authentication
- TLS works in the application layer

Web and HTTP

Web and HTTP

1. Web pages consists of objects
    - HTML file
    - JPEG image

    - Java applet
    - Audio file
    - …
- Web page consists of base HTML-file which includes several referenced objects
    - Each addressable by a URL
    - E.g. www.somewschool.edu/someDept/pic.gif

HTTP overview

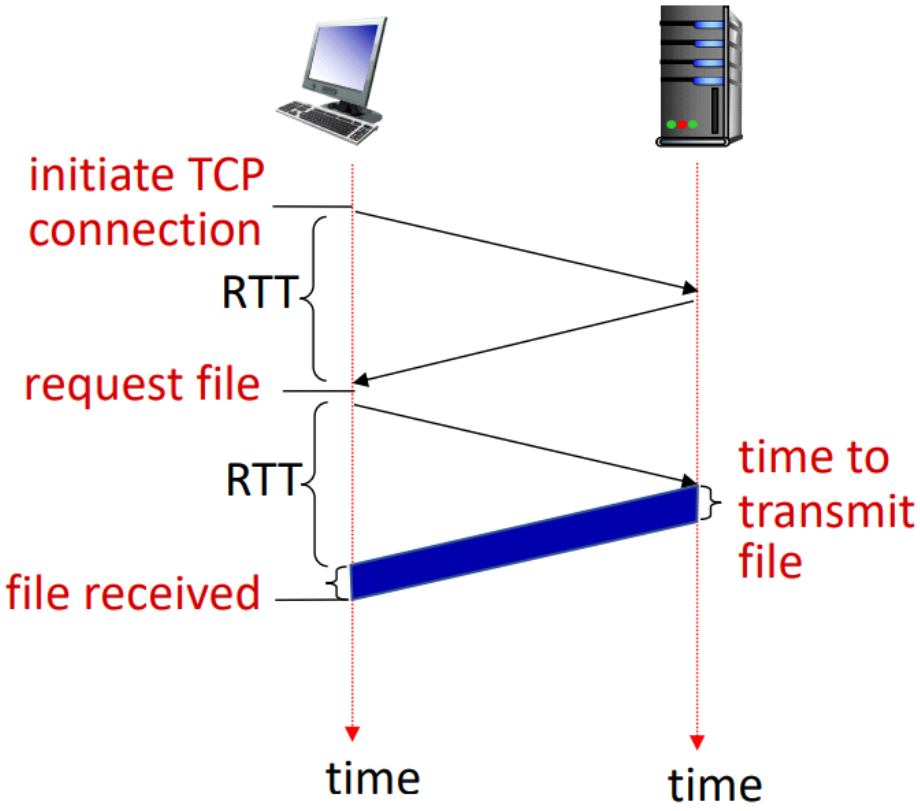
- Hypertext Transfer Protocol
- Client - browser that requests Web objects
- Server - web server that sends responses containing Web objects
- HTTP (uses TCP (transport layer)
- Stateless
    - Server maintains no information about past client requests

HTTP connections: two types

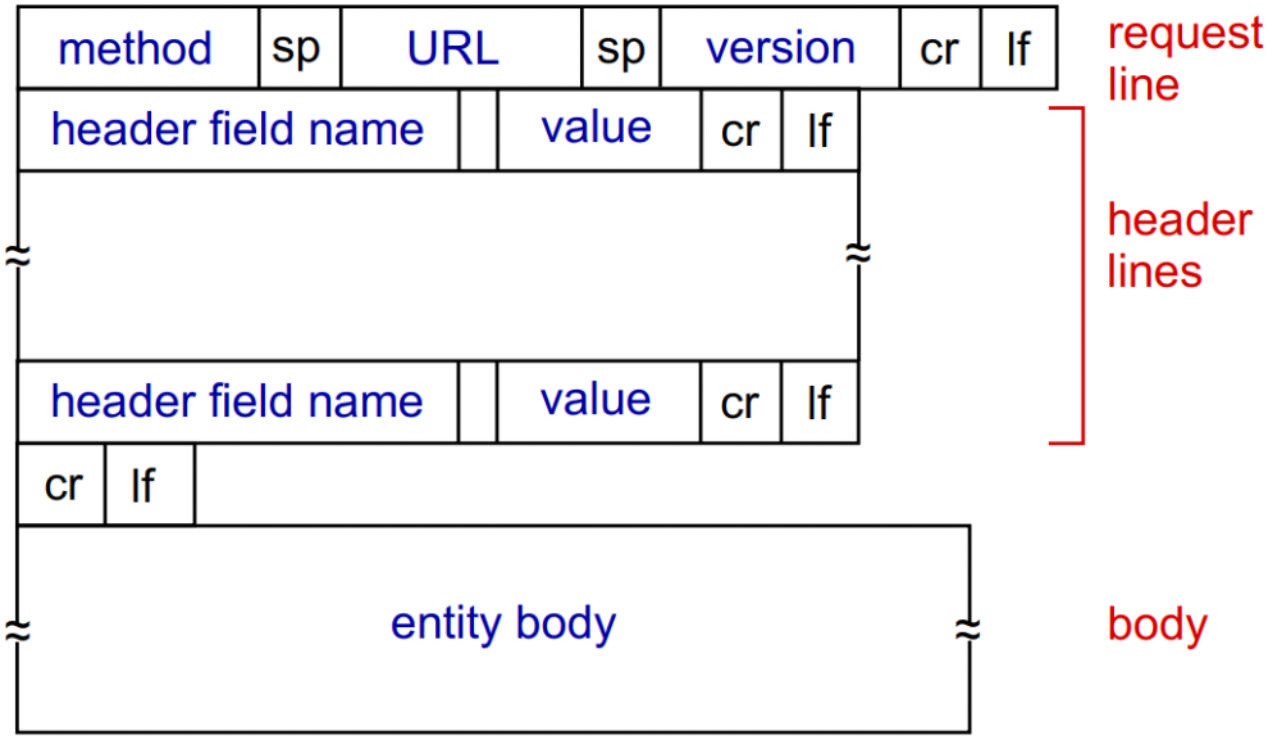
- Non-persistent HTTP
    1. TCP connection opened
    2. At most one object sent over TCP connection
    3. TCP connection closed
- Persistent HTTP
    1. TCP connection opened to a server
    2. Multiple objects can be sent over single TCP connection between client and server
    3. TCP connection closed

Response time

- RTT: time for a small packet to travel from client to server and back
- HTTP response time (per object):
    - One RTT to initiate TCP connection
    - One RTT for HTTP request and first few bytes of HTTP response
    - Object/file transmission time



HTTP request message general format



HTTP request methods

- POST - usually for form input
- GET - for retrieving data from server
- HEAD - for retrieving headers for URL that were requested by a GET method

- PUT - replace objects previously uploaded with a POST request

HTTP response message

- The 1st line is the status line
  E.g. HTTP/1.1 200 OK
- Common status
  - 200 OK
  - 301 moved Permanently
  - 400 Bad Request
  - 404 Not Found
  - 505 HTTP Version Not Supported

HTTP/2

- Goal: decrease delay in multi-object HTTP requests
- HTTP1.1 introduced multiple, pipelined GETS over single TCP connection
  - Server responds FCFS
  - With FCFS, small objects may have to wait
  - This is called **head-of-line blocking** (HOL)
- In HTTP/2
  - Transmission order is based on client-specified object priority
  - Objects are divided into frames to mitigate HOL blocking

HTTP/3

- HTTP/2 problems
  - Recovery form packet loss still stalls object transmission
  - No security
- HTTP/3 changes
  - Adds security
  - Adds per object error and congestion control
  - Uses UDP

Cookies

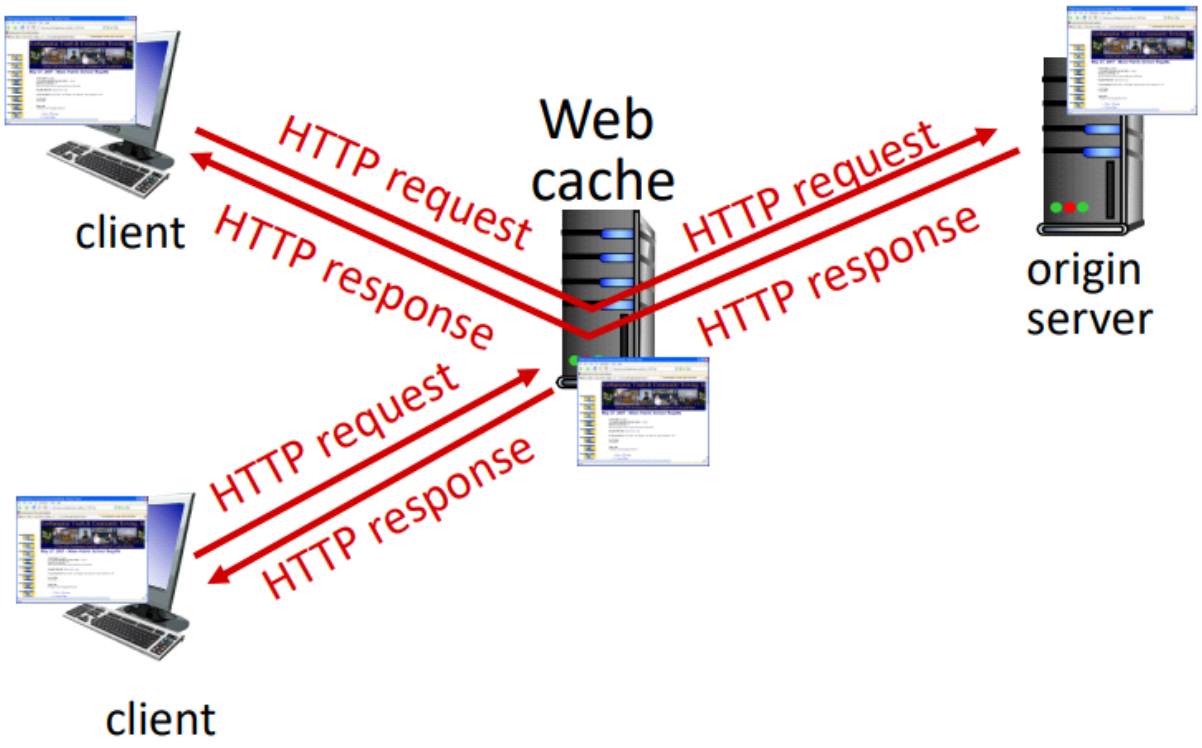
- Components
  - Cookie header line of HTTP response message
  - Cookie header line in next HTTP request message
  - Cookie file kept on users' host
  - Backend database record at Web site
- Applications
  - Authorization
  - Shopping carts
  - Recommendations
  - User session

Web caches

- Aka proxy servers
- Satisfy client requests without involving origin server
- If object in cache: cache returns object to client
- Else cache requests object from origin server, caches received object, then returns object

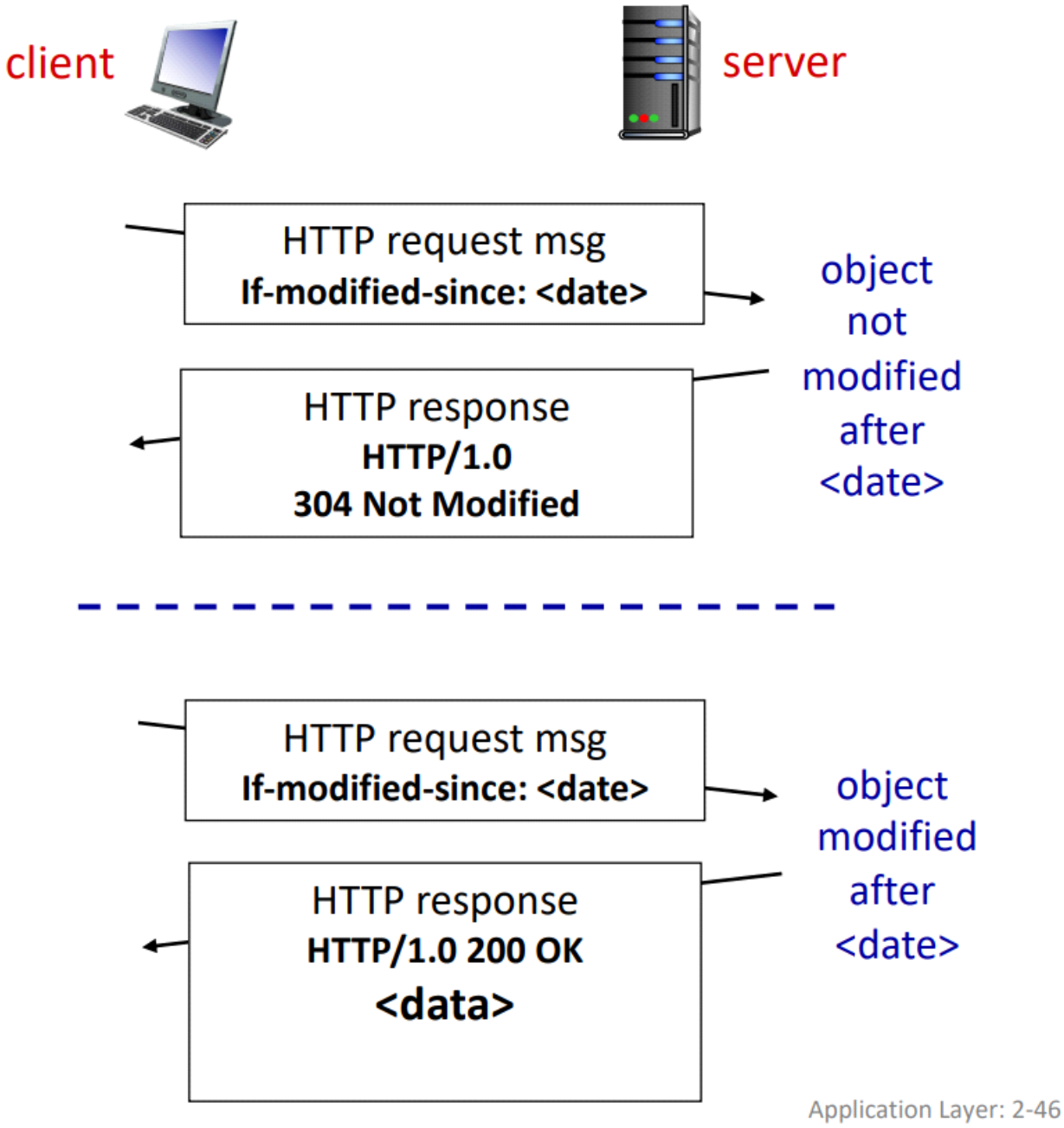
to client

- Why web caching?
  - Reduce response time for client request
  - Reduce traffic on an institution's access link
  - Internet is dense with caches



Conditional GET

- Goal: do not send object if cache has up-to-date cached version
  - No object transmission delay



Email, SMTP, IMAP

Email

- User agents
- Mail servers
- Simple Mail Transfer Protocol: SMTP

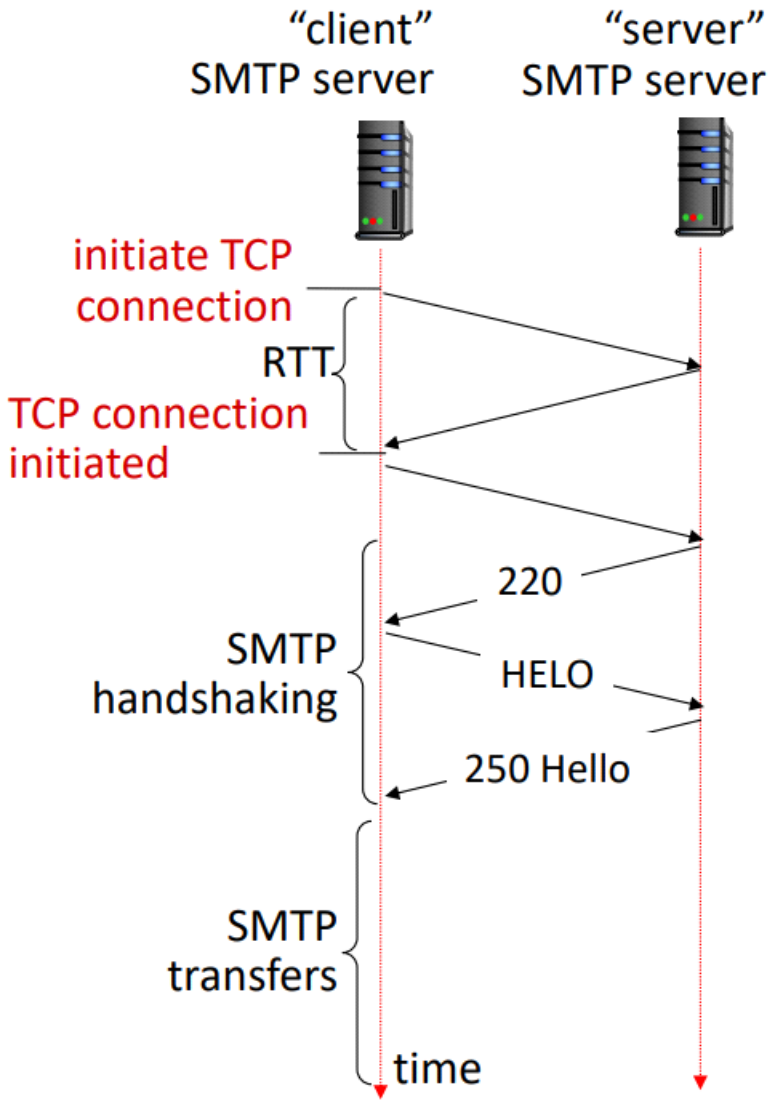
User agent

- Aka mail reader
- Composing, editing and reading mail messages

Mail servers

- Mailbox contains incoming messages

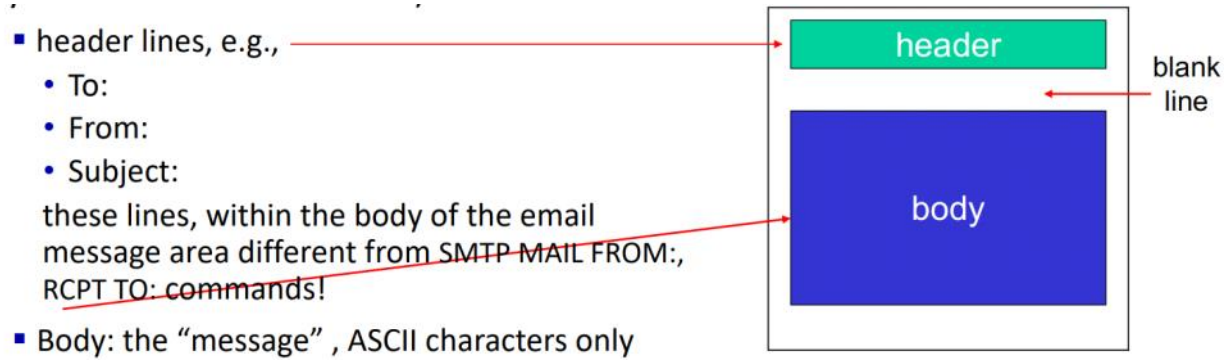
- Message queue of outgoing messages
- SMTP protocol between mail servers to send email messages



SMTP vs. HTTP

- Client
  - HTTP: client pull
  - SMTP: client push
- Objects
  - HTTP: each object encapsulated in its own response message
  - SMTP: multiple objects sent in multipart message
- Connections
  - HTTP: can be non-persistent
  - SMTP: persistent connections
- SMTP requires message to be in 7-bit ASCII
- SMTP server uses CRLF.CRLF to determine end of message

Mail message format



Domain Name System

Domain Name System (DNS)

- Distributed database implemented in the hierarchy of many name servers
- Application layer protocol
- Complexity at network's edges

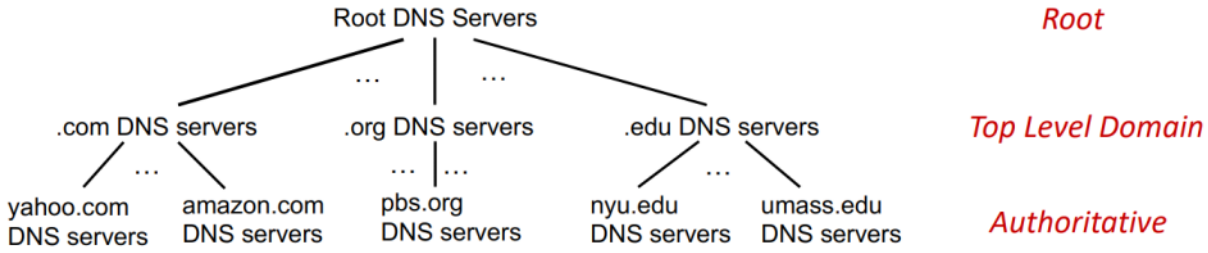
DNS services

- Hostname-to-IP-address translation
- Host aliasing
- Mail server aliasing
- Load distribution
    - Replica servers: many IP addresses correspond to one name

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

Distributed, hierarchical database



Types of DNS servers

- Root server
    - official, contact-of-last-resort
- Top-Level Domain servers
    - For .com, .org, .net, etc.
    - Also for top-level country domains
- Authoritative DNS servers
    - Organization's own DNS servers

    - Provide authoritative hostname to IP mappings for organization's named hosts
- Local DNS servers
    - Does not strictly belong to hierarchy

Caching DNS information

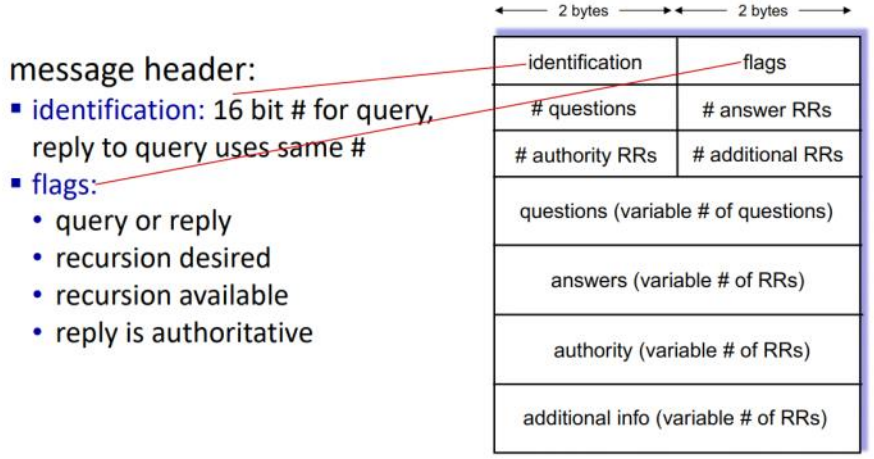
- Any name server caches a mapping once it learns mapping
- Improves response time
- TLD servers typically cached in local DNS servers

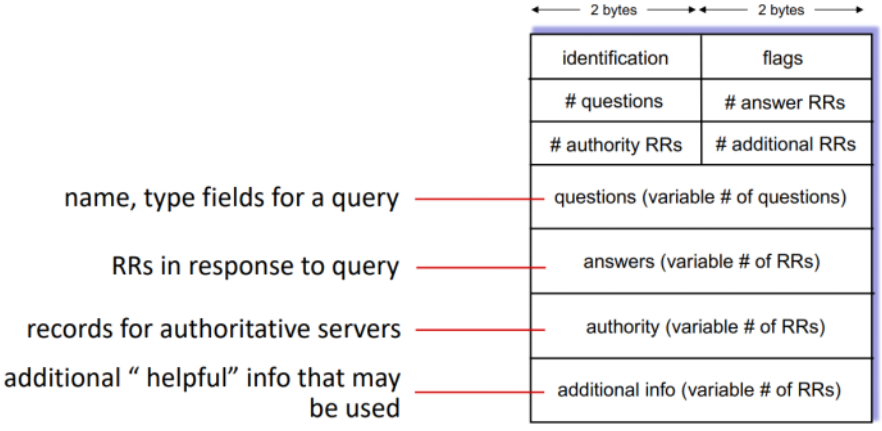
DNS records

- RR format: (name, value, type, ttl)
- Type=A: hostname to IP
- Type=NS: domain to hostname of DNS server for this domain
- Type=CNAME: alias name to canonical name
- Type=MX: domain to name of SMTP mail server

DNS protocol messages

- Query and reply messages have the same format





File distribution time

- Client-server architecture, file size = F bits
    - The server must send NF bits
    - Min distribution time = $\frac{NF}{u_s}$

    - Let $d_{min} = \frac{F}{\min\{d_1, d_2, \ldots d_N\}}$
    - Min download time = d_{min}
    - Total time needed $\geq \max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\}$
    - Distribution time grows linearly with respect to number of clients
- P2P architecture, file size = F bits
    - The server must send one copy
    - Min distribution time = $\frac{F}{u_s}$
    - Min download time for one client = d_{min}
    - Clients as aggregate must download NF bits
    - Max download rate is $u_s + \sum u_i$
    - Total time needed $\geq \max\{\frac{F}{u_s}, \frac{F}{\min\{d_1, d_2, \ldots d_N\}}, \frac{NF}{u_s + \sum u_i}\}$
    - Distribution time with respect to number of clients grows slower than client-server

BitTorrent

- File divided into 256kB chunks
- **Tracker**: tracks peers participating in swarm
- **Swarm**: group of peers exchanging chunks of a file
- **Churn**: peers may come and go
- Requesting chunks
    - Periodically asks each peer for list of chunks that they have
    - Requests missing chunks from peers, rarest first
- Sending chunks: tit-for-tat
    - Sends chunks to top four peers currently sending her chunks at the highest rate
    - Reevaluate top 4 every 10 seconds
    - Every 30 seconds, randomly select another peer and starts sending chunks

Video Streaming and CDNs

Domain Name System (DNS)

- Distributed database implemented in the hiearchy of many name servers
- Application layer protocol
- Complexity at network's edges

DASH

- Dynamic, Adaptive Streaming over HTTP
- Server
    - Divides video file into multiple chunks
    - Each chunk encoded at multiple different rates, stored in different files
    - File replicated in various CDN nodes
    - **Manifest file** provide URLs for different chunks
- Client determines
    - When to request chunk
    - What encoding rate to qequest
    - Where to request chunk

Chapter 2 Review Questions

Monday, June 12, 2023

SECTION 1.1

R1. List five nonproprietary Internet applications and the application-layer protocols that they use.

Web browser - HTTP
Email - SMTP
Domain Name System - DNS
File transfer - FTP
Multimedia Streaming - RTP

R2. What is the difference between network architecture and application architecture?

From the application developer's perspective, the network architecture is fixed and provides a specific set of services to applications.
The application architecture, on the other hand, is designed by the application developer and dictates how the application is structured over the various end systems.

R3. For a communication session between a pair of processes, which process is the client and which is the server?

The client is the process initiating the connection.
The server is the process listening for the other process's requests.

R4. Why are the terms client and server still used in peer-to-peer applications?

We can use the term "client" to describe a peer that is receiving chunks from other peers, and the term "server" to describe a peer that is sending chunks to other peers.

There is no always-on server.

R5. What information is used by a process running on one host to identify a process running on another host?

IP address and port number

R6. What is the role of HTTP in a network application? What other components are needed to complete a Web application?

HTTP is a protocol for web browsers in the application layer. It specifies the formats of the messages exchanged between web browsers and servers.

R7. Referring to Figure 2.4, we see that none of the applications listed in Figure 2.4 requires both no data loss and timing. Can you conceive of an application that requires no data loss and that is

also highly time-sensitive?

Safety-related applications? Air navigation systems?

R8. List the four broad classes of services that a transport protocol can provide. For each of the service classes, indicate if either UDP or TCP (or both) provides such a service.

Reliable data transfer - TCP
Timing - neither
Throughput - neither
Security - neither

R9. Recall that TCP can be enhanced with TLS to provide process-to-process security services, including encryption. Does TLS operate at the transport layer or the application layer? If the application developer wants TCP to be enhanced with TLS, what does the developer have to do?

TLS operates in the application layer.

SECTIONS 2.2–2.5

R10. What is meant by a handshaking protocol?

In a handshaking protocol, the client needs to initiate a connection by sending a request to the server, like a "greetings"

A protocol uses handshaking if the two communicating entities first exchange control packets before sending data to each other.

SMTP uses handshaking at the application layer whereas HTTP does not.

R11. What does a stateless protocol mean? Is IMAP stateless? What about SMTP?

A stateless protocol means previous transactions are not recorded / stored.

R12. How can websites keep track of users? Do they always need to use cookies?

Websites keep track of users using cookies.
They always need to use cookies because HTTP is stateless.

R13. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Why?

Web caching reduces the delay if the requested object is in the cache. Proxies can therefore return the cached object back to the client without going to the origin server.
Web caching reduces the delay for only some of the objects because it is not guaranteed that all objects are cached. There will be a portion of requests going to the origin server.

R14. Telnet into a Web server and send a multiline request message. Include in the request message the If-modified-since: header line to force a response message with the 304 Not Modified status code.

R15. Are there any constraints on the format of the HTTP body? What about the email message body sent with SMTP? How can arbitrary data be transmitted over SMTP?

HTTP message body is ?

SMTP message body is in 7-bit ASCII, so to transfer arbitrary data, we need to rely on other protocols/extensions?

SECTION 2.5
R16. Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server using IMAP. Discuss how the message gets from Alice's host to Bob's host. Be sure to list the series of application-layer protocols that are used to move the message between the two hosts.

Alice uses a user agent to compose her message and sends it to the mail server using HTTP. Then the mail server uses SMTP to send the message to Bob's mail server. Then Bob's mail server transfers the message from his mail server to his host over POP3

R17. Print out the header of an e-mail message you have recently received. How many Received: header lines are there? Analyze each of the header lines in the message.

R18. What is the HOL blocking issue in HTTP/1.1? How does HTTP/2 attempt to solve it?

In HTTP/1.1, the server uses FCFS to transfer multiple objects to the client, which means if there the first object is very large, there will be high latencies for the subsequent objects.

In HTTP/2, the objects are divided into smaller chunks, and server sends clients based on client-specified priorities, so that only the larger objects experience high latencies.

R19. Why are MX records needed? Would it not be enough to use a CNAME record? (Assume the email client looks up email addresses through a Type A query and that the target host only runs an email server.)

R20. What is the difference between recursive and iterative DNS queries?

Recursive: local -> root -> TLD -> authoritative -> TLD -> root -> local
Iterative: local -> root -> local -> TLD -> local ->

Recursive DNS queries put heavier burden on the root servers?

SECTION 2.5

R21. Under what circumstances is file downloading through P2P much faster than through a centralized client-server approach? Justify your answer using Equation 2.2.

R22. Consider a new peer Alice that joins BitTorrent without possessing any chunks. Without any chunks, she cannot become a top-four uploader for any of the other peers, since she has nothing to upload. How then will Alice get her first chunk?

Each peer optimistically unchokes a peer every 30 seconds, so if a few peers unchoked Alice by chance, Alice will be able to evaluate her top 4 peers and start reciprocating. After a few rounds, Alice can become a top 4 uploader for other peers.

Alice gets her first chunk if a peer optimistically unchokes her.

R23. Assume a BitTorrent tracker suddenly becomes unavailable. What are its consequences? Can files still be downloaded?

Newly joined users cannot register themselves or obtain a list of peers to obtain chunks.

SECTION 2.6

R24. CDNs typically adopt one of two different server placement philosophies. Name and briefly describe them.

Enter deep - put servers in the access network
Bring home - POP servers near network edge

Enter deep

- deploy server clusters in access ISPs all over the world
- The goal is to reduce delays and increase throughput between end users and the CDN servers

Bring home

- Bring the ISPs home by building large CDN server clusters at a small number of sites and placing these server clusters in IXPs
- Typically results in lower maintenance and management cost

R25. Besides network-related considerations such as delay, loss, and bandwidth performance, there are other important factors that go into designing a CDN server selection strategy. What are they?

Load-balancing - clients should not be directed to overload clusters
Diurnal effects
Variations across DNS servers
Limited availability of rarely accessed video
Alleviate hot-spots that may arise due to popular video content

SECTION 2.7

R26. In Section 2.7, the UDP server described needed only one socket, whereas the TCP server needed two sockets. Why? If the TCP server were to support n simultaneous connections, each from a different client host, how many sockets would the TCP server need?

In UDP, a connection does not need to be established before the client and the server can exchange messages.

In TCP, a connection needs to be established before the client and the server can exchange messages. The socket used to establish connections is called the "welcoming socket", and the socket for transactions is called a "connection socket".

TCP server needs n + 1 sockets to support n simultaneous connections.

R27. For the client-server application over TCP described in Section 2.7, why must the server program be executed before the client program? For the client-server application over UDP, why may the client program be
executed before the server program?

In TCP, the server needs to listen for requests to establish a connection from the client, so the server needs to start first.

In UDP, the server does not need to listen for requests to establish a connection from the client, since the destination address and port number are included in every message. Therefore, client needs to start first to send the server messages

Chapter 2 Problems

Monday, June 12, 2023

P1.

a. False
b. True
c. False
d. False
e. False

P3.

The local DNS server queries the root server for the address of .com TLD server, then queries .com TLD server for the address of yourbusiness.com authoritative DNS server, then sends a HTTP GET request to yourbusiness.com/about.html, which responds with about.html

Application layer protocols: DNS and HTTP
Transport layer protocols: UDP for DNS and TCP for HTTP

P4.

a. https://gaia.cs.umass.edu/cs453/index.html
b. HTTP/1.1
c. Persistent
d. ???
   This information is not contained in an HTTP message. One would need information from the IP datagrams.
e. Mozilla/5.0; so that the server know which version of webpage to include to include (ensures compatibility)

P5.

a. Yes; Tue, 07 Mar 2008
b. Sat, 10 Dec 2005
c. 3874
d. ?; Yes
   <!doc

P25.

a. Yes, that is possible. Because other peers can optimistically unchoke Bob and sends him file chunks
b. By using a collection of multiple computers, the expected time needed for receiving the entire can be reduced, since all of his computers will be periodically unchoked by other peers.

P26.

a. N
b. $2N$

P27.

a. TCPClient will fail to establish a connection
b. UDPClient will send the message successfully
c. It does not matter for UDP but TCP will fail

P28.

It is not necessary to change UDPServer.py.
In UDPClient, the port number for the socket is 5432.
In UDPServer, the port number for the socket is 12000.

P29.

I can open multiple simultaneous connections to a website by opening up multiple tabs.
One advantage is that it can reduce the time it takes to get the response containing objects from the web server.
One disadvantage is that it might cause network congestion.

P31.

Netflix uses Amazon cloud to upload copies of multiple versions of video to CDN servers. Amazon cloud is also used for storing manifest files and returning them upon user requests. Users can therefore use manifest files and DASH to retrieve the video from CDN servers.

Lecture 9

Tuesday, May 16, 2023

Midterm

- Friday, June 16
- Coverage: Chapter 1 & 2, Chapter 3 (first four topics)

3-4

- The network layer provides a host-host communication service
  - It can be thought of as "a link"
  - The physical link is not provided by the network layer, but instead provided by layers below
  - However the communication is unreliable
  - Reliability is provided by protocols in the transport layer (i.e. TCP) that is above the network layer
- Transport layer is also responsible for **encapsulation** of message from the application layer by adding headers, creating a **segment**

H	message

- **Multiplexing (mux)** - multiple applications from the application layer send messages to the transport layer, so the transport layer needs to process (i.e. encapsulates) them and send them to the network layer
- **Demultiplexing (demux)** - the transport layer processes (i.e. decapsulates) datagrams from the network layer to create segments and send them to the corresponding applications in the application layer.
- IP is the primary protocol in the network layer

3-9

- Transmission Control Protocol (TCP)
  - Reliable, in-order delivery - recovery from packet loss
  - Congestion control - throttle senders during congestion
  - Flow control - does not overflow receivers' buffer
  - Connection setup - a connection must be established before sending messages
  - Larger overhead than UDP
- User Datagram Protocol (UDP)
  - Unreliable, unordered delivery
  - Provides multiplexing and demultiplexing service for the network layer
  - Smaller overhead than TCP
- Neither offers delay guarantees and timing guarantees

3-12

- Each packet contains a header and a message from the layers above, regardless of the protocol being used
- In a datagram, there is a source IP address and a destination IP address in the header
- In a segment, there is a source port number and a destination port number in the header
- Hosts use IP addresses and port numbers as identifies

3-19

- Bare bone, best effort transport protocol on top of the network layer
- Applications that need throughput but tolerant to packet loss might use UDP
- TCP has higher delay due to connection setup and maintaining state
- UDP header is small - 8 bytes
- TCP header is larger - 20 bytes

3-20

- UDP use
    - streaming multimedia apps
    - DNS
    - SNMP
    - HTTP/3 - adds reliable transfer

3-22

- Length in UDP segment header = max # of bytes in the entire segment (including the header itself)
- Checksum
    - Check for bit errors
    - The sending side breaks the segment into 2-byte words
    - The words are summed (one's complement)
    - The checksum value is put into the checksum field
    - The receiver side computes checksum in the same way and check whether they are equal
    - Does not check for all possible errors

Lecture 10

Tuesday, June 13, 2023

3-27

- Reliable data transfer is not limited to the transport layer

3-28

- Reliable service
  - No bit error
  - Packets are sent in order
- Reliable service implementation
  - Sender-side of reliable data transfer protocol
  - Receiver -side of reliable data transfer protocol
- Reliable service abstraction
  - The sender and the receiver are oblivious of the implementation

3-32

- Unidirectional channel
  - The flow of data is only from the sending process to the receiving process
  - The receiving process can still send control messages to the sending side

3-34

- Assuming that the underlying channel is perfectly reliable, the sender and receiver can both have a simple implementation

3-35

- Feedback from receiver
  - Acknowledgement (OK)
  - Negative acknowledgement (repeat)
- Automatic Repeat Request (ARQ)

3-36

- Stop and wait - no packets can be received by the sender

3-40

- Problem: control messages are sent over the same unreliable channel
- The ACK or NAK messages can be corrupted as well
- Solution: sequence number [0,1]

3-42

- Twice as many states

3-58

- Using k-bit sequence number, the range of sequence numbers will be $[0, 2^k - 1]$
- Window size: how many packet can be transferred simultaneously without having been acknowledged?
- If only the acknowledge for the last bit is received, then the sender can be sure that the previous bits have been transmitted

Lecture 11

Thursday, June 15, 2023

Midterm

- T/F
- Brief short answer questions

3-66

- Solution to the problem: window size is less than equal to half of the size of sequence number range

3-68

- Point-to-point
  - Connection between two processes
- Has a send buffer and a receive buffer
- Treats messages as bytes teams
- Periodically fetch bytes from the send buffer
- Maximum segment size (MSS): maximum number of bytes TCP can fetch from the send buffer
- Maximum transmission unit (MTU): maximum size of a data-link frame
- MSS = MTU - IP header size (20 bytes) - TCP header size (20 bytes)
- Not all links have the same MTU
  - Fragmentation can happen
  - IPV6 does not allow fragmentation
  - Fragmentation is not desired
- Path MTU discovery

3-71

- Example: F = 500000 bytes, MSS = 1000 bytes
  - Segment 1: bytes 0-999, seq # = 0
  - Segment 2: bytes 1000-1999, seq # = 1000
  - …

3-72

- RTT changes over time since it depends on the congestion level in the network
- To estimate the RTT, use a sample

3-73

- Use an exponential weighted moving average
- Give significantly more weights to the recent RTTs

3-74

- Large variation in estimated RTT => use a larger safety margin

Lecture 12

Tuesday, June 20, 2023

3-74

- TCP uses a single timer for the oldest packet that is not ACKed
- On timeout, TCP retransmits the packet
- Sample RTT values exclude retransmitted packets

3-75

- Data from above
    - Create segment with seq #
        - Segment size = MSS
        - Sequence # is chosen randomly
- Timeout
- ACK from below

3-76

- In-order
    - Delayed ACK
    - Wait for more packets to give a cumulative ACK
    - Typically wait for 500 ms
- Out-of-order
    - Buffers
    - ACK the last in-order packet

3-77

- Lost ACK scenario
    - Host B buffers the first time and discards the packet the second time.
- Premature timeout
    - In the third transaction, Host B ACK the last in-order packet

3-78

- Doubling timeout
- Fast retransmit

3-79

- If sender receives 3 additional ACKs for same data, resend unACKed segment with smallest sequence
    - Why? It is likely that unACKed segment lost, so don't wait for timeout

3-85

- Network layer delivers data faster than application layer removes data from socket buffer
- Flow control - sender does not overwhelm receiver
    - If there is no flow control then the receiver might have buffer overflow
- LastByteRcvd - LastByteRead <= RcvBuffer
- rwnd = RcvBuffer - (LastByteRevd - LastByteRead)
- # of in flight and unACKed bytes <= rwnd

3-86

- rwnd = 0 means that the RevBuffer is full
    - If the sender does not send any more packets to the receiver, the sender will block
    - Instead, the sender sends 1 byte of data to the receive

3-87

- The host initiating the connection is referred to as the client
- The host waiting for the request to establish a connection is referred to as a server

3-88

- When the client requests for a connection, the first segment has SYN flag = 1
- This flag is similar to the ACK flag
- The seq # is randomly chosen
- DOS attack can be launched by sending SYN packets, creating half-open connections and consuming TCB resources
    - Solution: set up TCB later
    - Use a SYN cookie, sent in the second step in 3-way handshake

C	S
SYNbit =1, seq = x	
	ACKbit=1, ACKnum=x+1
	SYNbit=1, seq=y
ACKbit=1, ACKnum=y+1	

Step 2 and 3 are actually combined:

C	S
SYNbit =1, seq = x	
	ACKbit=1, ACKnum=x+1 SYNbit=1, seq=y Setup TCB (Transmission Control Block)
ACKbit=1, ACKnum=y+1 Setup TCB	

Lecture 13

Tuesday, June 27, 2023

3-107

- Assume two connections have the same MSS and RTT, and both are in the CA (congestion avoidance) state

3-110

- HTTP3 QUIC uses UDP as the underlying transport layer protocols

Chapter 3.1 to 3.4

Friday, June 23, 2023

Transport Layer Services

- Provides for logical communication between application processes running on different hosts
    - From the application's perspective, it is as if the hosts running the processes were directly connected
- Converts application layer messages it receives into transport-layer packets, known as segments
    - By breaking the application messages into smaller chunks and adding a transport-layer header to each chunk
- Passes the segment to the network layer, where the segment is encapsulated within a network-layer packet (datagram) and sent to the destination
- On the receiving side, the network layer extracts the segment from the datagram and passes it up to the transport layer
- The transport layer then processes the segment and makes it available to the receiving application layer

Overview of the Transport Layer

- UDP - unreliable, connectionless service
- TCP - reliable, connection-oriented service
- The network-layer IP provides logical communication between hosts
    - Best-effort delivery service
    - Makes no guarantees for segment delivery, orderly delivery and integrity of the data in the segments

Multiplexing and Demultiplexing

- Demultiplexing - delivering the data in a transport-layer segment to the correct socket
- Multiplexing - gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information and passing the segments to the network layer
- Two special fields
    - Source port number field
    - Destination port number field
- This is basically how UDP implements it
- TCP implementation is a bit different

Connectionless Multiplexing and Demultiplexing

- A UDP socket is fully identified by a two-tuple of a destination IP address and a destination port number
- If two UDP segments have different source IP addresses and/or source port numbers, but have the same destination IP address and destination port number, then they will be directed to the same socket

Connection-oriented Multiplexing and Demultiplexing

- A TCP socket is identified by a four-tuple (source IP address, source port number, destination IP address, destination port number)

- The server host may support many simultaneous TCP connection sockets, with each socket attached to a process, and with each socket identified by its own four-tuple

Connectionless Transport: UDP

- Aside from the multiplexing/demultiplexing function and some light error checking, UDP adds nothing to IP
- **Connectionless** - there is no handshaking between sending and receiving transport-layer entities
- **DNS** - example of application-layer protocol that uses UDP
- Applications are better suited for UDP because of
    - Finer application-level control over what data is sent, and when; no congestion control
    - No connection establish
    - No connection state
    - Small packet header overhead
- Potential problems
    - Cause packet overflow at routers
    - Cause TCP senders to dramatically decrease their rates

UDP Segment Structure

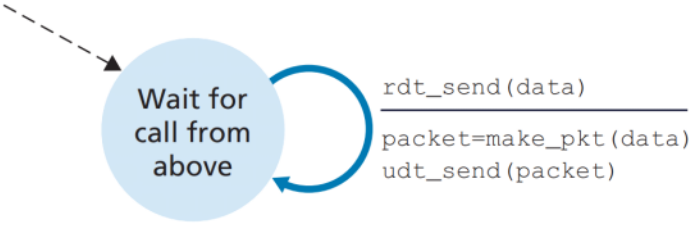
- Four fields, each consisting of two bytes
- Port numbers
- Length - number of bytes (header included)
- Checksum - for error detection

UDP Checksum

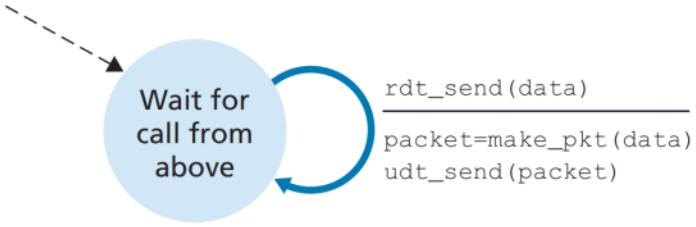
- The sender performs the 1s complement of the sum of all the 16-bit words in the segment, with any overflow wrapped around
- The receiver adds up all 16-bit words and the checksum to see if the result is all 1's
- Many link-layer protocols also provide error checking
- However there is no guarantee that all the links between source and destination provide error checking
- **End-end principle**: certain functions must be implemented on an end-end basis
- UDP provides error checking but does not recover from an error

rdt1.0

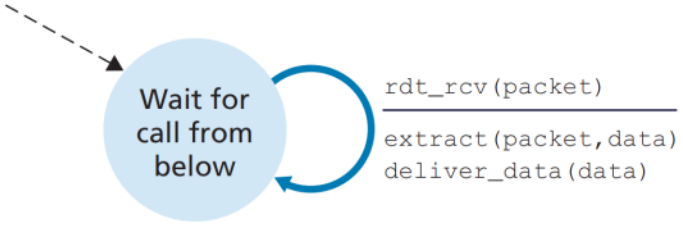
- Assume the underlying channel is completely reliable



a. rdt1.0: sending side



a. rdt1.0: sending side

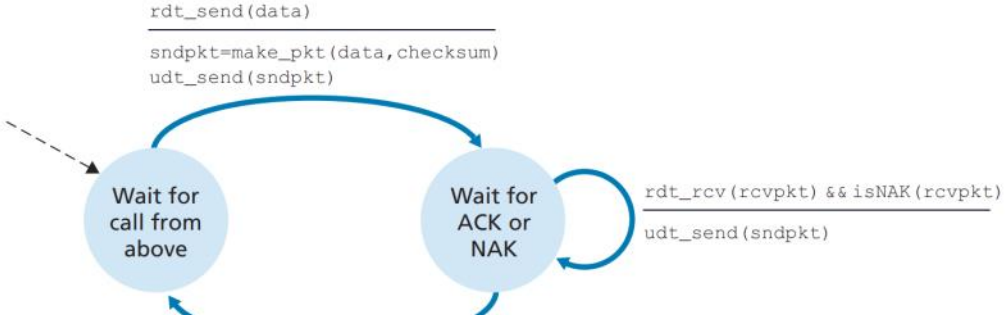


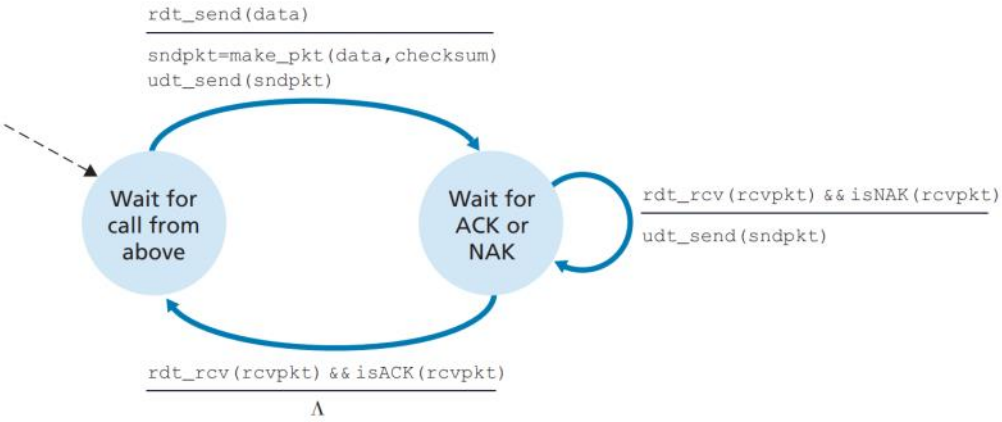
b. rdt1.0: receiving side

Figure 3.9 ♦ rdt1.0—A protocol for a completely reliable channel

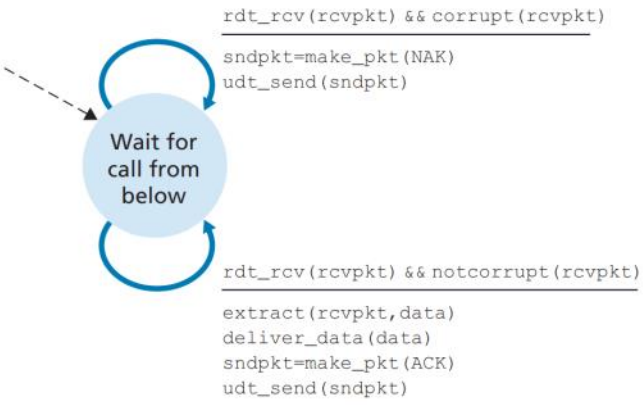
rdt2.0

- Bits in a packet may be corrupted
- The new protocols need positive acknowledgments and negative acknowledgments
- In a computer network setting, reliable data transfer protocols are based on such retransmission are known as ARQ protocols
- New capabilities
  - Error detection - detect bit errors by adding extra bits
  - Receiver feedback - ACK and NAK
  - Retransmission
- Known as a stop-and-wait protocol
- Problem
  - ACK or NAK packet could be corrupted
  - The receiver does not know whether the ACK or NAK it last sent was received correctly at the sender, so it cannot know whether an arriving packet contains new data or is a transmission
- Solution - 1 bit sequence number





a. rdt2.0: sending side



b. rdt2.0: receiving side

Figure 3.10 ♦ rdt2.0—A protocol for a channel with bit errors

rdt2.1

- Both sender and receiver FSMs now have twice as many states as before
- Since we are currently assuming a channel that does not lose packets, ACK and NAK packets do not themselves need to indicate the sequence number of the packet they are acknowledging
- The sender knows that a received ACK or NAK packet was generated in response to its most recently transmitted data packet
- When an out-of-order packet is received, the receiver sends an ACK for this packet

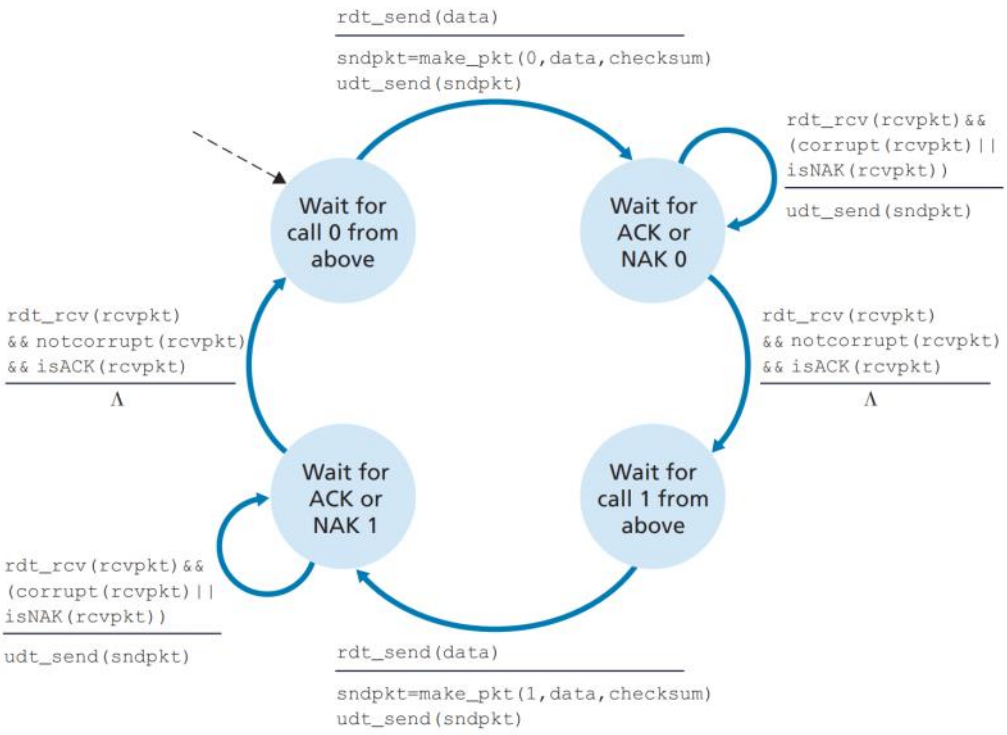


Figure 3.11 ♦ rdt2.1 sender

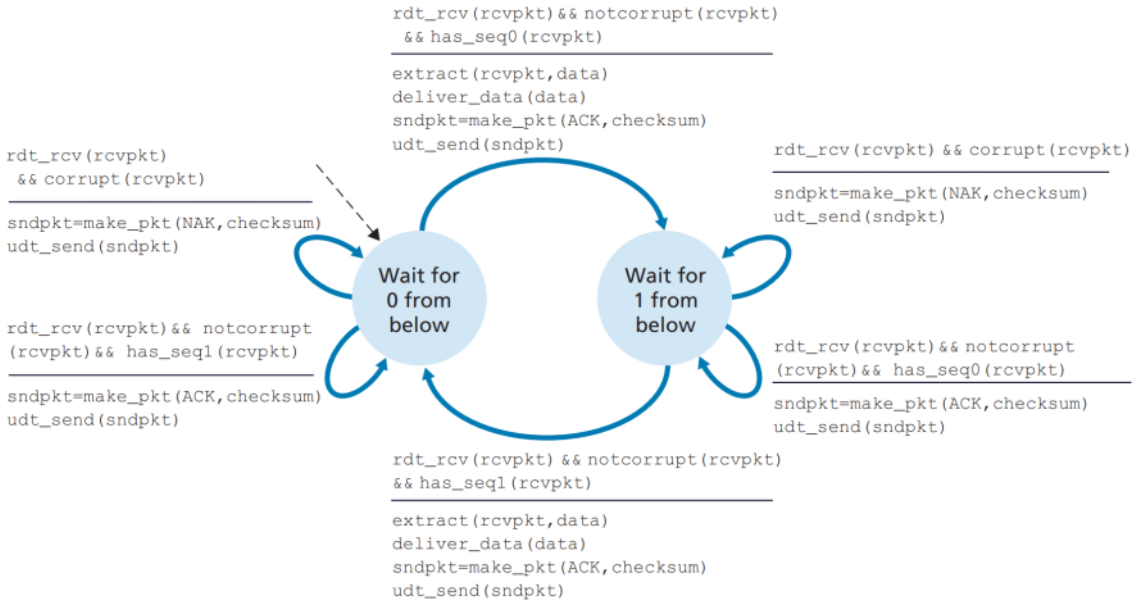


Figure 3.12 ♦ rdt2.1 receiver

rdt2.2

- The receiver includes the sequence number of the packet being acknowledged by an ACK message (ACK 0 or ACK 1)
- The sender checks the sequence number of the packet being acknowledged by a received ACK message

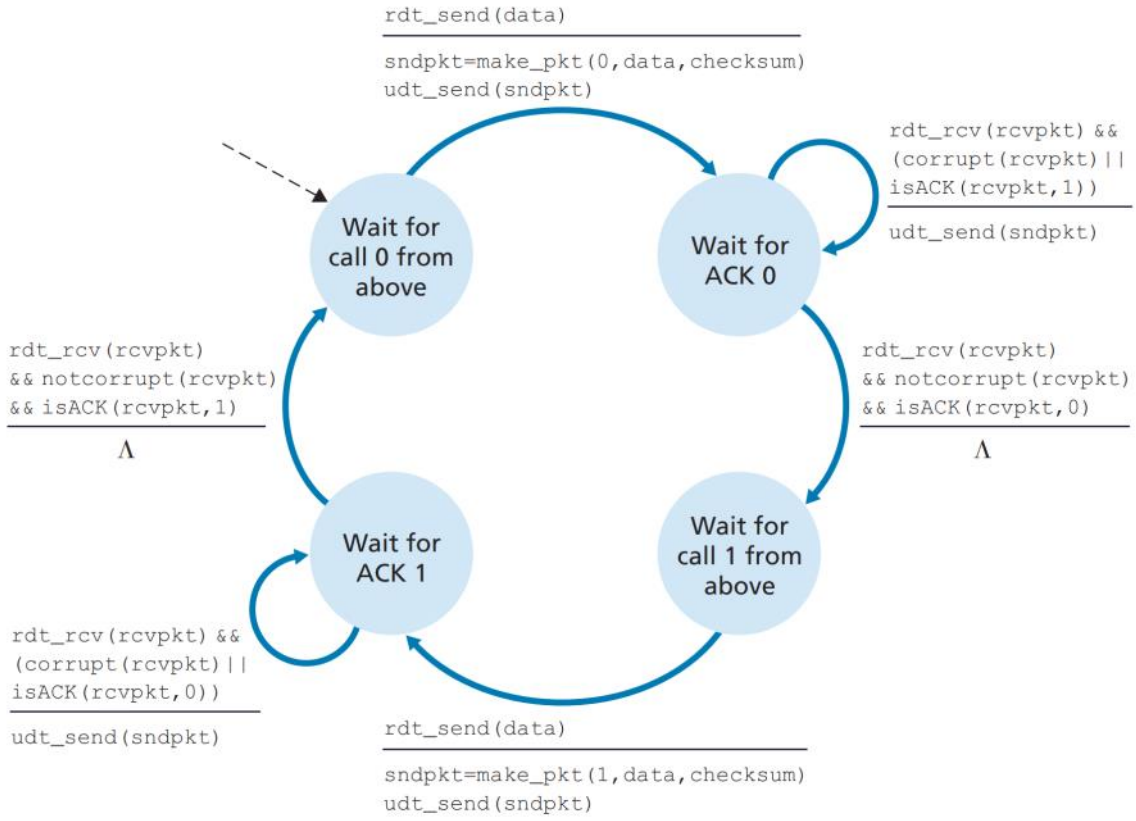


Figure 3.13 ♦ rdt2.2 sender

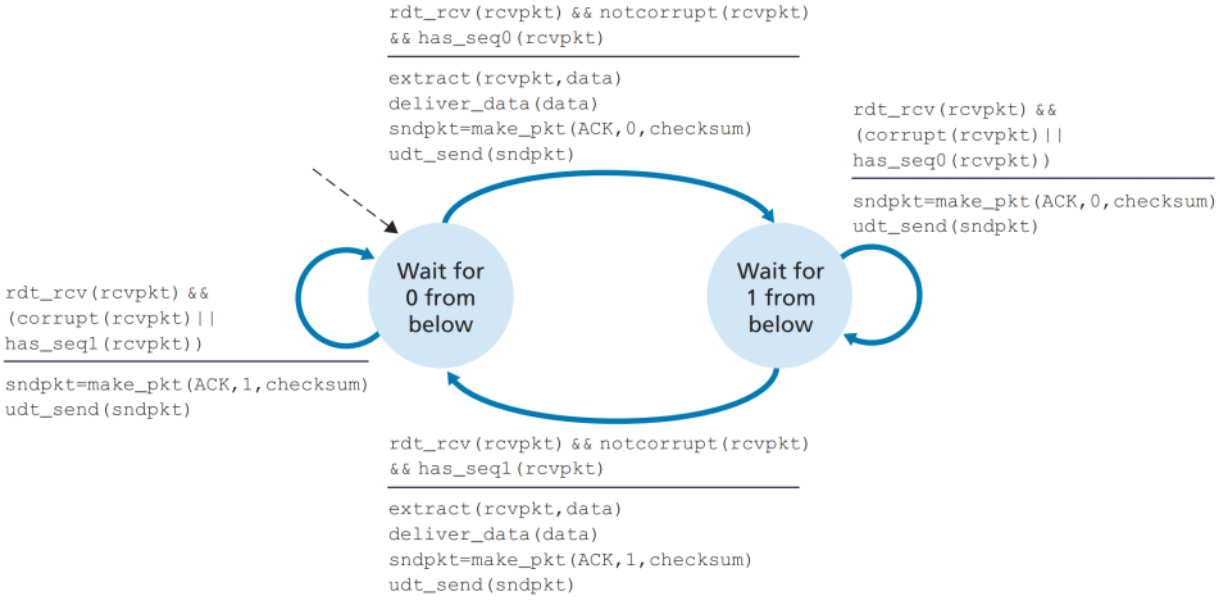


Figure 3.14 ♦ rdt2.2 receiver

rdt3.0

- Suppose now that in addition to corrupting bits, the underlying channel can lose packets as well
- The sender chooses a time value such that packet loss is likely, although not guaranteed, to have happened.

- If an ACK is not received, the packet is retransmitted
- Known as the **alternating-bit protocol**
- The receiver is the same as rdt2.2

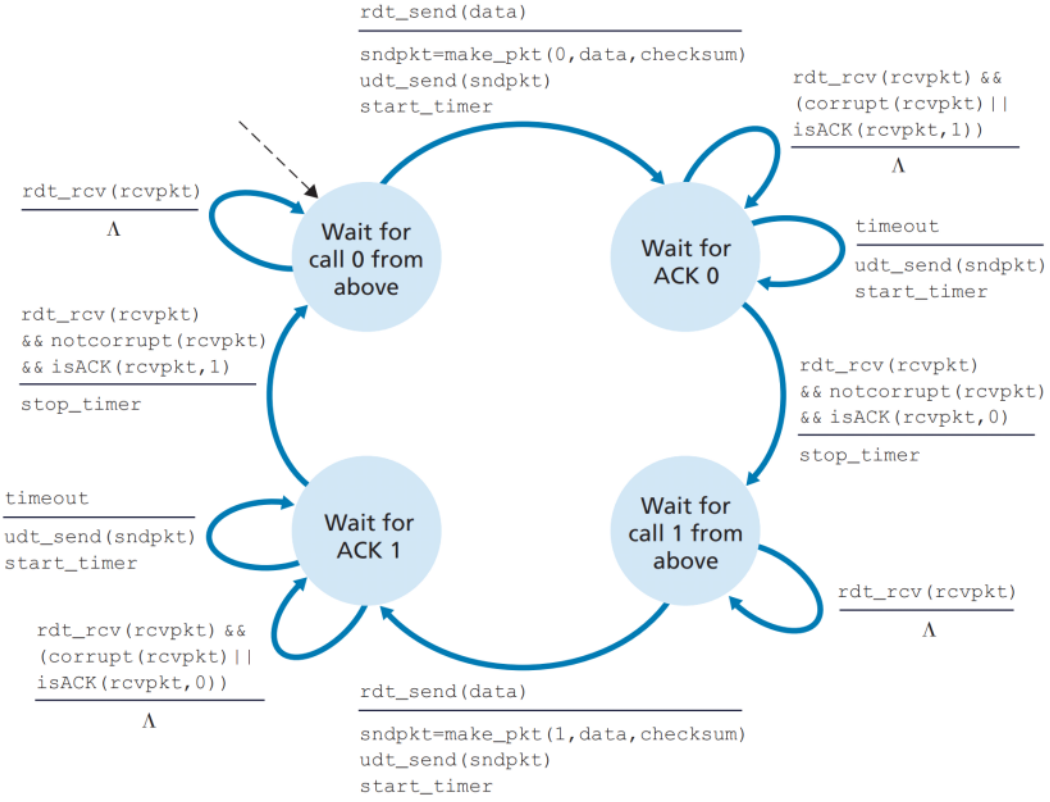


Figure 3.15 ♦ rdt3.0 sender

Go-Back-N (GBN)

- The range of permissible sequence numbers for transmitted but not yet ACKed packets can be viewed as a window of size N over the range of sequence numbers
- N is referred to as the **window size** and the GBN is referred to as a **sliding-window protocol**
- If k is the number of bits in the sequence number field, then the range of sequence numbers is $[0, 2^k - 1]$
- Sender
  - Invocation from above
    - If the window is not full, the sender creates and transmits a packet
    - Otherwise, return the data back to the upper layer or block
  - Receipt of an ACK
    - **Cumulative acknowledgment** - all packets with a sequence number up to and including n have been correctly received at the receiver
  - Timeout
    - The sender resends all packets that have been previously sent but have not yet been acknowledged
    - A single timer for the oldest outstanding packet
    - If an ACK is received but there is still outstanding packets, the timer is restarted
    - Otherwise, stop the timer
- Receiver
  - If a packet with sequence number n is received correctly and is in order (packet with sequence number n-1 has been delivered)

        - Sends an ACK for packet n and delivers to the upper layer
    - Otherwise
        - Discard the packet
        - Resend an ACK for the most recently received in-order packet

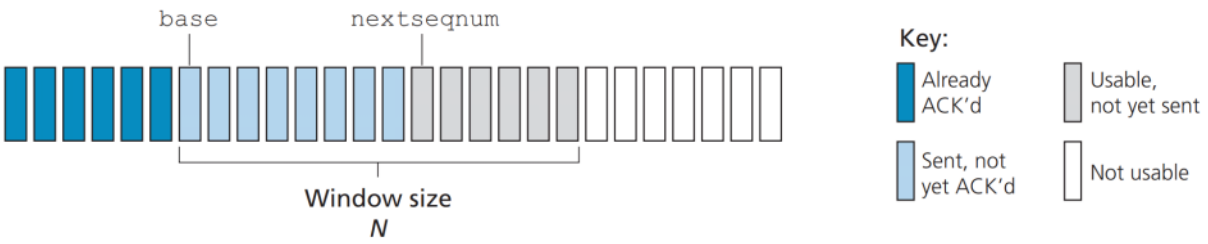


Figure 3.19 ♦ Sender's view of sequence numbers in Go-Back-N

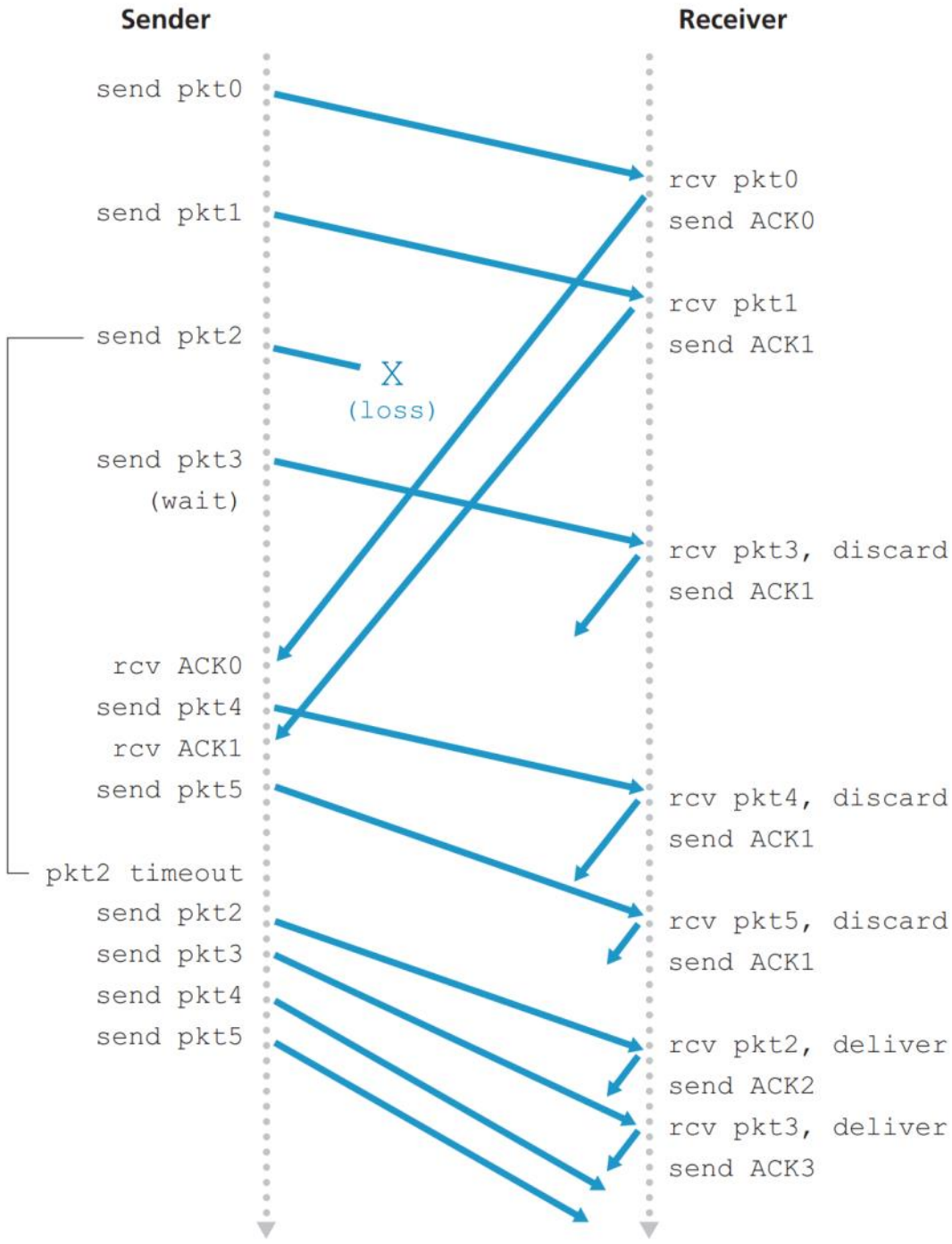


Figure 3.22 ♦ Go-Back-N in operation

Selective Repeat (SR)

- Problem with GBN: retransmit a large number of packets, many unnecessarily
- SR lets the sender retransmit only those packets that it suspects were received in error
- Sender events
    - Data received from above
        - If the next sequence number is within the sender's window, the data is packetized and sent
    - Timeout
        - For detecting lost packets

        - Each packet must have its own logical timer, since only a single packet will be transmitted on timeout
    - ACK received
        - The sender marks that packet as having been received
        - If the packet's sequence number is equal to send_base, the window base is moved forward to the unACKed packet with the smallest sequence number
        - If the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these are transmitted
- Receiver events
    - Packet with sequence number in [rcv_base, rcv_base+N-1] is correctly received
        - A selective ACK packet is returned to the sender
        - If the packet was not previously received, it is buffered
        - If the sequence number is equal to rcv_base, then this packet, and any previously buffered and consecutively numbered packets are buffered and consecutively numbered packets are delivered to the upper layer
        - The receive window is then moved forward by the number of packets delivered to the upper layer
    - Packet with sequence number in [rcv+base-N, rcv_base-1] is correctly received
        - An ACK must be generated
    - Otherwise
        - Ignore the packet

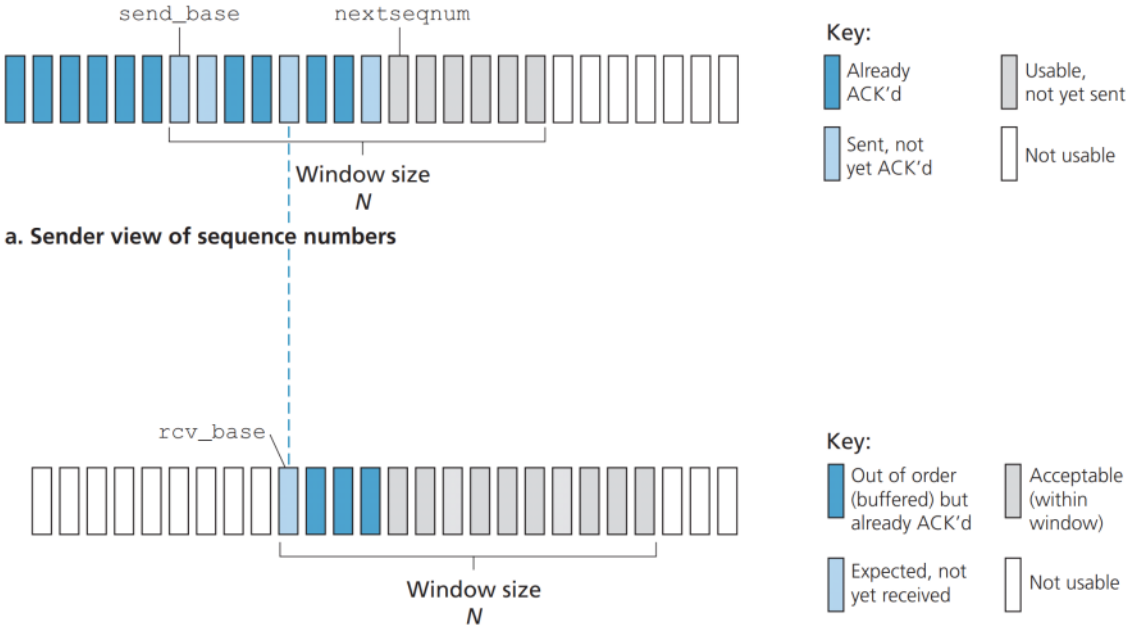


Figure 3.23 ♦ Selective-repeat (SR) sender and receiver views of sequence-number space

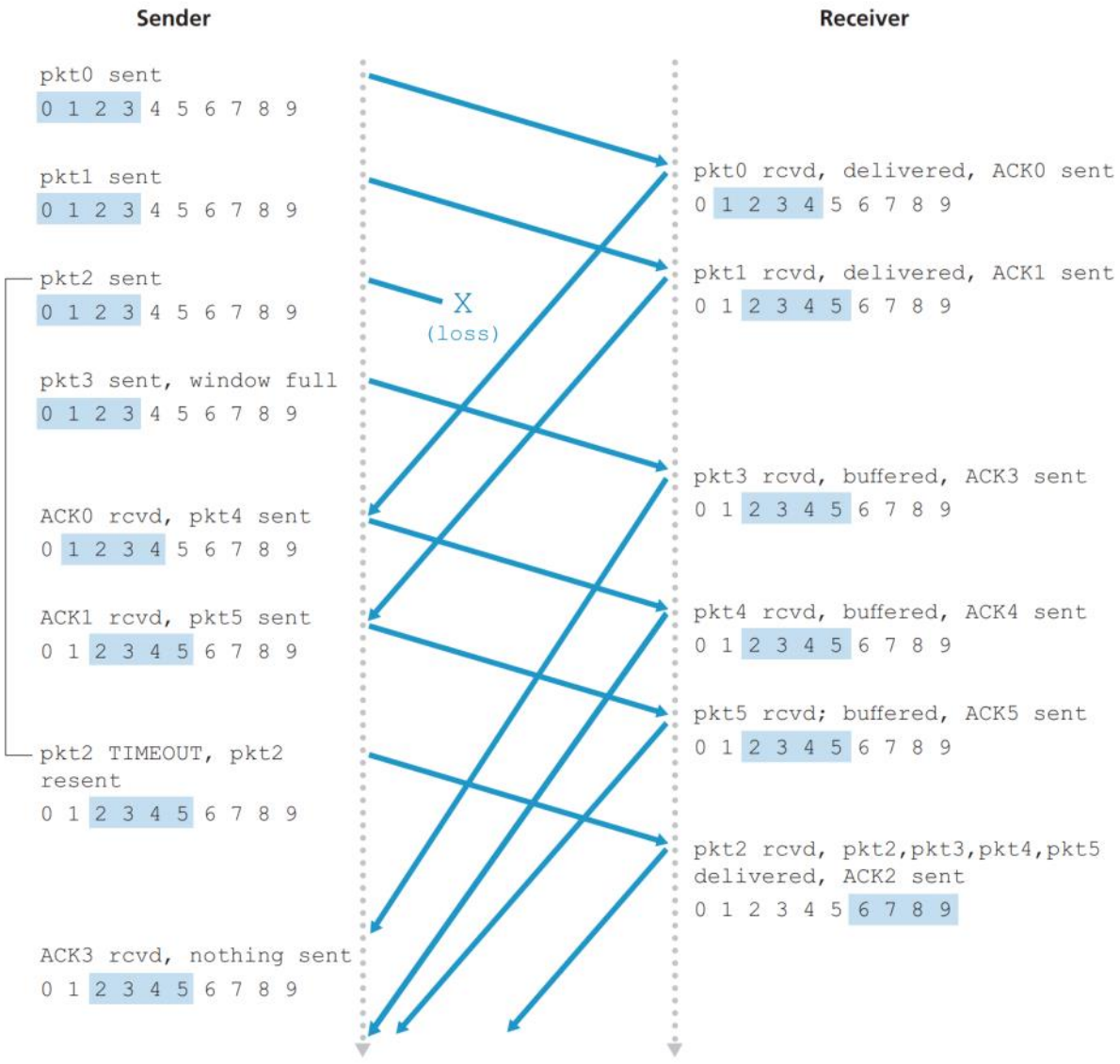


Figure 3.26 ♦ SR operation

Chapter 3.5 - Connection-Oriented Transport: TCP

Sunday, August 6, 2023

The TCP Connection

- Connection-oriented
  - The two processes must first "handshake" with each other
  - They must send some preliminary segments to each other to establish the parameters of the ensuring data transfer
- Runs only in the end systems and not in the intermediate network elements
- The intermediate routers are oblivious to TCP connections
- Full-duplex service
  - Bi-directional
  - A process can be both a sender and a receiver
- Point-to-point
  - Between a single sender and a single receiver
  - Multicasting is not possible with TCP
- Three-way handshake
  - The client first sends a special TCP segment
  - The server responds with a second special TCP segment
  - The client responds again with a third special segment
- Send buffer
  - Set aside during the initial three-way handshake
  - From time to time, TCP will grab chunks of data from the send buffer and pass the data to the network layer
  - Maximum segment size (MSS) - the maximum amount of data that can be grabbed and placed in a segment
    - Typically determined by maximum transmission unit (MTU) minus TCP/IP header length
    - A typical value of MSS is 1460 bytes (1500-byte MTU minus 1460-byte TCP/IP header)
    - Note that the MSS is the maximum amount of application-layer data in the segment
- Receive buffer
  - The application reads the stream of data from this buffer
  - Each side of the connection has its own send and receive buffer

TCP Segment Structure

- Source and destination port numbers - multiplexing/demultiplexing data from/to upper layer
- Checksum field
- 32-bit sequence number field - reliable data transfer
- 32-bit acknowledgement number field - reliable data transfer
- 16-bit receive window - flow control
- 4-bit header length field
  - specify the length of the TCP header in 32-bit words
  - Needed when the TCP header is of variable length
- Options field
- Flag field
  - ACK bit - for acknowledgments
  - RST, SYN, FIN bits - for connection setup and teardown
  - PSH - pass the data to the upper layer immediately

    - URG - there is data marked as "urgent"
        - The location is indicated by the 16-bit urgent data pointer field

Sequence Numbers and Acknowledgement Numbers

- The sequence number for a segment is the byte-stream number of the first byte in the segment
- The acknowledgment number that Host A puts in its segment is the sequence number of the next byte Host A is expecting from Host B
- TCP only acknowledges bytes up to the first missing byte in the stream, so it is said to provide cumulative acknowledgments
- Out-of-order segments
    - Immediately discard, or
    - Keep the out-of-order bytes and waits for the missing bytes to fill in the gaps

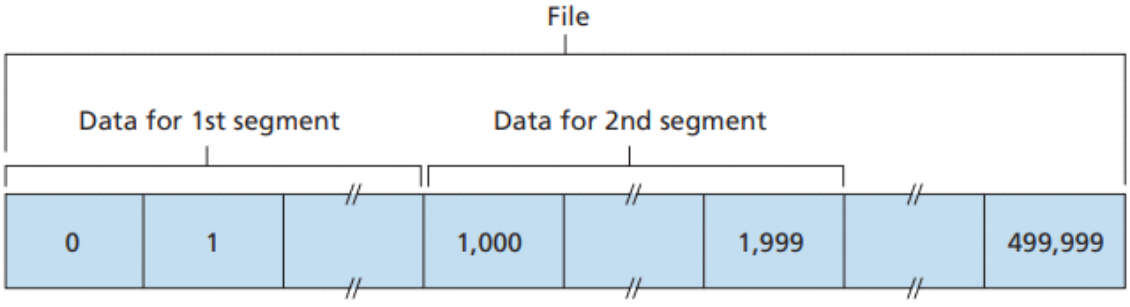


Figure 3.30 ♦ Dividing file data into TCP segments

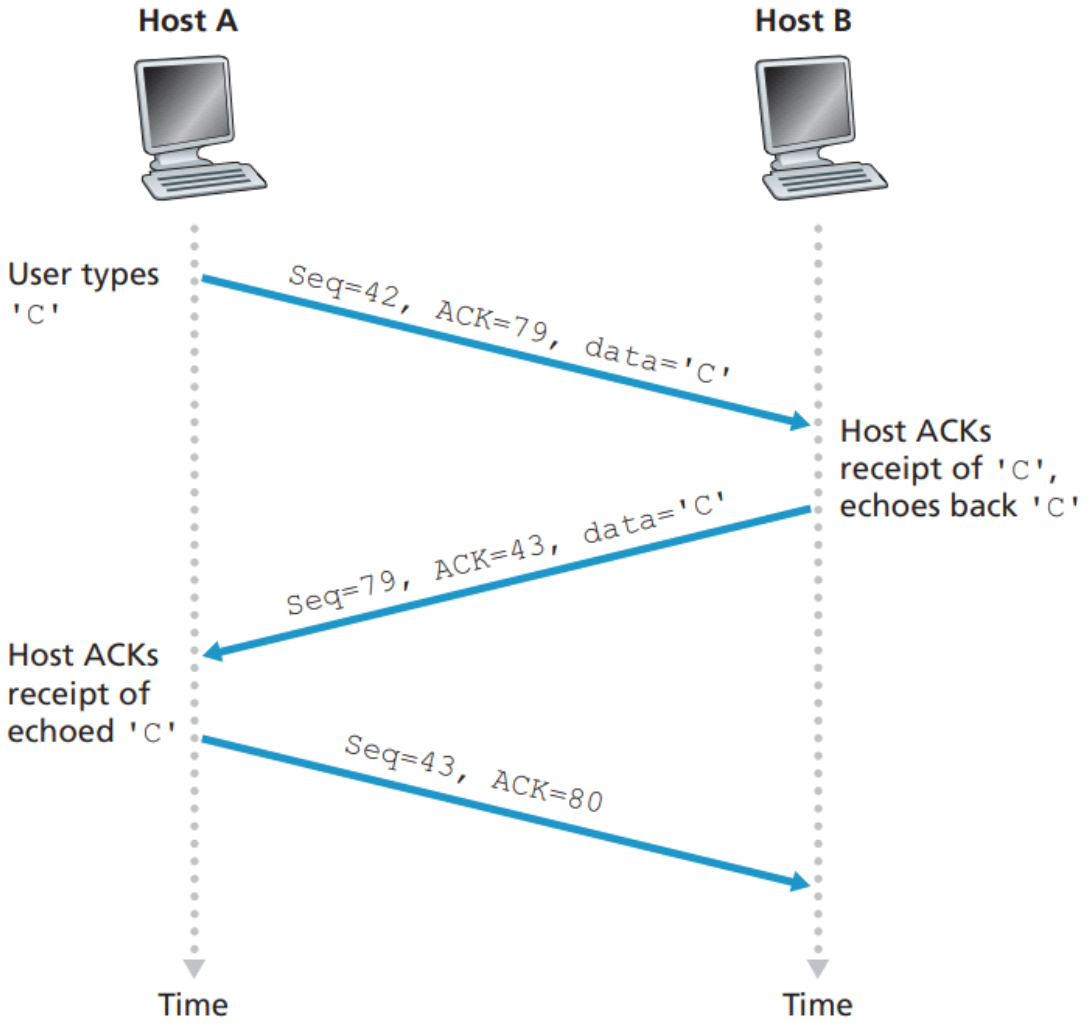


Figure 3.31 ♦ Sequence and acknowledgment numbers for a simple Telnet application over TCP

Estimating the Round-Trip Time

- SampleRTT: the amount of time between when the segment is sent and when an acknowledgment for the segment is received
- TCP only measures SampleRTT for segments that have been transmitted once
- EstimatedRTT: average of SampleRTT values, updated below using this formula:
  - EstimatedRTT = $(1 - \alpha) \cdot$ EstimatedRTT + $\alpha \cdot$ SampleRTT
  - The new value of EstimatedRTT is a weighted combination of the previous value of EstimatedRTT and the new value for SampleRTT
  - The recommended value of α is 0.125
  - Puts more weight on recent samples than on old samples since more recent samples better reflect the current congestion in the network
  - Called an exponential weighted moving average (EWMA)
- DevRTT: an estimate of how much SampleRTT typically deviates from EstimatedRTT
  - DevRTT = $(1 - \beta) \cdot$ DevRTT + $\beta \cdot$ | SampleRTT – EstimatedRTT |
  - DevRTT is an EWMA of the difference between SampleRTT and EstimatedRTT
- It is desirable to set the timeout equal to the EstimatedRTT plus some margin
  - TimeoutInterval = EstimatedRTT + 4 · DevRTT

- An initial TimeoutInterval value of 1 second is recommended
- When a timeout occurs, the value of TimeoutInterval is doubled to avoid a premature timeout
- However, when the segment is received, TimeoutInterval is again computed using the formula above

Reliable Data Transfer of TCP

- Timer
    - TCP uses only a single retransmission timer, even if there are multiple transmitted but not ACKed segments
    - The timer is associated with the oldest unacknowledged segment
- Simplified TCP sender
    - Data received from application above
        - Encapsulates the data in a segment
        - Passes the segment to IP
        - Starts the timer if the timer is not already running for another segment
    - Timer timeout
        - Retransmits the segment that caused the timeout and restarts timer
    - Arrival of an ACK segment from the receiver
        - Compares the ACK value y with its variable SendBase, which is the sequence number of the oldest unacknowledged byte
        - Cumulative acknowledgment - this ACK acknowledges the receipt of all bytes before byte number y
        - Updates its SendBase variable
        - Restarts the timer if there currently are any not-yet-ACKed segments
- Doubling the timeout Interval
    - Each time TCP retransmits, it sets the next timeout interval to twice the previous value
    - However, whenever the timer is started after data reception from application above or ACK received, the TimeoutInterval is derived from the most recent values from EstimatedRTT and DevRTT
        - Provides a limited form of congestion control
- Fast Retransmit
    - A duplicate ACK is an ACK that reacknowledges a segment for which the sender has already received an earlier acknowledgment
    - When the TCP receiver detects an out-of-order segment, it reACKs the last in-order byte of data it has received, causing a duplicate ACK
    - If the TCP sender receives three duplicate ACKs for the same data, it knows that the segment following the ACKed segment has been lost
    - In this case, the TCP sender performs a fast retransmit, which is retransmitting the missing segment *before* the timer expires.
- GBN or SR?
    - A proposed modification to TCP is called selective acknowledgment
        - It allows the receiver to ACK out-of-order segments selectively rather than just cumulatively ACK the last correctly received, in-order segment
    - TCP's error recovery mechanism is best categorized as a hybrid of GBN and SR protocol

Flow Control

- Why flow control
    - When TCP receivers bytes that are correct and in sequence, it places the data in the receive buffer
    - The associated application process will read data from this buffer, but not necessarily at the

instant the data arrives
    - If the process is slow at reading the data, the sender can overflow the receive buffer
- Flow control service
    - Match the rate at which the sender is sending against the rate at which the receiving application is reading
    - The sender maintains a variable called the **receive window**
        - Used to give the sender an idea of how much free buffer space is available at the receiver
    - The receiver maintains the following variables:
        - LastByteRead: the number of the last byte read from the buffer
        - LastByteRcvd: the number of the last byte that has arrived from the network and placed in the receive buffer
    - We must have
        - $\text{LastByteRcvd} - \text{LastByteRead} \leq \text{RcvBuffer}$
    - The receive window, is set to the amount of spare room in the buffer
        - $\text{rwnd} = \text{RcvBuffer} - [\text{LastByteRcvd} - \text{LastByteRead}]$
    - The receiver places its current value of rwnd in the receive window field of every segment it sends to the sender
        - Initially, rwnd = RcvBuffer
    - The sender keeps track of the following:
        - LastByteSent
        - LastByteAcked
    - The sender must ensure throughout the connection that
        - $\text{LastByteSent} - \text{LastByteAcked} \leq \text{rwnd}$
- If rwnd=0
    - The sender has nothing to send as the receiver empties the buffer
    - The sender is never informed that some space has opened up in the receive buffer
    - The sender will be blocked and can transmit no more data
    - Solution: the sender continues to send segments with one byte when rwnd=0
    - Eventually the buffer will begin to empty and the acknowledgments will contain a nonzero rwnd value

TCP Connection Management

- Step 1
    - The client first sends a special **SYN segment** to the server
        - SYN bit = 1
        - Seq number = clinet_isn (randomly chosen initial sequence number)
    - The segment is then encapsulated within an IP datagram and sent to the server
- Step 2
    - The server extracts the segment from the IP datagram and allocates TCP buffers and variables
    - The server sends a connection-granted segment to the client.
        - SYN bit = 1
        - ACK field = client_isn + 1
        - Seq number = server_isn (randomly chosen initial sequence number)
    - This segment is referred to as a **SYNACK** segment
- Step 3
    - Upon receiving the SYNACK segment, the client allocates buffers and variables to the connection
    - The client then sends the server another segment, which acknowledges the server's connection-granted segment

        - ACK field = server_isn + 1
        - SYN bit = 0 (since the connection is established)
        - May carry client-to-server data in the segment payload
- Future segments will have SYN bit = 0
- This is referred to as a three-way handshake
- Either process can end the connection
- When the connection ends, all buffers and variables are deallocated
- Suppose the client issues a close command
    - The client sends a special TCP segment (with FIN bit = 1) to the server
    - When the server receives the segment, it sends an ACK segment in return
    - The server then sends its own shutdown segment with FIN bit =1
    - Finally the client ACK the server's shutdown segment
    - Now all the resources are deallocated
- SYN flood attack
    - The attackers send a large number of TCP SYN segments, without completing the 3rd handshake step (sending back an ACK to the server)
    - The server's connection resources become exhausted as they are allocated for half-open connections
    - Solution: SYN cookies

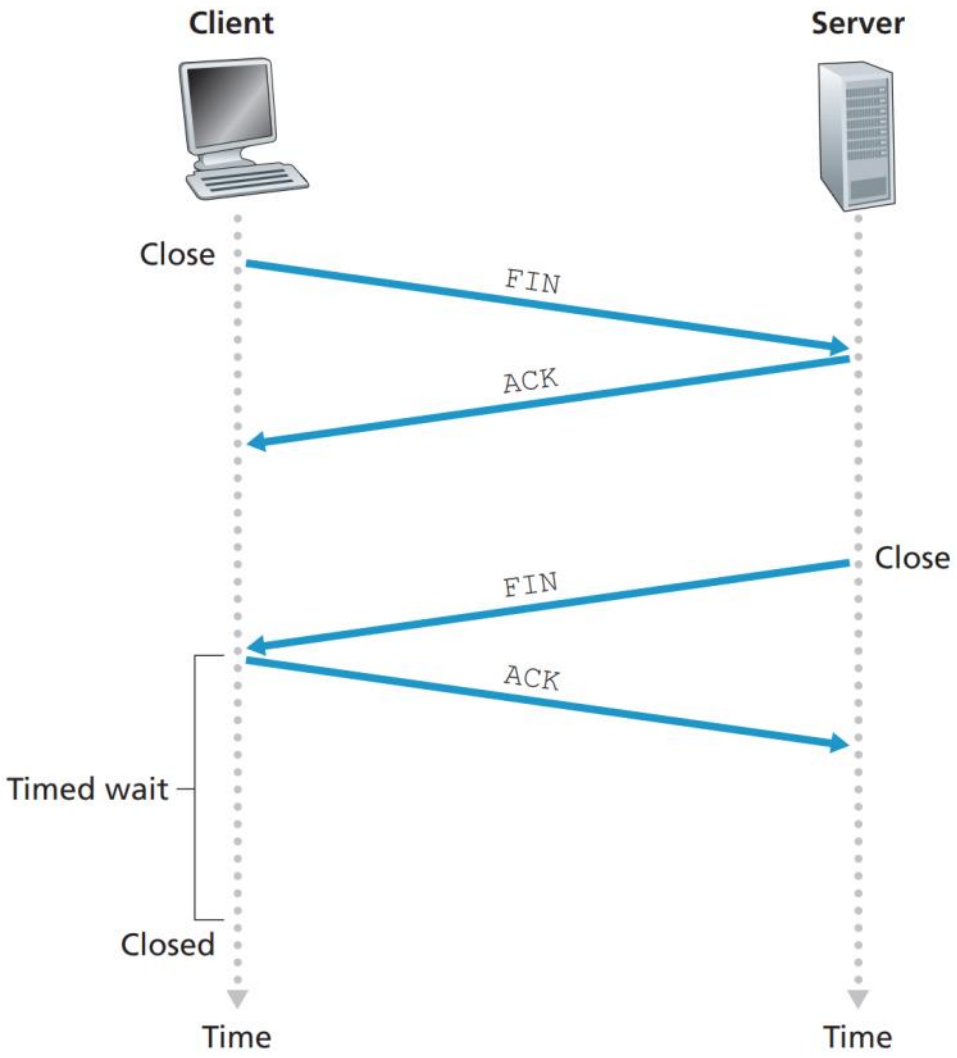


Figure 3.40 ♦ Closing a TCP connection

Chapter 3.7 - TCP Congestion Control

Sunday, August 6, 2023

Approaches to Congestion Control

- End-to-end congestion control
  - The network layer provides no explicit support to the transport layer
  - TCP takes this approach, since the IP layer is not required to provide feedback to hosts regarding network congestion
  - TCP segment loss is taken as an indication of network congestion
- Network-assisted congestion control
  - Routers provide explicit feedback to the sender and/or receiver regarding the congestion state of the network
  - More recently, IP and TCP may optionally implement this approach

TCP Congestion Control

- Each sender limit the rate at which it sends traffic into its connection as a function of perceived network congestion
- The sender keeps track of a variable called the congestion window, cwnd
- This cwnd imposes a constraint on the rate at which a TCP sender can send traffic into the network
  - Specifically, $\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{cwnd}, \text{rwnd}\}$
- Suppose that the receive buffer is large enough, so $\text{LastByteSent} - \text{LastByteAcked}$ is limited by cwnd
- At the beginning of every RTT, the sender is allowed to send cwnd bytes, and at the end of the RTT the sender receives ACK for the data
  - The sender's send rate is roughly cwnd/RTT bps
- By adjusting the value of cwnd, the sender can adjust the rate at which it sends data into its connection
- How should a TCP sender determine the rate at which it should send? TCP uses the following guiding principles:
  - A lost segment implies congestion, and hence the sender's rate should be decreased
  - An acknowledged segment indicates that the network is delivering the sender's segments to the receiver, and hence, the sender's rate can be increased
  - Bandwidth probing - TCP increases its rate in response to arriving ACKs until a loss event occurs, at which point, the transmission rate is decreased
- Three components in TCP congestion-control algorithm
  - Slow start
  - Congestion avoidance
  - Fast recovery

Slow Start

- When a TCP connection begins, cwnd is typically initialized to a small value of 1 MSS, resulting in an initial sending rate of roughly 1 MSS/RTT
- The value of cwnd is increased by 1 MSS every time a transmitted segment is first ACKed
- Sending rate is doubled every RTT
- This exponential growth ends when
  - There is a timeout - the sender sets cwnd to 1 MSS and ssthresh to old_cwnd/2
  - cwnd equals ssthresh - slow start ends and TCP transitions into congestion avoidance

mode
    - Three duplicate ACKs are detected - TCP performs a fast retransmit and enters the fast recovery state

Congestion Avoidance

- TCP increases the value of cwnd by just a single MSS every RTT
    - By increasing cwnd by $\frac{1}{\text{\# MSS in cwnd}} \cdot \text{MSS}$ whenever a new ACK arrives
    - e.g. if MSS is 1460 bytes and cwnd is 14600 bytes, then 10 segments are being sent within an RTT
    - Each arriving ACK increases cwnd by 1/10 MSS
    - Thus cwnd will increase by 1 MSS in total
- This linear increase ends when
    - There is a timeout - the sender sets cwnd to 1 MSS and ssthresh to old_cwnd/2 and enters slow start
    - Three duplicate ACKs are detected - the sender halves the value of cwnd (adding 3 MSS to account of for the triple duplicate ACKS received) and sets ssthresh to old_cwnd/2, and enters fast recovery state

Fast recovery

- The value of cwnd is increased by 1 MSS for every duplicate ACK received for the missing segment that caused the fast-recovery state
- Eventually, when an ACK arrives for the missing segment, TCP enters congestion avoidance state after deflating cwnd
- Fast recovery ends when
    - A timeout event occurs - set cwnd to 1 MSS, ssthresh = old_cwnd/2 and enters slow start
- Recommended but not required
- TCP Tahoe (old) does not have fast recovery but TCP Reno (new) does.

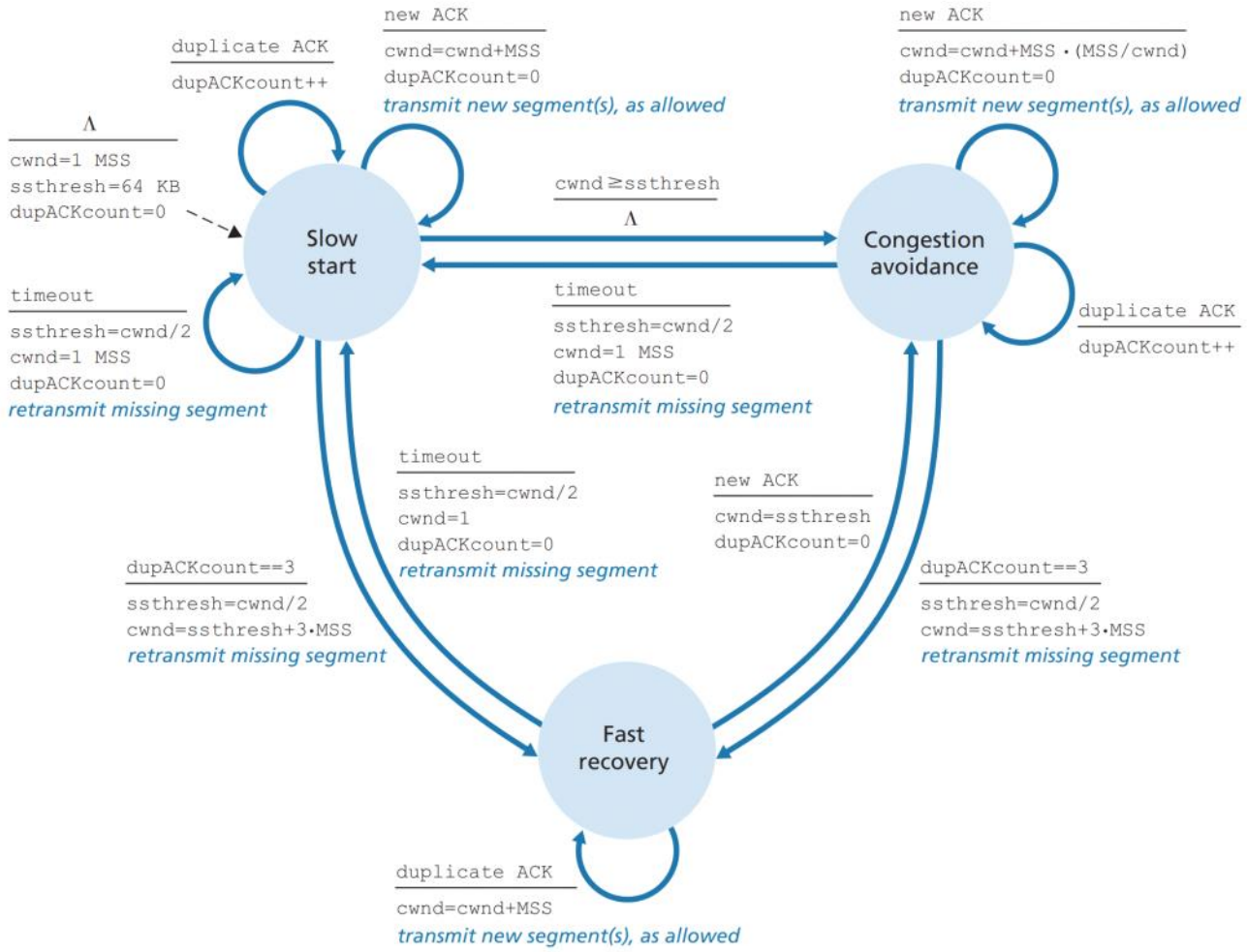
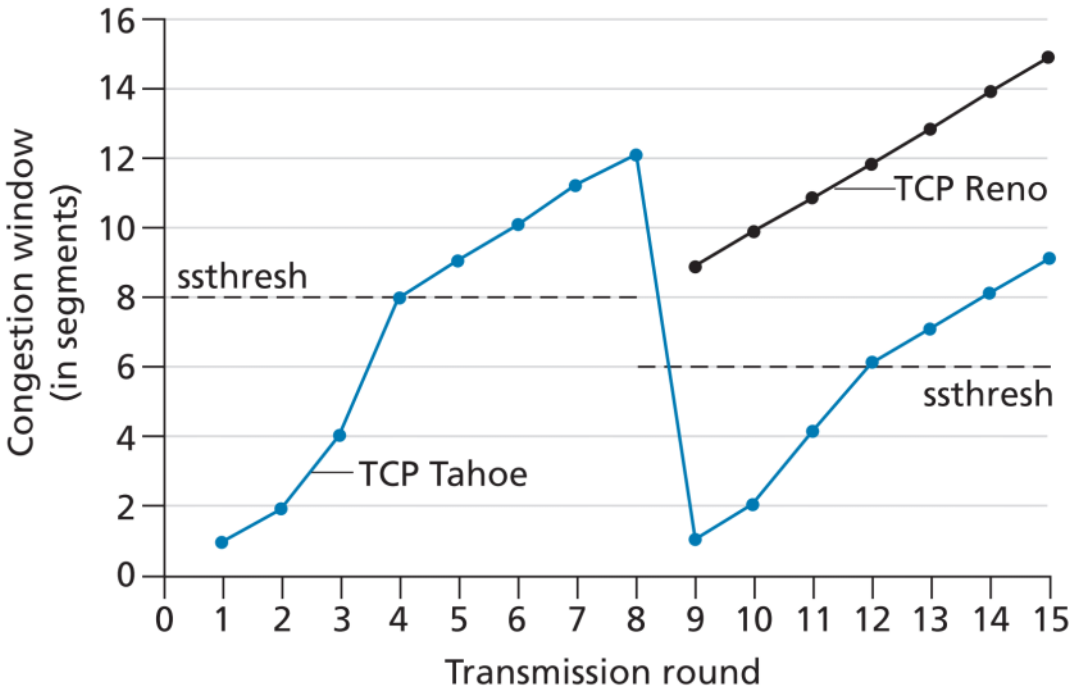


Figure 3.51 ♦ FSM description of TCP congestion control



Explicit Congestion Notification

- Network-assisted congestion control performed within the Internet
- Both TCP and IP are involved
- Two bits in the **Type of Service** field of the IP datagram header
- First setting
    - Used by a router to indicate that it is experiencing congestion
    - The congestion indication bit can be set to signal the onset of congestion to send before loss actually occur
- Second setting
    - Used by the sending host to inform routers that the sender and receiver are ECN-capable
    - Thus capable of taking action in response to ECN-indicated network congestion

Delay-based Congestion Control

- A second congestion-avoidance approach that detects congestion onset before packet loss occurs
- In TCP Vegas
    - The sender measures the RTT for all ACKed packets
    - If the actual sender-measured throughput is close to $cwnd/RTT_{min}$, the TCP sending rate can be increased since the path is not yet congested
    - However, if the actual sender-measured throughput is significantly less than the uncongested throughput rate, the path is congested and the sender will decrease its sending rate

Fairness

- K TCP connections sharing a bottleneck link with transmission rate R bps
- A congestion control mechanism is said to be **fair** if the average transmission rate of each connection is approximately R/K
- TCP congestion control converges to provide an equal share of a bottleneck link's bandwidth among competing TCP connections
- In practice, when multiple connections share a common bottleneck link, sessions with a smaller RTT are able to grab the available bandwidth at that link more quickly as it becomes free
    - Thus they will have higher throughput than those connections with larger RTTs

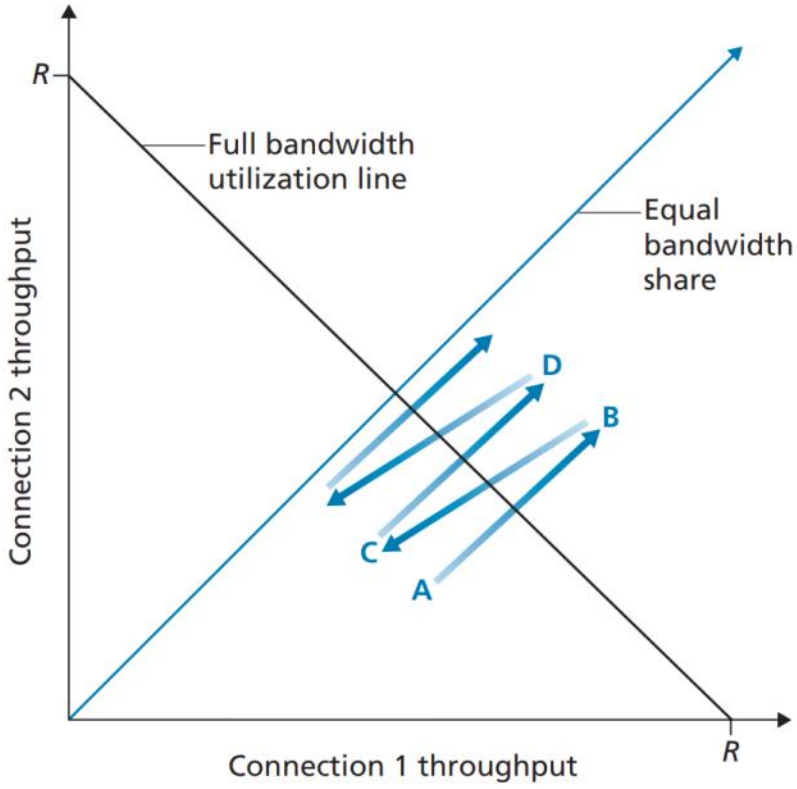


Figure 3.57 ♦ Throughput realized by TCP connections 1 and 2

Fairness and UDP

- Many multimedia applications such as Internet phone and video conferencing often run over UDP since it does not have built-in congestion control
- TCP congestion control will decrease its transmission rates in the face of increasing congestion (loss), while UDP sources need not, so UDP sources might crowd out TCP traffic

Fairness and Parallel TCP Connections

- Web browsers often use multiple parallel TCP connections to transfer the multiple objects within a webpage
- When an application uses multiple parallel connections, it gets a larger fraction of the bandwidth in a congested link

QUIC

- An application-layer protocol providing reliable, congestion-controlled data transfer between two endpoints.
- Connection-oriented and secure
  - Requires a handshake between endpoints to set up the QUIC connection state
  - All QUIC packets are encrypted
  - Combines the handshakes needed to establish connection state with those needed for authentication and encryption
  - Faster establishment
- Streams
  - Multiple "streams" to be multiplexed through a single QUIC connection

    - In HTTP/3, there would be a different stream for each object in a webpage
    - Data from multiple streams may be contained within a single QUIC segment, which is carried over UDP
- Reliable, TCP-friendly congestion-controlled data transfer
    - QUIC provides a reliable in-order delivery on a per-stream basis
    - A lost UDP segment only impacts those streams whose data was carried in that segment
    - HTTP messages in other streams can continue to be received and delivered to the application

Chapter 3 Self Test

Monday, August 7, 2023

Chapter 3.5 - TCP

1. Briefly describe the three-way handshake.
2. What is maximum segment size (MSS)?
3. What is maximum transmission unit (MTU)?
4. What is the relationship between MSS and MTU?
5. Does the length of TCP header vary from segment to segment? Why?
6. Is the receive window field in a TCP header for congestion control, flow control, or something else?
7. How is a TCP segment's sequence number determined?
8. How is a TCP segment's acknowledgement number determined?
9. Why does TCP compute the EstimatedRTT using a exponential weighted moving average?
10. How is DevRTT measured?
11. Is TimeoutInterval set to equal EstimatedRTT?
12. Why is the timeout interval doubled when a timeout occur?
13. Does TCP use a single timer or multiple timers? Which segment is each timer associated with?
14. Describe cumulative acknowledgment.
15. What is a duplicate ACK? When does it happen?
16. Describe TCP's fast retransmit. When does it occur?
17. Why is flow control needed?
18. What are LastByteRead, LastByteRcvd and RcvBuffer? What is the relationship between them?
19. What are LastByteSent, LastByteAcked and rwnd? What is the relationship between them?
20. What is the problem when the receiver's buffer become full so that rwnd=0? What is the solution?
21. Describe the three-way handshake and specify the flag values.
22. Suppose the client issues a close command, describe the four steps of connection termination.
23. Briefly describe SYN flood attack and the countermeasure.

Chapter 3.7 - TCP Congestion Control

1. Why does TCP need congestion control?
2. Why is the sender's send rate roughly cwnd/RTT bps?
3. What does a lost segment imply? What should we do to the sender's rate?
4. What does an ACKed segment imply? What should we do to the sender's rate?
5. What is bandwidth probing?
6. In the slow start, does cwnd increase linearly or exponentially? How is this increase achieved?
7. In the slow start, what happens when
    a. A timeout occurs?
    b. The cwnd equals ssthresh?
    c. There are three duplicate ACKs?
8. In the congestion avoidance, does cwnd increase linearly or exponentially? How is this

increase achieved?

9. In the congestion avoidance, what happens when
    a. A timeout occurs?
    b. Three duplicate ACKs are detected?
10. In the fast recovery, what happens to cwnd for every duplicate ACK received for the missing segment?
11. In the fast recovery, what happens when
    a. A timeout event occurs?
    b. A new ACK is received?
12. What is the difference between TCP Tahoe and TCP Reno?
13. Which header field is used for Explicit Congestion Notification?
14. For a TCP connection sharing a bottleneck link with transmission rate R bps, what does it mean by fair?
15. What kind of applications tends to run over UDP instead of TCP? Why?
16. Can web browsers open multiple parallel TCP connections?
17. What is QUIC? In which layer does it reside? What kind of data transfer is offered?
18. Does QUIC work on top of TCP or UDP?
19. How is QUIC different from TCP/UDP in terms of security?
20. How is QUIC different from TCP + TLS in terms of connection establishment?
21. How is QUIC different in terms of the impact of a lost UDP segment?

Chapter 3 Tricky Concepts

Monday, August 7, 2023

Causes/costs of congestion

Approaches toward congestion control

- End-end congestion control
- Network-assisted congestion control

TCP CUBIC

QUIC

Chapter 3 Review Questions

Thursday, June 15, 2023

R3.

A UDP socket is fully identified by its destination IP address and destination port number.
A TCP socket is fully identified by its source and destination IP addresses, and source and destination port numbers.

R4.

UDP might be better for real-time applications (e.g. video streaming and games) since they are tolerant with packet loss but can benefit from a smaller overhead and lower delay.

R5.

This is because TCP provides reliable data transfer.

R6.

Yes, it is possible to achieve so if reliable data transfer is implemented in the application layer. An example would be QUIC over UDP.

R7.

Yes, they will be directed to the same socket at Host C.
The process at Host C identifies the two segments by the source IP address and port number in the header.

R8.

Not all of the requests are being sent through the same socket at Host C.
Both of the sockets have port 80 as the destination port number. However, they may have different source IP address and source IP number.

R9.

Sequence numbers are needed for checking bit errors since the ACK/NAK bits themselves could have errors.

R10.

Timers are needed for detecting packet loss.

R11.

Yes, the timer would still be necessary because it is used to detect timeout, instead of

counting RTT.

R13.

In GBN, if a packet is not correctly received by the receiver, all packets in the window will be retransmitted.
In Selective Repeat, if a packet is not correctly received by the receiver, only the packets that failed to transmit will be retransmitted.

R14.

a. False
b. False
c. True
d. False
e. True
f. False
g. False

R15.

a. 20 bytes
b. 90

R16.

R17.

Host A has 1/10 R and Host B has 9/10 R, therefore the situation is not fair.

R18.

False. The value of ssthresh is set to one half of the value of cwnd.

Chapter 3 Practice Problems

Thursday, June 15, 2023

P1.

a. source port = 33000; destination port = 80
b. source port = 80; destination port = 33000
c. No; HTTP uses TCP
d. No; It needs multiple TCP connections

P3.

11010001

P4.

Adding up the checksum with the sum of all 16-bit words we get 10001111 11111111
This segment is not considered correctly received.
The receiver might simply discard the segment or pass the damaged segment to the application with a warming.

P5.

The receiver cannot be absolutely certain that no bit errors have occurred since the checksum field itself could have been altered

P6.

P7.

ACK packets do not require sequence numbers because they can use the sequence numbers on the sender's packets to detect timeout.

P9.

Sender	Receiver
Send pkt0	
	Receive corrupt
	Send NAK0
Receive corrupt	
Timeout; resend pkt0	
	Receive pkt0
	Send ACK0
Receive corrupt	
Timeout; resend pkt0	
	Receive pkt0
	Send ACK0
Receive ACK0	
Send pkt1	
	...

P10.

It would be the same as rdt3.0???

P11.

If wait-for-1-from-below state has no actions, the sender will be stuck at wait-for-ACK-1 state forever.
If wait-for-0-from-below state has no actions, the sender will be stuck at wait-for-ACK-0 state forever.

P12.

P13.

UDP has no congestion control so the application itself needs to implement congestion control on top of the transport layer.
TCP has congestion control.

P14.

This protocol would work over a channel with only bit errors.
However it would not work over a channel with bit errors and packet losses.

P25.

An application has more control of what data is sent because ??? .

An application has more control of what data is sent because the application write data to the sender buffer and TCP will grab bytes without necessarily putting a single message in the TCP segment. UDP, however, encapsulates in a segment whatever the application gives to it.
An application has more control of when data is sent because there is no acknowledgements, congestion or flow control in UDP.

P26.

a. $2^{32} \cdot 536$
b. $\frac{2^{32} \cdot (536 + 66)}{155 \cdot 10^6} = 16681$

a. 2^{32}
b. $2^{32} + \frac{2^{32}}{536} \cdot 66$

P27.

a. The sequence number is 97 + 40 = 137. The source port number is 302 and the destination port number is 80.
b. The acknowledgement number is 137. The source port number is 80 and the destination port number is 302.
c. The acknowledgement number is 97.
d. The sequence number is 97.

P28.

TCP flow control prevents the sender from overflowing receiver's buffer.

P29.

a. It uses a special initial sequence number in the SYNACK because it does not know if the segment is coming from a real user or a SYN flood attack. The special initial sequence number is the "cookie".
b. The attackers cannot create half-open or fully open connections by simply sending an ACK packet to the target.
c. The attacker can create many fully open connections if the ACK numbers are the cookie value plus one.

P31.

Round	SampleRTT	EstimatedRTT	DevRTT
0		120	6
1	112	119	
2	140	121.63	
3	110	120.17	
4	90	116.4	
5	90	113.1	

P33.

TCP avoids measuring SampleRTT for retransmitted segments because retransmission can happen when there is a high congestion in the network, but retransmitted RTT does not account for previous rounds, so it might not be able to reflect the situation accurately.

P34.

~~LastByteRcvd − SendBase < SenderWindow~~
SendBase − 1 ≤ LastByteRcvd

P35.

~~LastByteRcvd − y ≤ SenderWindow~~
y − 1 ≤ LastByteRcvd

P36.

If only one duplicate ACK has occurred, the congestion level in the network might not be very high, so a fast retransmit might increase the congestion level in the network.

P37.

a. GBN: Host A sent 5+4=9 packets (1 2 3 4 5 2 3 4 5). ~~Host B sent 5 ACKs (1 1 1 1 5)~~
GBN: Host B sent 8 ACKs (1 1 1 1 2 3 4 5)
SR: Host A sent 5+1=6 packets (1 2 3 4 5 2). Host B sent 5 ACKs (1 3 4 5 2)
TCP: Host A sent 5+1 = 6 packets (1 2 3 4 5 2). ~~Host B sent 5 ACKs (1 1 1 1 5)~~
TCP: Host B sent 5 ACKs (2 2 2 2 6)

b. TCP because it uses fast retransmit.

P38.

At the beginning of the RTT, sender is allowed to send cwnd segments of data and at the end of the RTT the sender receives ACK for the data. Thus the sender's sending rate is approximately cwnd segments per RTT.

P32.

a. $E_1 = 0.9E_2 + 0.1S_1$
$= 0.9(0.9E_3 + 0.1S_2) + 0.1S_1$
$= 0.9(0.9(0.9E_4 + 0.1S_3) + 0.1S_2) + 0.1S_1$
$= 0.9(0.9(0.9(0.9E_5 + 0.1S_4) + 0.1S_3) + 0.1S_2) + 0.1S_1$
$= 0.9^4E_5 + 0.9^3 \cdot 0.1S_4 + 0.9^2 \cdot 0.1S_3 + 0.9 \cdot 0.1S_2 + 0.1S_1$

b. $0.9^nE_{n+1} + 0.9^{n-1} \cdot 0.1S_n + 0.9^{n-2} \cdot 0.1S_{n-1} + \cdots + 0.9^0 0.1S_1$
$= 0.9^nE_{n+1} + \sum_{i=0}^{n-1} 0.9^i \cdot 0.1S_{i+1}$

P40.

a. 1 to 6 and 23 to 26
b. 6 to 16 and 17 to 22
c. Triple duplicate ACK
d. Timeout
e. ~~64 KB~~
   32 MSS
f. ~~16 MSS~~
   21 MSS
g. ~~13th~~
   7th
h. The ssthresh is set to cwnd/2 = 4 MSS and the cwnd is set to 4 + 3 = 7 MSS
i. ~~The ssthresh is set to cwnd/2 = 18 MSS~~ and the cwnd is set to 1 MSS
j. At round 17, ssthresh = 21 MSS and cwnd = 1.
   Round 17: 1
   Round 18: 2
   Round 19: 4
   Round 20: 8
   Round 21: 16
   ~~Round 22: 17~~
   ~~Total 48 segments~~
   Round 22: 21
   Total 52 segments

P42.

TCP needs a window-based congestion-control mechanism to limit the number of new segments the sender sends into the network

P43.

TCP congestion control because the receiver buffer will never have an overflow.

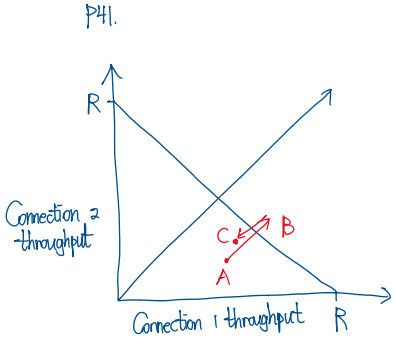
P44.

a. 6 RTTs
b.

RTT	# segments sent
1	6
2	7
3	8
4	9
5	10
6	11

Average = 8.5 MSS/RTT

P41.



P48.

a. $\frac{54 \times 10^6}{536} = 100746$

b. $\frac{100746}{2} = 50373$

   $\frac{100746}{2} \cdot 536 = 27 \text{ Mbps}$

c. 100746 secs

a. We need $\frac{W \cdot MSS}{RTT} = 54 \text{ Mbps}$

b. $\frac{0.75W}{RTT}$

c. $\frac{W}{2} \cdot RTT$

Lecture 13

Tuesday, May 16, 2023

4-5

- Routing
  - Determine route taken by packets from source to destination
  - Relies on global information
  - Routers cooperate and exchange information
  - Router protocols specify how this works
  - The end goal is to facilitate forwarding
- Forwarding
  - Move packets from a router's input link to appropriate router's output link
  - Relies on local information
  - Per packet decision

4-6

- Data plane
  - Local, per-router function
  - Determines how datagram arriving on router input port is forwarded to router output port
- Control plane

4-7

- Every router has a control plane and a data plane
- When an IP packet arrives, the values in the header are used to look up the output link in the local forwarding table
- If this value does not exist, use the default value (also specified in the forwarding table)

4-8

- Logically centralized control plane
- Physically distributed across server

4-10

- Best effort service model
- No guarantees on
  - Reliability (ensuring no loss)
  - Timing or order
  - Throughput
  - …
- They can be implemented in the layers above. For example:
  - TCP in transport layer provides reliability
  - TLS in application layer provides security

4-12

- Shallow copies of the forwarding tables are sent to each router input port as soon as they are computed

- Green - physical layer
- Blue - data-link layer
- Red - network layer
- The routing processor needs to have a high performance

4-15

- In the forwarding table, only specify a range of IP addresses
- Match the longest prefix
- Example answers:
    - 0
    - 1

4-20

- Switching fabrics transfer packet from input link to appropriate output link
- Switching rate = rate at which packets can be transferred from inputs to outputs
    - Measured as multiple of input/output line rate
    - N inputs: a switching rate of N times line rate is desirable
- More recent approach: shared bus
    - Widely used in small networks
- More recent approach: crossbar switch
    - More expensive
    - Better performance
    - Small networks tend to use this

Lecture 14

Thursday, June 29, 2023

4-22

- Priority scheduling
    - Being assigned a low priority is referred to as **throttling**

4-24

- ECN - explicit congestion notification
- RED - random early detection
    - Randomly drops packets in queues

4-29

- IP is not the only protocol at the network layer

4-31

- ToS: 0:5 stands for differentiated service an 6:7 stands for ECN
- Maximum length = 2^{16} 65536 bytes
- MTU discovery
    - Deduce MTU for the path
    - Ethernet MTU = 1500 bytes
- MTU limits the datagram's size
- TTL: maximum hop that the IP datagram can take
    - Avoid cycles
- Upper layer protocol
    - TCP=6
    - UDP=17
- After the packets are fragmented, the routers are not responsible for reassembling back
    - The burden is put on the receiver side
- MF: more fragments
    - MF=1 means there is fragmentation

4-32

- Reassembly only happens at the receiver

4-33

- Offset = 1480/8 = 185

4-34

- a.b.c.d, each letter stands for 8 bits
- IP addresses are assigned to router **interface**
- Interface
    - Connection between host/router and physical link
    - Routers typically have multiple interfaces
    - Host typically have one or two interfaces
- Subnet mask
    - a.b.c.d/**24**

    - The first three bytes are the network part
    - The last one byte is the host part
    - There can be 2^8 hosts
- The clouds are referred to as LAN

Chapter 4.1 - Overview of Network Layer

Thursday, July 6, 2023

Forwarding and Routing: The Data and Control Planes

- Forwarding
  - Router-local action of transferring a packet from an input link interface to the appropriate output link interface
  - Takes a few nanoseconds
  - Typically implemented in hardware
- Routing
  - Network-wide process that determines the end-to-end paths that packets take from source to destination
  - Takes seconds
  - Typically implemented in software
- A key element in every network router is its **forwarding table**
  - Forwards a packet by examining the value of one or more fields in the arriving packet's header
  - Then use these header values to index into its forwarding table

Control Plane: The Traditional Approach

- The **routing algorithm** determines values in the forwarding tables
- A routing algorithm runs in each and every router and both forwarding and routing functions are contained within a router
- The routing algorithm function in one router communicates with the routing algorithm function in other routers to compute the values for its forwarding table
- This is done by exchanging routing messages containing routing information according to a routing protocol

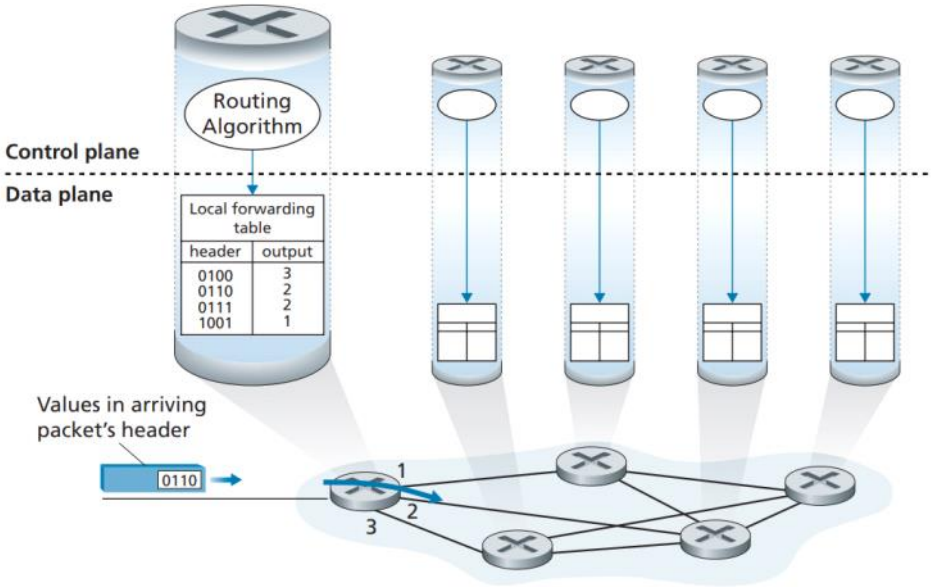


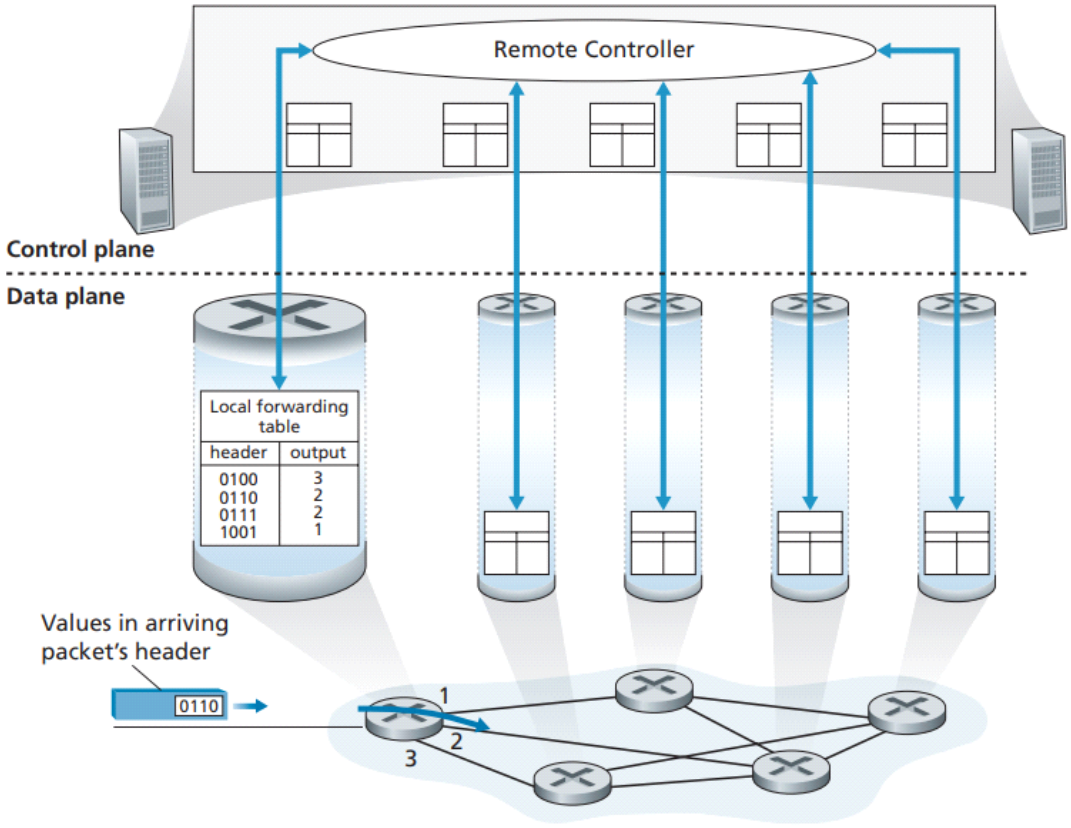
Figure 4.2 ♦ Routing algorithms determine values in forward tables

Control Plane: The SDN Approach

- A physically separate, remote controller computes and distributes the forwarding tables to

be used by each and every router

- Note that the data plane components are identical
- However, control-plane routing is separated from the physical router
  - The routing device performs forwarding only
- This is referred to as **software-defined networking (SDN)**



Network Service Model

- Possible services that the network layer could provide
  - Guaranteed delivery
  - Guaranteed delivery with bounded delay
  - In-order packet delivery
  - Guaranteed minimal bandwidth
  - Security
- The Internet's network layer provides a single service, known as **best-effort service**
- None of the services mentioned above are provided by IP

Chapter 4.2 - Router

Thursday, July 6, 2023

High-level View of a Generic Router Architecture

- Input ports perform several key functions, including
  - Physical layer function of terminating an incoming physical link at a router
  - Link-layer functions needed to interoperate with the link layer at the other side of the incoming link
  - Lookup function in the rightmost box
- Switching fabric
  - Connects the router's input ports to its output ports
  - Completely contained within the router
- Output ports
  - Store packets received from the switching fabric and transmit them on the outgoing link
- Routing processor
  - Performs control-plane functions
  - In traditional routers
    - Executes the routing protocols
    - Maintains routing tables
    - Computes forwarding tables
  - In SDN routers
    - Communicates with the remote controller in order to receive forwarding table entries
    - Installs these entries in the router's input ports
    - Also performs the network management functions
- If N ports are combined on a line card, the datagram-processing pipeline must operate N times faster
- Therefore they are usually implemented in hardware

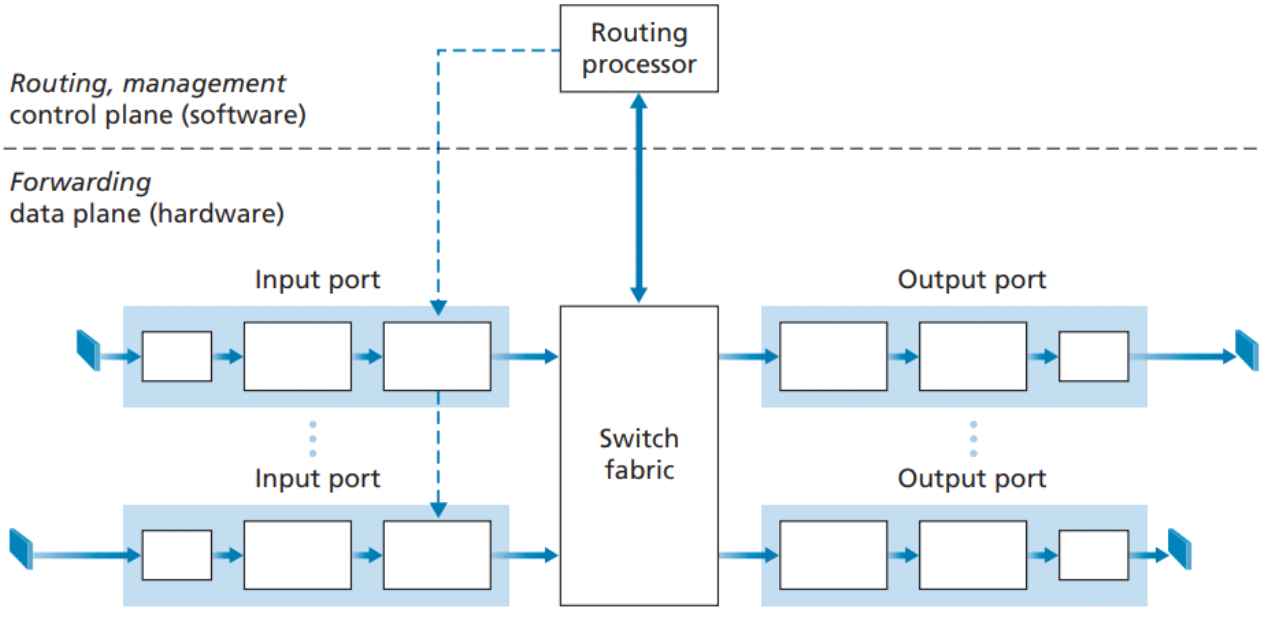


Figure 4.4 ♦ Router architecture

Input Port Processing and Destination-Based Forwarding

- The forwarding table is copied from the routing processor to the line cards over a separate bus
- With such a shallow copy at each line cards, forwarding decisions can be made locally, at each input port, without invoking the centralized routing processor

Prefix	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
Otherwise	3

- The router matches a **prefix** of the packet's destination address
- Example: suppose a destination address is 11001000 00010111 00010110 10100001
  - Since the 21-bit prefix matches the first entry in the table, the router forwards the packet to link interface 0
- If a prefix does not match any of the first three entries, then the router forwards the packet to the default interface 3
- When there are multiple matches, the router uses the **longest prefix matching rule**
- Example: 11001000 00010111 00011000 10101010 matches with second entry (interface 1) in the table
- In practice, **Ternary Content Addressable Memories (TCAMs)** are often used for lookup
  - With a TCAM, a 32-bit IP address is presented to the memory, which returns the content of the forwarding table entry for that address in essentially constant time

Match Plus Action

- The input port steps of looking up a destination IP address ("match") and then sending the packet into the switching fabric to the specific output port ("action") is a specific case of a more general **match plus action** abstraction
- It is performed in many networked devices, not just routers
- Examples:
  - In link-layer switches, link-layer destination addresses are looked up (match) and the frame is sent into the switching fabric (action)
  - In firewalls, an incoming packet whose header matches a given criteria may be dropped (action)
  - In a network address translator (NAT), an incoming packet whose transport-layer port number matches a given value will have its port number rewritten before forwarding (action)

Switching

- Switching via memory
  - Switching done under direct control of the CPU (routing processor)
  - Input and output ports functioned as traditional I/O devices in a traditional operating system

    - If the memory bandwidth is such that a maximum of B packets per second can be written into, or read from, memory, then the overall forwarding throughput must be less than $B/2$
    - Two packets cannot be forwarded at the same time, even if they have different destination ports, since only one memory read/write can be done at a time
- Switching via a bus
    - An input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
    - The input port prepend a switch-internal label (header) to the packet indicating the local output port to which this packet is being transferred
    - All output ports receive the packet, but only the port that matches the label will keep the packet
    - The label is then removed at the output port, since it is only used within the switch
    - The switching speed of the router is limited to the bus speed
    - Sufficient for routers that operate in small local area and enterprise networks
- Switching via an interconnection network
    - A **crossbar switch** is an interconnection network consisting of $2N$ buses that connect N input ports to N output ports
    - Each vertical bus intersects each horizontal bus at a crosspoint which can be opened or closed at any time by the switch fabric controller
    - When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the crosspoint at the intersection of busses A and Y, and port A then sends the packet onto its bus, which is picked up (only) by bus Y
    - Note that a packet from port B can be forwarded to port X at the same time, since the packets use different input and output busses
    - A crossbar switch is **non-blocking** - a packet being forwarded to an output port will not be blocked from reaching that output port as long as no other packet is currently being forwarded to that output port
    - A router's switching capacity can also be scaled by running multiple switching fabrics in parallel
        - Input ports and output ports are connected to N switching fabrics that operate in parallel
        - An input port breaks a packet into K smaller chunks, and sends the chunks through K of these N switching fabrics to the selected output port
        - The output port reassembles the K chunks back into the original packet

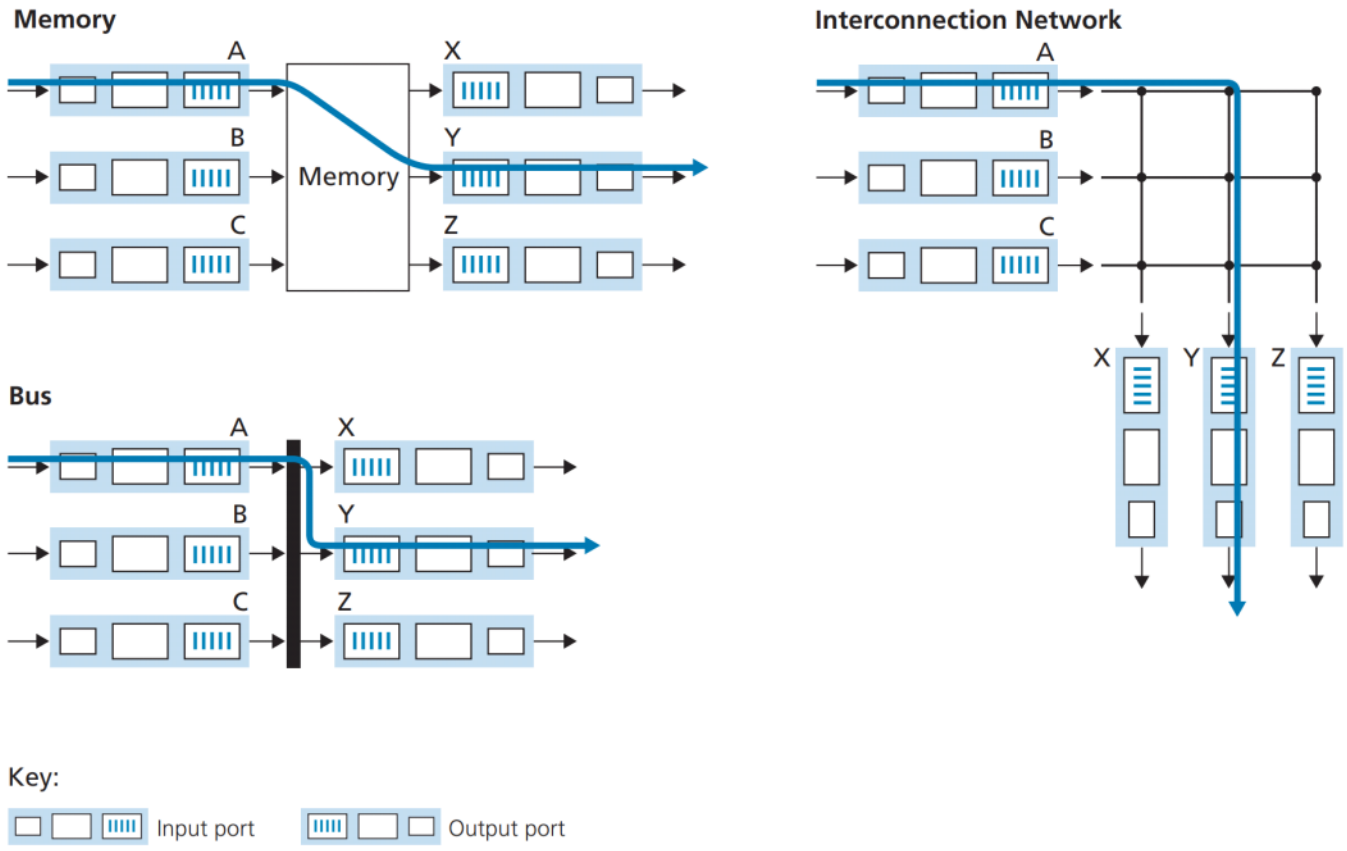


Figure 4.6 ♦ Three switching techniques

Where Does Queueing Occur

- Packet queues may form at both the input ports and the output ports
- As these queues grow large, the router's memory can eventually be exhausted and **packet loss** will occur when no memory is available to store arriving packets
- Suppose that the input and output line speeds (transmission rates) all have an identical transmission rate of R_{line} packets per second, and there are N input ports and N output ports
- Define the switching fabric transfer rate R_{switch} as the rate at which packets can be moved from input port to output port
- If R_{switch} is N times faster than R_{line}, then only negligible queuing will occur at the input ports
- In the worst case, all N input lines are receiving packets that are to be forwarded to the same port
    - Each batch of N packets (one per input port) can be cleared through the switch fabric before the next batch arrives

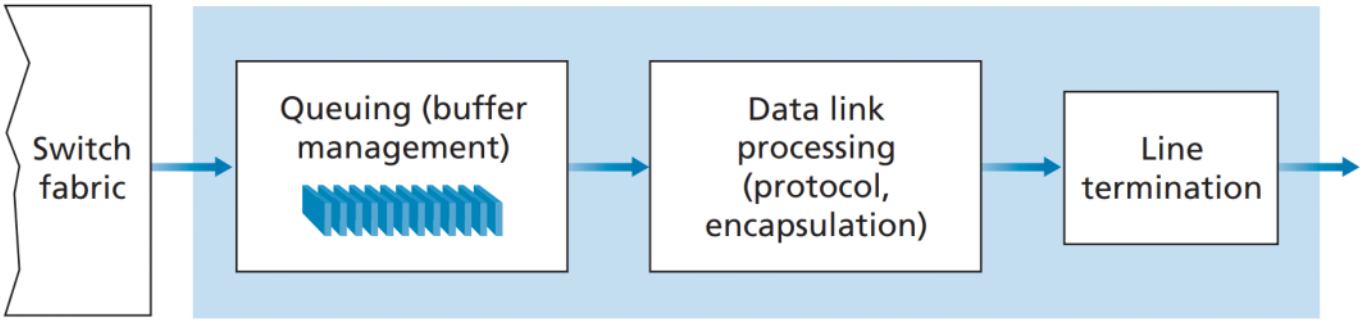


Figure 4.7 ♦ Output port processing

Input Queuing

- Head-of-the-line (HOL) blocking in an input-queued switch - a queued packet in an input queue waiting for transfer through the fabric (even though its output port is free) because it is blocked
- Due to HOL blocking, the input queue will grow to unbounded length (significant packet loss will occur) under certain assumptions as soon as the packet arrival rate on the input links reaches only 58 percent of their capacity

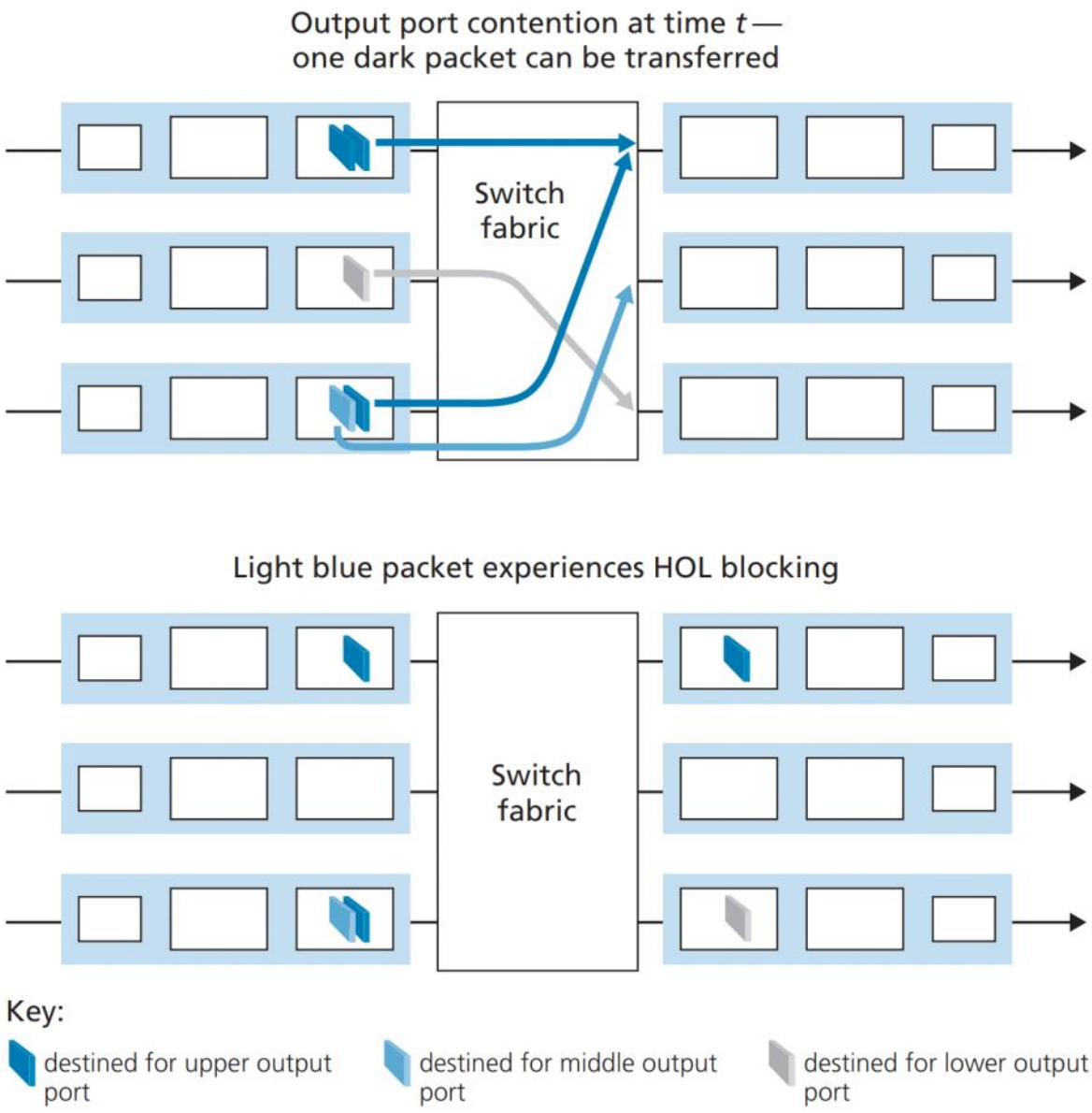


Figure 4.8 ♦ HOL blocking at and input-queued switch

Output Queueing

- Suppose that R_{switch} is N times faster than R_{line} and that packets arriving at each of the N input ports are destined to the same output port
- In the time it takes to send a single packet onto the outgoing link, N new packets will arrive at this output port
- Since the output port can transmit only a single packet in a unit of time, the N arriving packets will have to queue
- Thus packet queues can form at the output ports even when the switching fabric is N times faster than the port line speeds
- When there is not enough memory to buffer an incoming packet, a decision must be made to either drop the arriving packet (drop-tail) or remove one or more already-queued packets.
- In some cases, it might be useful to drop (or mark the header of) a packet before the buffer is full in order to provide a congestion signal to the sender
    - This marking could be done using the ECN bits

- A number of proactive packet-dropping and packet-marking policies are collectively known as **active queue management** (AQM) algorithms
    - One of them is the **Random Early Detection (RED)** algorithm

How Much buffering Is "Enough"?

- The rule of thumb [RFC 3439] for buffer sizing was that the amount of buffering (B) should be equal to an average RTT times the link capacity (C)
    - $B = RTT \cdot C$
    - Example: a 10-Gbps link with an RTT of 250 msec (= 0.25 sec) would need an amount of buffering equal to 2.5 Gbits of buffers
- More recent theoretical and experimental efforts suggest that when a large number of independent TCP flows (N) pass through a link, the amount of buffering needed is $B = RTT \cdot C/\sqrt{N}$
- Larger buffers
    - Advantage: absorb larger fluctuations in the packet arrival rate, thereby decreasing the router's packet loss rate
    - Disadvantage: potentially longer queuing delays

Packet Scheduling

- FCFS (aka FIFO)
- Priority queuing
- Round-robin queuing

First-in-First-Out (FIFO)

- Selects packets for link transmission in the same order in which they arrived at the output link queue
- Packets leave in the same order in which they arrived

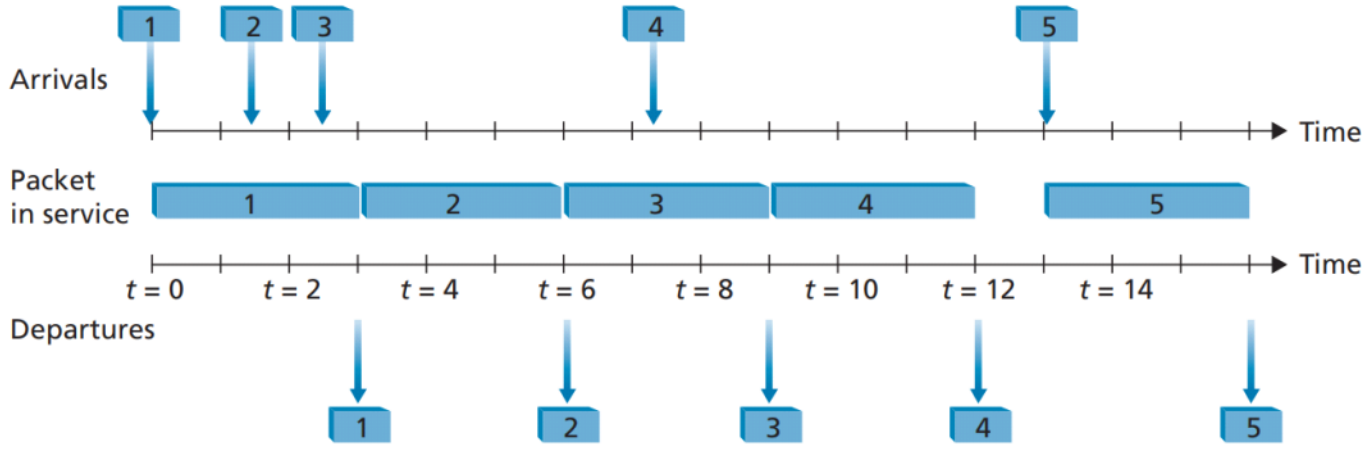


Figure 4.12 ♦ The FIFO queue in operation

Priority Queueing

- Packets arriving at the output link are classified into priority classes upon arrival at the queue
- Each priority class typically has its own queue
- When choosing a packet to transmit, the priority queuing discipline will transmit a packet

from the highest priority class that has a nonempty queue

- The choice among packets in the same priority class is typically done in a FIFO manner

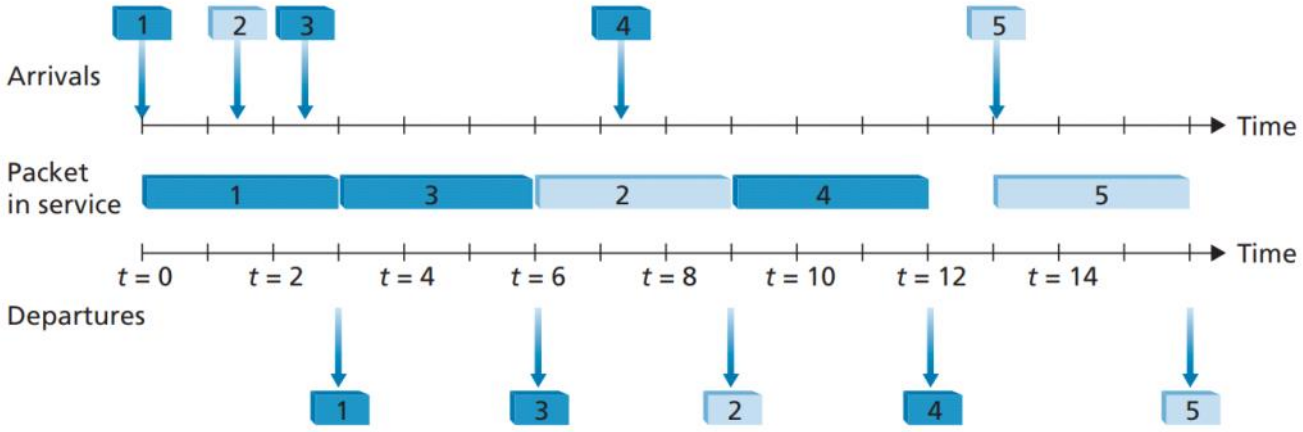


Figure 4.14 ♦ The priority queue in operation

Round Robin and Weighted Fair Queuing (WFQ)

- Round robin queuing
  - Under the round robin queuing discipline, packets are sorted into classes as with priority queuing, and alternates service among the classes
  - A so-called work-conserving queuing discipline will never allow the link to remain idle whenever there are packets (of any class) queued for transmission
- Weighted fair queuing
  - A generalized form of round robin queuing
  - A WFQ scheduler will serve classes in a circular manner and is also work-conserving queuing discipline
  - However, each class may receive a different amount of service in any interval of time
  - Each class i is assigned a weight w_i
  - During any interval of time during which there are class i packets to send, class i will be guaranteed to receive a fraction of service equal to $w_i/\sum w_j$
  - For a link with transmission rate R, class i will always achieve a throughput of at least $R \cdot w_i / \sum w_j$

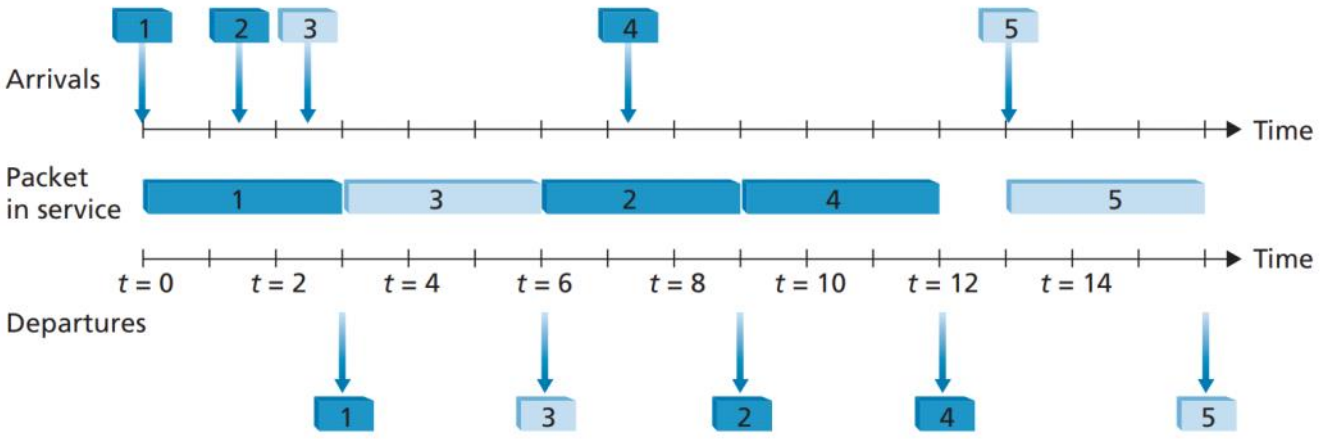


Figure 4.15 ♦ The two-class robin queue in operation

Chapter 4.3 - The Internet Protocol (IP)

Friday, July 7, 2023

IPv4 Datagram Format

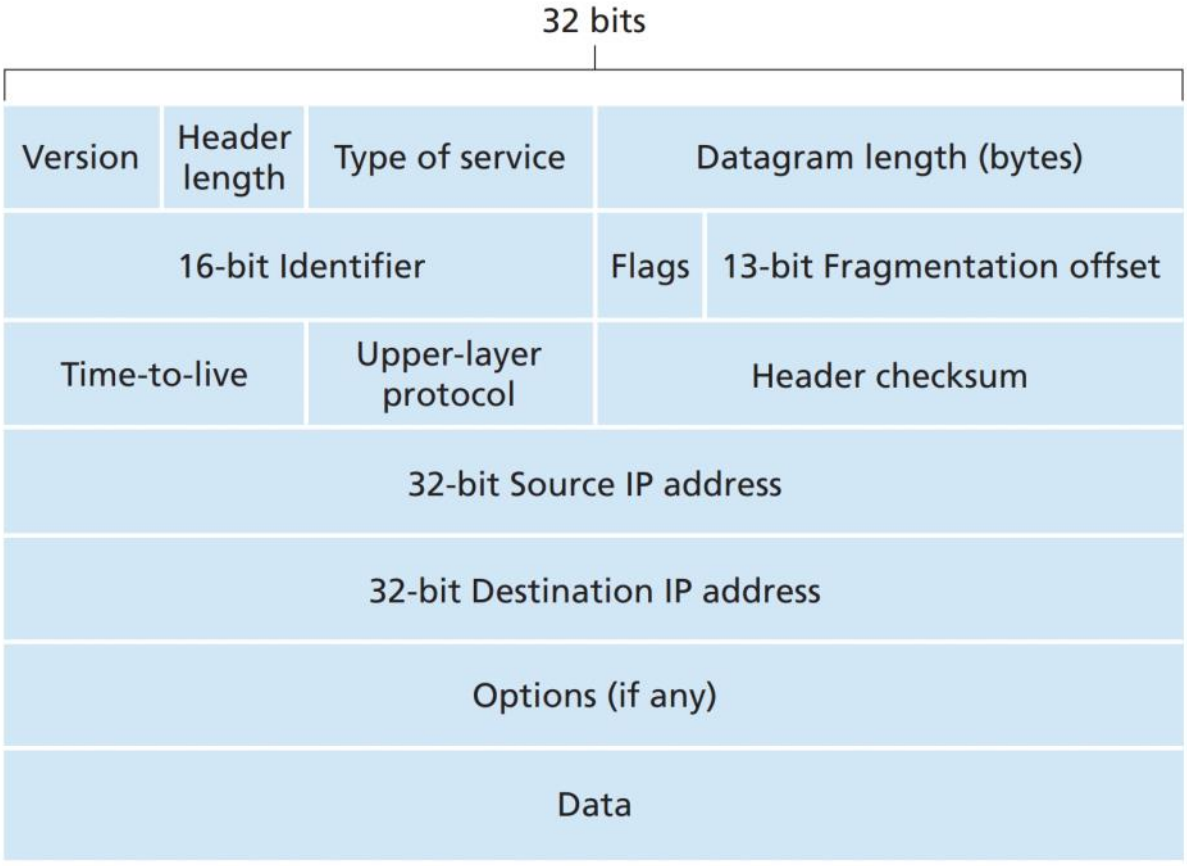


Figure 4.17 ♦ IPv4 datagram format

- Version
  - To determine how to interpret the remainder of the IP datagram
- Header length
  - Needed because an IPv4 datagram can contain a variable number of options in the header
- Type of service (TOS)
  - E.g. to distinguish real-time datagrams from non-real-time traffic
- Datagram length
  - The total length of the IP datagram (header plus data) measured in bytes
  - Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65535 bytes
  - However, datagrams are rarely larger than 1500 bytes
- Identifier, flag, fragmentation offset
  - For IP fragmentation - a large IP datagram is broken into several smaller IP datagrams which are then forwarded independently to the destination, where they are reassembled
  - IPv6 does not allow for fragmentation
- Time-to-live

    - Ensure that datagrams do not circulate forever in the network
    - Decremented by one each time the datagram is processed by a router
    - If the TTL field reaches 0, a router must drop that datagram
- Protocol
    - Indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed
    - A value of 6 indicates TCP, while a value of 17 indicates UDP
    - The protocol number is the glue that binds the network and transport layer together
- Header checksum
    - Aids a router in detecting bit errors in a received datagram
    - The checksum must be recomputed and stored again at each router, since the TTL field and possibly the options field as well will change
- Options
    - Since some datagrams may require options processing and others may not, the amount of time needed to process an IP datagram at a router can vary greatly
    - IP options were not included in the IPv6 header
- Data
    - Usually the transport-layer segment to be delivered

IPv4 Addressing

- A host typically has only a single link into the network
- The boundary between the host and the physical link is called an **interface**
- The boundary between router and any one of its links is also called an interface
- A router has multiple interfaces, one for each of its links
- The Internet Protocol requires each host and router interface to have its own IP address
- Thus, an IP address is technically associated with an interface, rather than with the host or router containing that interface
- Each IP address is 32 bits long, and there are thus a total of 2^{32} possible IP addresses
- These addresses are typically written in **dotted-decimal notation**, in which each byte of the address is written in its decimal form and is separate by a dot from other bytes
- Each interface on every host and router must have an IP address that is globally unique
- A portion of an interface's IP address will be determined by the subnet to which it is connected

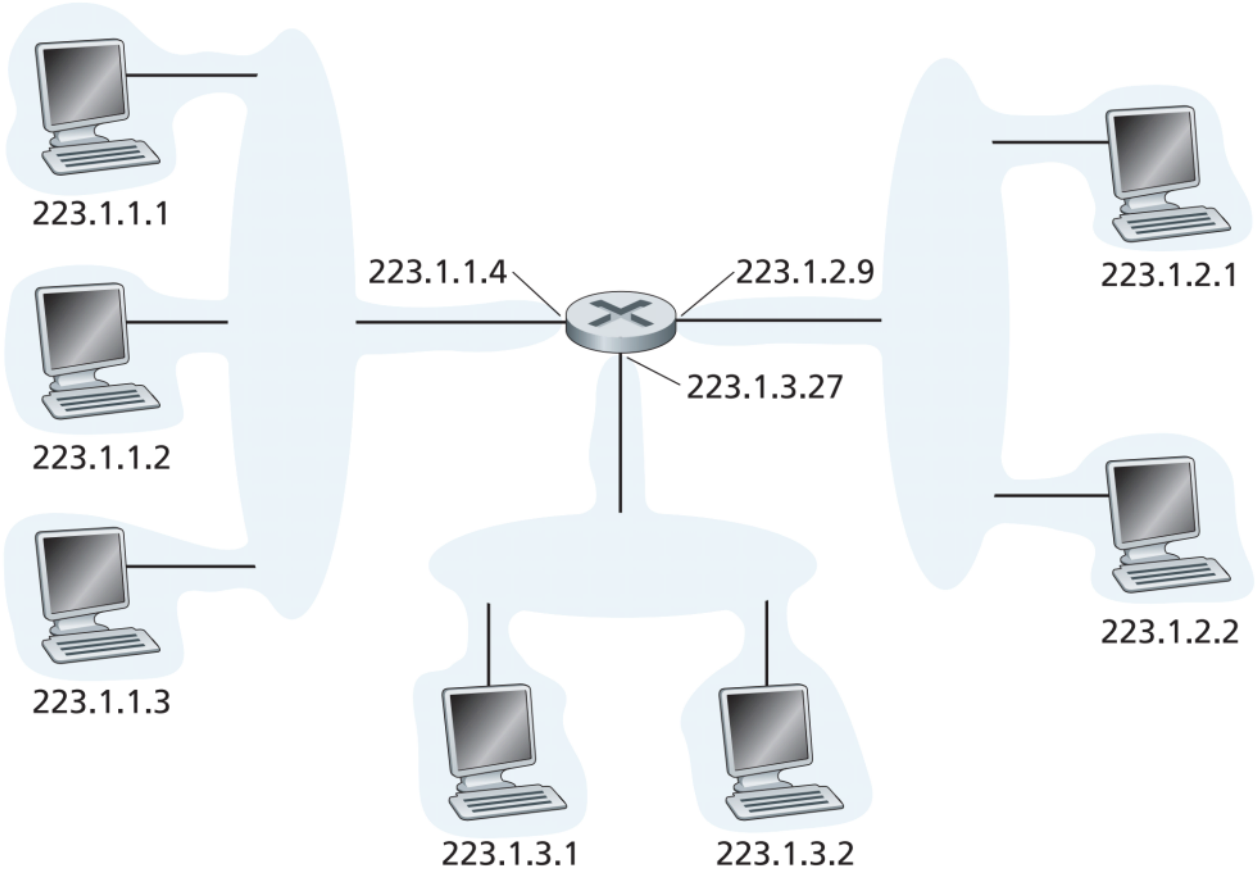


Figure 4.18 ♦ Interface addresses and subnets

Subnet

- In the figure above, the three hosts in the upper-left and the router interface to which they are connected all have the same leftmost 24 bits
- These four interfaces are also interconnected to each other by a network that contains no routers
- This network forms a **subnet** (also called an IP network or simply a network in the Internet literature)
- IP addressing assigns an address to this subnet: 234.1.1.0/24, where the /24 is known as a **subnet mask** and indicates that the leftmost 24 bits of this address define the subnet address
- Use the following recipe to define the subnets in a system:
    - To determine the subnets, detach each interface from its host or router, creating islands of isolated networks, with interfaces terminating the end points of the isolated networks. Each of these isolated networks is called a subnet
- Example: In the below diagram, 6 subnets are
    - 223.1.1.0/24
    - 223.1.2.0/24
    - 223.1.3.0/24
    - 223.1.7.0/24
    - 223.1.8.0/24
    - 223.1.9.0/24

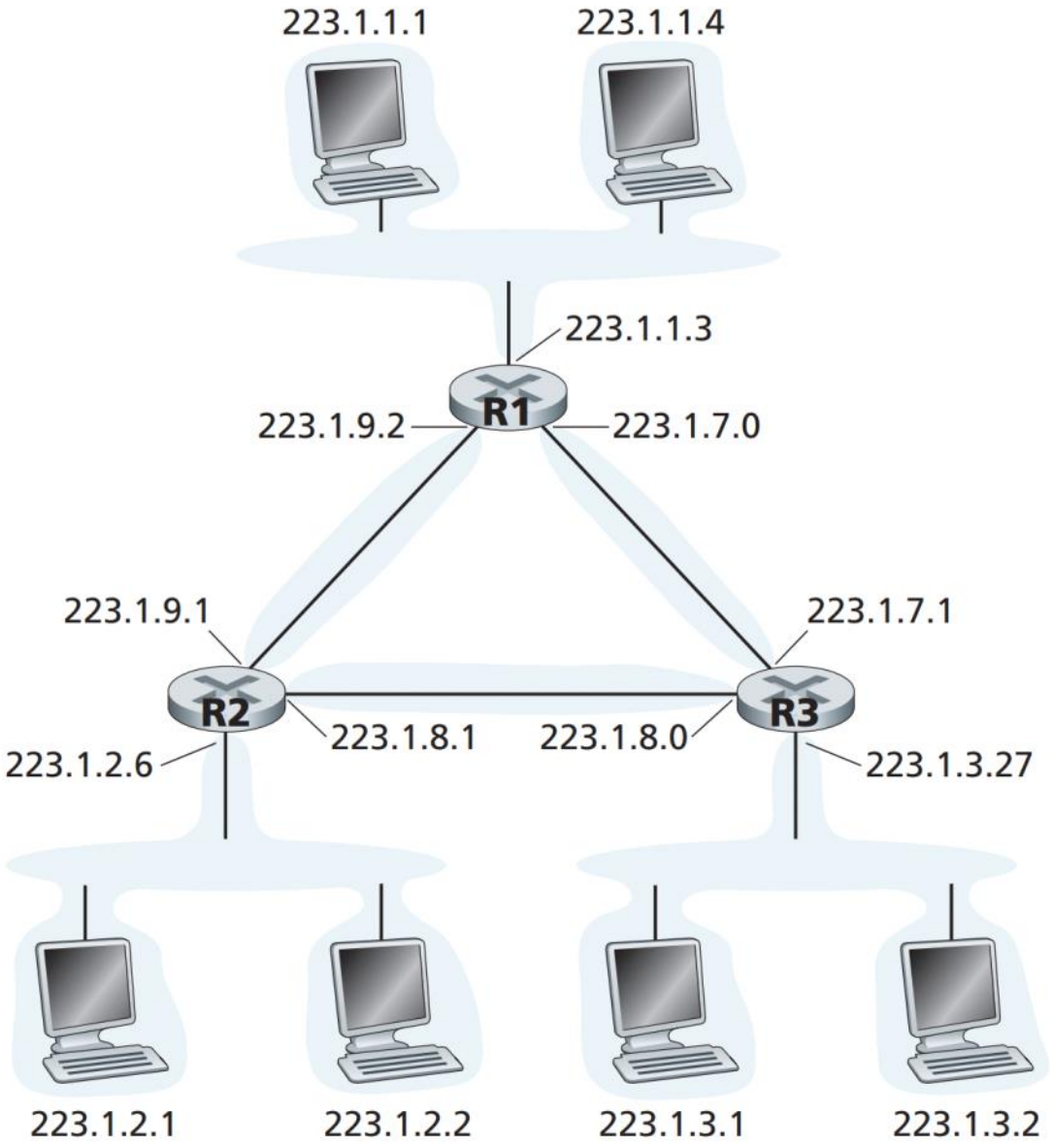


Figure 4.20 ♦ Three routers interconnecting six subnets

- The Internet's address assignment strategy is known as Classless Interdomain Routing (CIDR)
  - The 32-bit IP address is divided into two parts and has the dotted-decimal form $a.b.c.d/x$
  - The x most significant bits constitute the network portion of the IP address, and are often referred to as the prefix of the address
  - An organization is typically assigned a block of contiguous addresses - a range of addresses with a common prefix
  - The remaining bits can be thought of as distinguishing among the devices within the organization, all of which have the same network prefix
- Classful addressing - the network portions were constrained to be 8, 16, or 24 bits in length (known as class A, B and C networks, respectively)
- A class C subnet could accommodate up to $2^8 - 2 = 254$ hosts (subtract 2 because 0.0.0.0 and 255.255.255.255 are reserved), which is too small, while a class B subnet supports up

to 65634 hosts, which is too large

Obtaining a Block of Addresses

- To obtain a block of IP addresses for use within an organization's subnet, a network administrator might first contact its ISP, which would provide addresses from a larger block of addresses that had already been allocated to the ISP
- Example

ISP's block:	200.23.16.0/20	11001000 00010111 00010000 00000000
Organization 0	200.23.16.0/23	11001000 00010111 00010000 00000000
Organization 1	200.23.18.0/23	11001000 00010111 00010010 00000000
Organization 2	200.23.20.0/23	11001000 00010111 00010100 00000000
… …		…
Organization 7	200.23.30.0/23	11001000 00010111 00011110 00000000

- IP addresses are managed under the authority of the **Internet Corporation for Assigned Names and Numbers (ICANN)**, which also
  - Manages DNS root servers
  - Assigns domain names
  - Resolves domain name disputes
- The ICANN allocates addresses to regional Internet registries and handles the allocation/management of addresses within their regions

Obtaining a Host Address (DHCP)

- Host addresses can be configured manually, but typically done using the **Dynamic Host Configuration Protocol (DHCP)**
- DHCP allows a host to obtain (be allocated) an IP address automatically
- It also allows a host to learn its subnet mask, the address of its first-hop router (aka the default gateway) and the address of its local DNS server
- Because of DHCP's ability to automate the network-related aspects of connecting a host into a network, it is often referred to as a **plug-and-play** or **zeroconf** protocol
- DHCP is a client-server protocol, where a client is typically a newly arriving host
- If no server is present on the subnet, a **DHCP relay agent** (typically a router) that knows the address of a DHCP server for that network is needed
- For a newly arriving host, the DHCP protocol has four steps:
  - *DHCP server discovery*
    - Find a DHCP server with which to interact
    - Done using a **DHCP discover message**, which a client sends within a UDP packet to port 67
    - The broadcast destination IP address is 255.255.255.255
    - The source IP address is 0.0.0.0
    - The DHCP client passes the IP datagram to the link layer, which then broadcasts the frame to all nodes attached to the subnet
  - *DHCP server offer*

        - A DHCP server receiving a DHCP discover message responds to the client with a **DHCP offer message** that is broadcast to all nodes on the subnet, again using the destination IP address of 255.255.255.255 (source IP address is the server's address)
        - Each offer message contains the transaction ID of the received discovery message, the proposed IP address, the network mask and an IP **address lease time** - the amount of time for which the IP address will be valid
    - *DHCP request*
        - The client will choose from among one or more server offers
        - It responds to its selected offer with a **DHCP request message**
        - Echos back the configuration parameters
    - *DHCP ACK*
        - The server responds to the DHCP request message with a **DHCP ACK message**
        - Confirms the requested parameters
- If a client want to use its address beyond the lease's expiration, DHCP also allows a client to renew its lease on an IP address

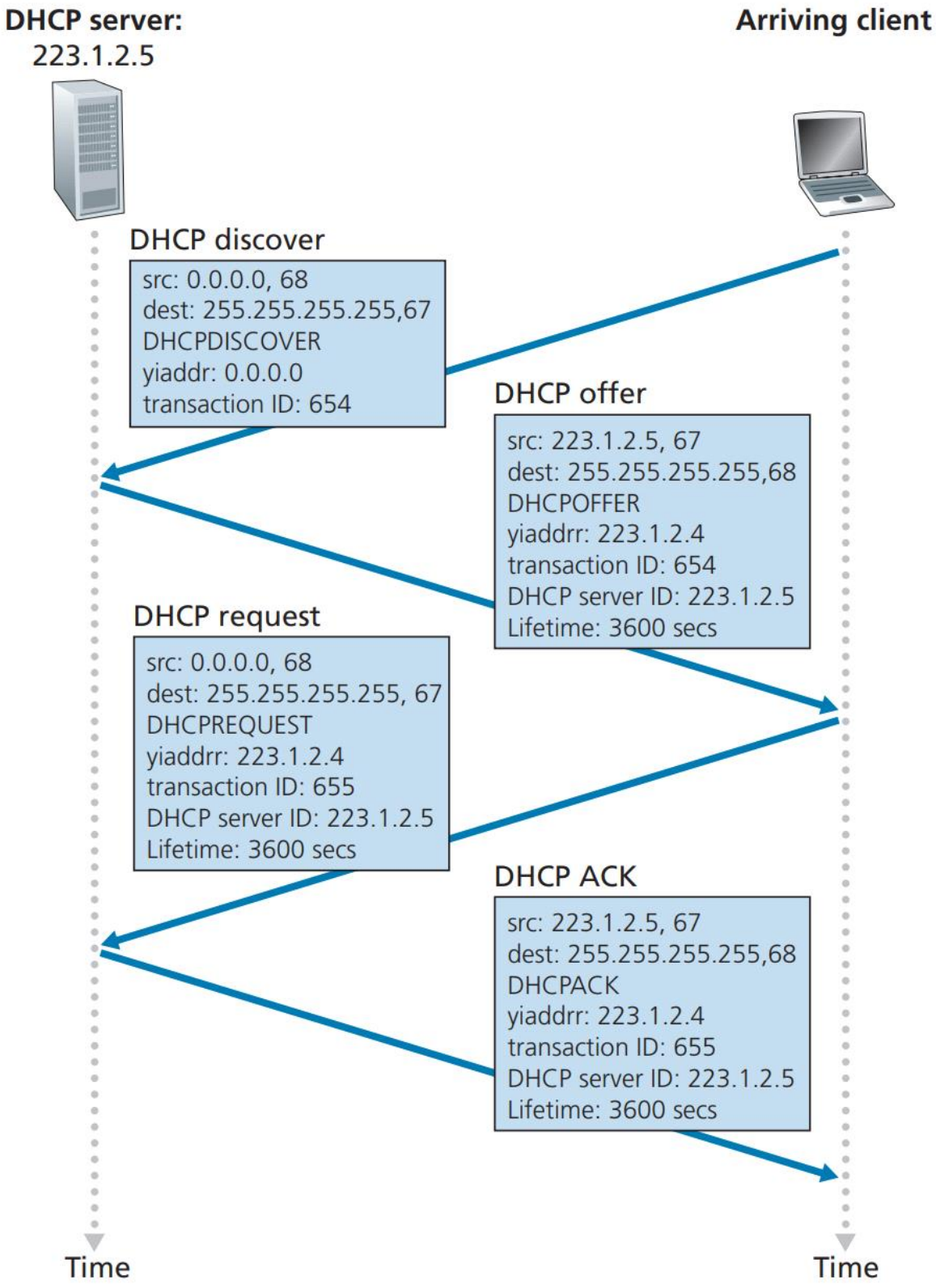


Figure 4.24 ♦ DHCP client-server interaction

Network Address Translation (NAT)

- The address space 10.0.0.0/8 is one of three portions of the IP address space that is reserved in [RFC 1918] for a **private network** or a realm with private addresses
- A **realm with private addresses** refers to a network whose addresses only have meaning to

devices within that network

- The NAT-enabled router behaves to the outside world as a single device with a single IP address
- The NAT-enabled router hides the details of the home network from the outside world
- The router gets its address from the ISP's DHCP server, and the router runs a DHCP server to provide addresses to computers within the NAT-DHCP-router-controlled home network's address space
- Concerns about NAT
    - Port numbers are meant to be used for addressing processes, not for addressing hosts
    - NAT violates this principle that hosts should be talking directly with each other, without interfering nodes modifying IP addresses, much less port numbers
- How can one peer connect to another peer that is behind a NAT server, and has DHCP-provided NAT address?
    - Solution includes **NAT traversal** tools [RFC 5389]
- Example:

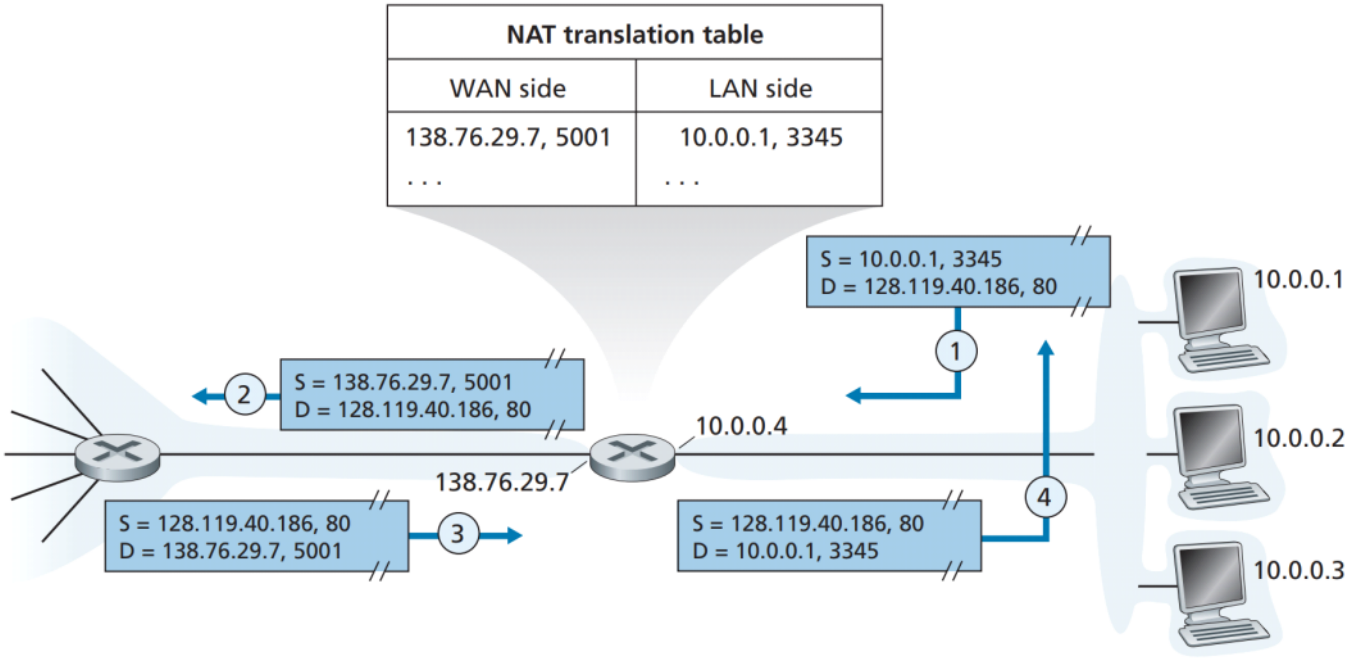


Figure 4.25 ♦ Network address translation

IPv6

- Motivation: 32-bit IPv4 address space was beginning to be used up

IPv6 Datagram Format

- The most important changes introduced in IPv6 are:
    - Expanded addressing capacities
        - The size of the IP address was increased to 128 bits
        - This ensures that the world won't run out of IP addresses
        - Also introduced a new type of address, called an **anycast address**, that allows a datagram to be delivered to any one of a group of hosts
    - A streamlined 40-byte header

        - A number of IPv4 fields have been dropped or made optional
        - The resulting 40-bytes fixed-length header allows for faster processing of an IP datagram by a router
    - Flow labeling
        - RFC 2460 states that this allows "labeling of packets belonging to particular flows for which the sender requests special handling, such as a non-default quality of service or real-time service"
        - Audio and video transmission might likely be treated as a flow
        - The more traditional applications like file transfer and email might NOT be treated as flows
        - The traffic carried by a high-priority user (someone paying for better service) might also be treated as a flow
- Similar, more streamlined structure of the IPv6 datagram fields:
    - Version
    - Traffic class
        - Like the TOS field in IPv4, this can be used to give priority to certain datagrams within a flow, or it can be used to give priority to datagrams from certain applications
    - Flow label
        - Flow identifier
    - Payload length
        - The number of bytes in the datagram following the fixed-length datagram header
    - Next header
        - Identifies the protocol to which the content (data field) will be delivered (TCP or UDP)
    - Hop limit
        - The value is decremented by one by each router that forwards the datagram
        - If the hop limit count reaches zero, a router must discard the datagram
    - Source and destination addresses
        - 128-bit addresses
    - Data
        - Payload portion
        - When the datagram reaches its destination, the payload will be removed from the datagram and passed on to the protocol specified in the next header field
- Several fields appearing in an IPv4 datagram that are no longer present in an IPv6 datagram
    - Fragmentation/reassembly
        - IPv6 does not allow for fragmentation and reassembly at intermediate routers
        - These operations can be performed only by the source and destination
        - If a router receives an IPv6 datagram that is too large, it will simply drop the datagram and send a "Packet Too Big" ICMP error message back to the sender
        - This speeds up IP forwarding within the network
    - Header checksum
        - Because the transport-layer and link-layer protocols in the Internet layers perform checksumming, the designers of IP removed this functionality for faster processing of IP packets
        - Since IPv4 header contains a TTL field, the header checksum needed to be recomputed at every router, which was very costly
    - Options
        - An options field is no longer a part of the standard IP header

- However, it has not gone away
- The options field is one of the possible next headers pointed to from within the IPv6 header
- Just as TCP or UDP protocol headers can be the next header within an IP packet, so too can an options field
- The removal of this field results in a fixed-length, 40-byte IP header

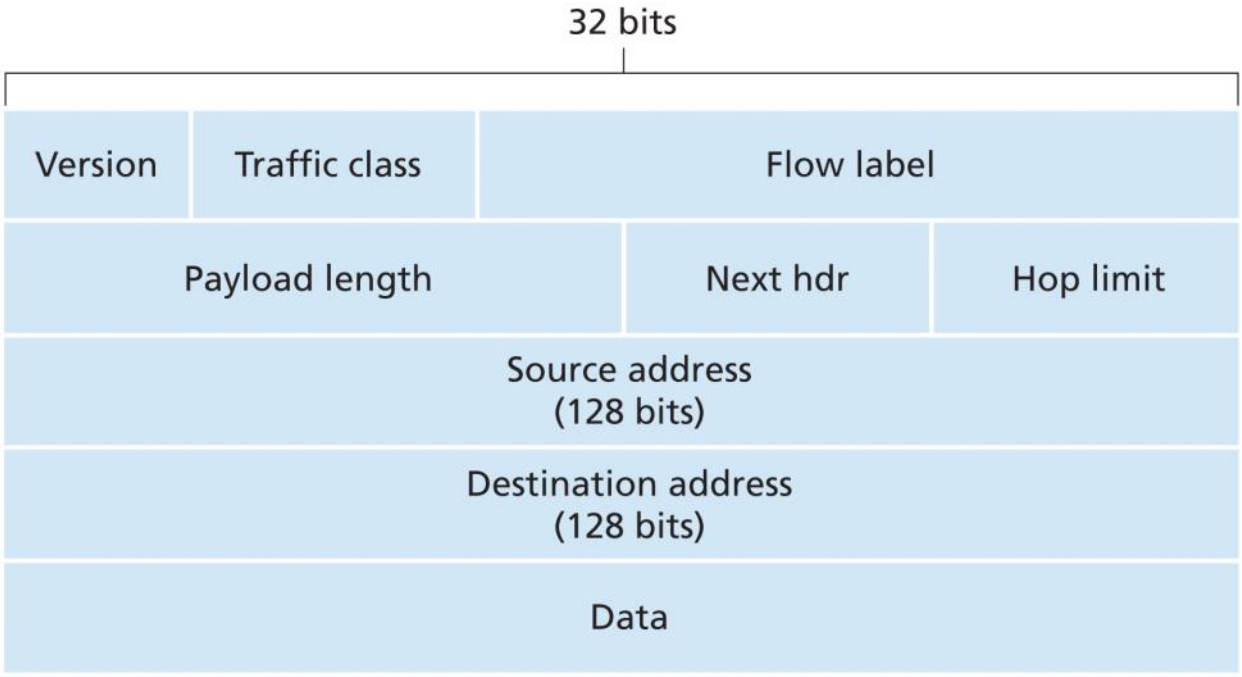


Figure 4.26 ♦ IPv6 datagram format

Transitioning from IPv4 to IPv6

- The approach to IPv4-to-IPv6 transition that has been most widely adopted in practice involves **tunneling**
- Suppose two IPv6 nodes want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers
- We refer to the set of IPv4 routers between two IPv6 routers as a **tunnel**
- The IPv6 node on the sending side of the tunnel (B) puts entire IPv6 datagram in the data (payload) field of an IPv4 datagram
- This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel (E) and sent to the first node in the tunnel (C)
- The intervening IPv4 routers (C and D) route this IPv4 datagram among themselves, just as they would for any other datagram
- The IPv6 node on the receiving side of the tunnel (E) eventually receives the IPv4 datagram from node D
- The IPv6 node E determines that the IPv4 datagram contains an IPv6 datagram (by observing that the protocol number field in the IPv4 datagram is 41 [RFC 4213], indicating that the IPv4 payload is a IPv6 datagram)
- The IPv6 node E extracts the IPv6 datagram and routes the IPv6 datagram exactly as if the IPv6 datagram was from a directly connected IPv6 neighbor

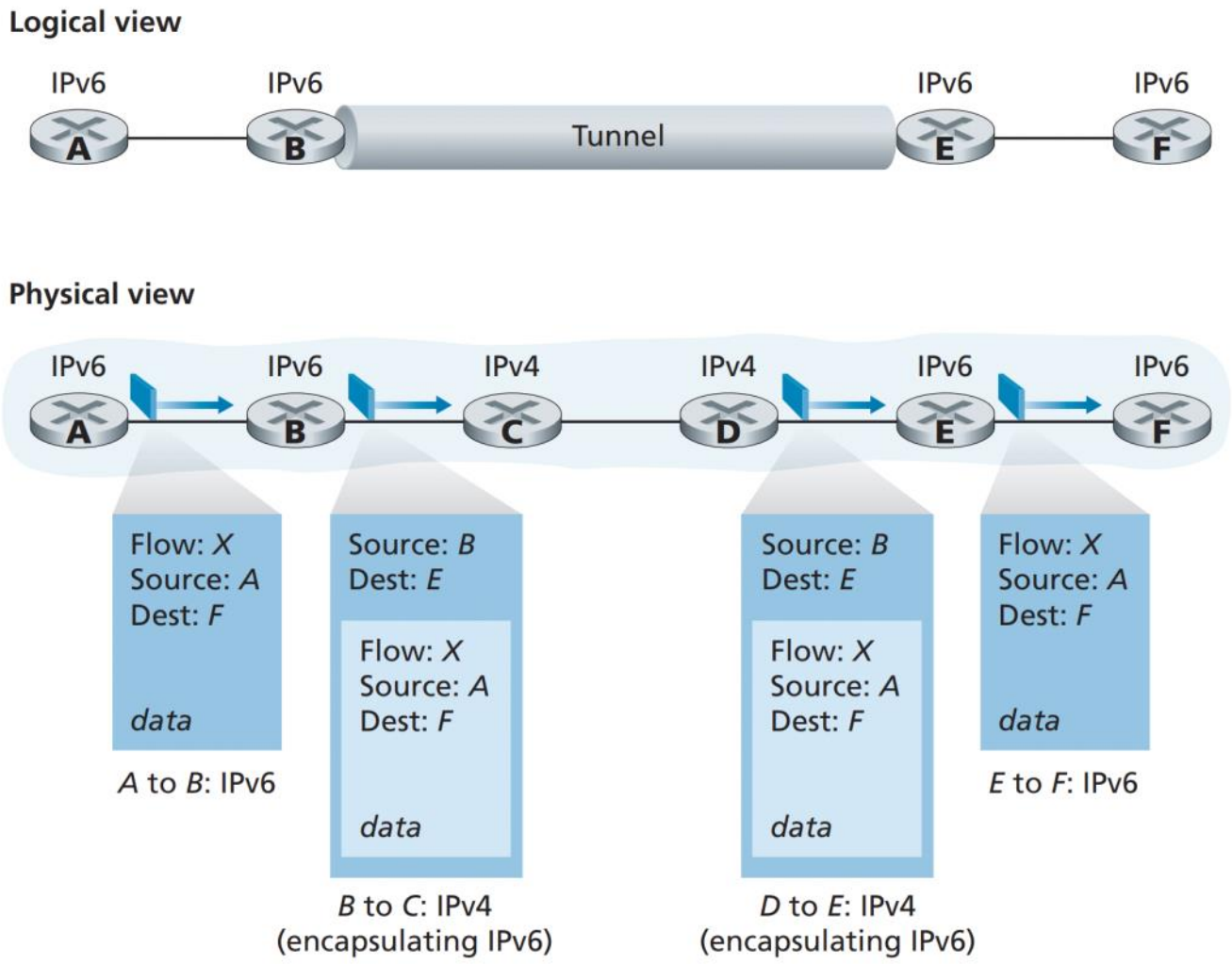


Figure 4.27 ♦ Tunneling

Chapter 4.4 - Generalized Forwarding and SDN

Sunday, July 9, 2023

Overview

- A more general "match-plus-action" paradigm
- Match
    - Made over multiple header fields associated with different protocols at different layers in the protocol stack
- Action
    - Forwarding the packet to output port(s)
    - Load balancing packets
    - Rewriting header values (in NAT)
    - Blocking/dropping a packet (in a firewall)
    - Sending a packet to a special server for further processing and action (in DPI)
- A match-plus-action table generalizes the forwarding table

OpenFlow

- A highly visible standard that has pioneered the notion of the match-plus-action forwarding abstraction and controllers
- Each entry in the match-plus-action forwarding table, known as a **flow table** in OpenFlow, includes:
    - A set of header field values
        - In the case of destination-based forwarding, hardware-based matching is most rapidly performed in TCAM memory
        - A packet that matches no entry can be dropped or sent to the remote controller for more processing
    - A set of counters
        - These counters might include the number of packets that have been matched by that table entry
        - Also the time since the table entry was last updated
    - A set of actions to be taken
        - Forward the packet to a given output port
        - Drop the packet
        - Make copies of the packet and send them to multiple output ports
        - Rewrite selected header fields
        - …

Match

- OpenFlow's match abstraction allows for a match to be made on selected fields from three layers of protocol headers
- Imagine that the source and destination MAC addresses are the link-layer addresses associated with the frame's sending and receiving interfaces
- By forwarding on the basis of Ethernet addresses rather than IP addresses, an OpenFlow-enabled device can equally perform as a router forwarding datagrams as well as a switch forwarding frames
- The **ingress port** is the input port at the packet switch on which a packet is received
- Flow table entries may also have **wildcard**
    - Example: an IP address of 128.119.*.* in a flow table will match the corresponding

address field of any datagram that has 128.119 as the first 16 bits of its address

- Not all fields in an IP header can be matched
  - Example: OpenFlow does not allow matching on the basis of TTL field or datagram length field
  - Provide for enough functionality to accomplish a task, without over-burdening the abstraction with so much detail and generality that it becomes bloated and unusable

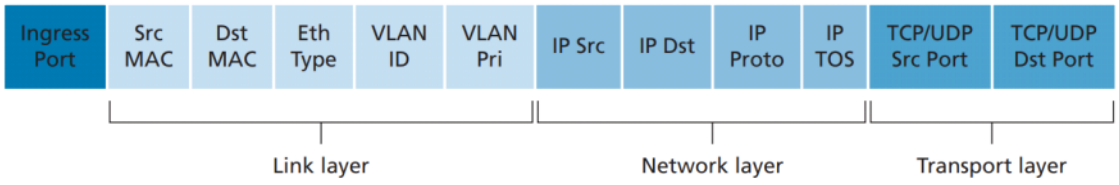


Figure 4.29 ♦ Packet matching fields, OpenFlow 1.0 flow table

Action

- Each flow table entry has a list of zero or more actions that determine the processing that is to be applied to a packet that matches a flow table entry
- If there are multiple actions, they are performed in the order specified in the list
- Forwarding
  - An incoming packet may be forwarded to a particular physical output port, broadcast over all ports, or multicast over a selected set of ports
  - The packet may be encapsulated and sent to the remote controller for this device
  - The controller may install new flow table entries, and may return the packet to the device for forwarding under the updated set of flow table rules
- Dropping
  - A flow table entry with no action indicates that a matched packet should be dropped
- Modify-field
  - The values in 10 packet-header fields may be rewritten before the packet is forwarded to the chosen output port

OpenFlow Examples of Match-plus-Action

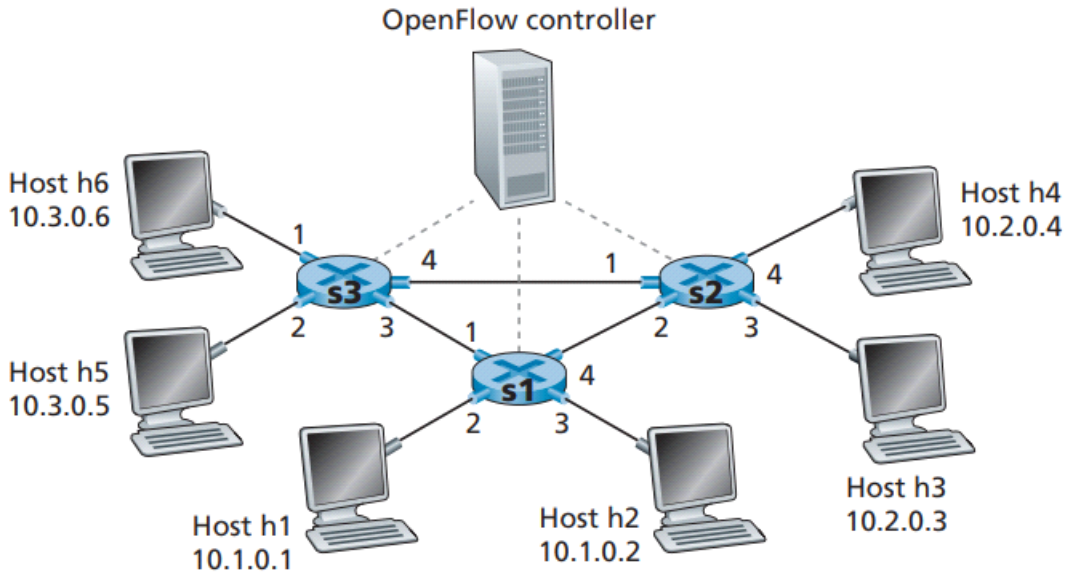


Figure 4.30 ♦ OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Example 1: Simple Forwarding

- Packets from h5 or h6 destined to h3 or h4 are to be forwarded from s3 to s1, and then from s1 to s2

s3 Flow Table (Example 1)	
Match	Action
IP Src = 10.3.*.* ; IP Dst = 10.2.*.*	Forward(3)
...	...

s1 Flow Table (Example 1)	
Match	Action
Ingress Port = 1 ; IP Src = 10.3.*.* ; IP Dst = 10.2.*.*	Forward(4)
...	...

s2 Flow Table (Example 1)	
Match	Action
Ingress port = 2 ; IP Dst = 10.2.0.3	Forward(3)
Ingress port = 2 ; IP Dst = 10.2.0.4	Forward(4)
...	...

Example 2: Load Balancing

- Datagrams from h3 destined to 10.1.*.* are to be forwarded over the direct link between s2 and s1
- Datagrams from h4 destined to 10.1.*.* are to be forwarded over the direct link between s2 and s3 (and then from s3 to s1)

s2 Flow Table (Example 2)	
Match	Action
Ingress port = 3; IP Dst = 10.1.*.*	Forward(2)
Ingress port = 4; IP Dst = 10.1.*.*	Forward(1)
...	...

Example 3: Firewalling

- s2 wants only to receive traffic sent from hosts attached to s3
- If there are no other entries in s2's flow table, then only traffic from 10.3.*.* would be forwarded to the hosts attached to s2

s2 Flow Table (Example 3)	
Match	Action
IP Src = 10.3.*.* IP Dst = 10.2.0.3	Forward(3)
IP Src = 10.3.*.* IP Dst = 10.2.0.4	Forward(4)
...	...

Epilogue

- These flow tables are actually a limited form of *programmability*, specifying how a router should forward and manipulate a datagram, based on the match between the datagram's header and the conditions.
- One could imagine an even richer form of programmability - a programming language with higher-level constructs
- P4 (Programming Protocol-independent Packet Processors) is such a language

Chapter 4 Self Test

Sunday, August 6, 2023

Chapter 4.1 - Overview of Network Layer

1. List three differences between forwarding and routing.
2. What does a forwarding table do?
3. In a traditional approach, how are forwarding tables computed?
4. In an SDN approach, how are forwarding computed?
5. What is a similarity between these two approaches?
6. What is a difference between these two approaches?

Chapter 4.2 - Router

1. What are the functions of input ports related to
    a. Physical layer?
    b. Link layer?
    c. Network layer?
2. What does a switching fabric do?
3. What does the routing processor do in
    a. Traditional routers?
    b. SDN routers?
4. Why is the datagram-processing pipeline usually implemented in hardware?
5. Why can forwarding decisions be made locally at each input port without invoking the centralized routing processor?
6. In destination-based forwarding, what happens if a prefix does not match any of the entries?
7. How can lookup be achieved in constant time? What kind of memory do we use?
8. Give three examples of match plus action.
9. Briefly describe the following switching methods:
    a. Switching via memory
    b. Switching via a bus
    c. Switching via an interconnection network (crossbar)
10. For each switching method, can two datagrams be transferred at the same time? Why?
11. Will there be input queueing if the switching fabric transfer rate is N times faster than input/output line speeds (transmission rates)?
12. Describe head-of-the-line (HOL) blocking in an input-queued switch.
13. Will there be output queueing if the switching fabric transfer rate is N times faster than input/output line speeds (transmission rates)?
14. What is the drop-tail policy?
15. What are active queue management (AQM) algorithms? Give an example.
16. According to the rule of thumb, what should be the amount of buffering for a 10-Gbps link with an RTT of 250 msec?
17. What are the advantages and disadvantages of large buffers?
18. Describe the following scheduling methods
    a. FIFO
    b. Priority queueing
    c. Round robin

d. Weighted fair queueing

Chapter 4.3 - The Internet Protocol (IP)

1. Why is header length field needed in IPv4?
2. What is one use case of the type of service field?
3. For IP fragmentation, where are the fragments reassembled?
4. Is there fragmentation in IPv6?
5. What does the router do when the TTL field reaches zero?
6. Why does the checksum need to be recomputed at each router?
7. Can a router have multiple IP addresses? Why or why not?
8. How many possible IPv4 addresses are there?
9. How is a subnet defined?
10. Which organization manages IP addresses?
11. List four things that DHCP allows a host to learn.
12. What are the four steps a new arriving host must perform to obtain an IP address? What is the source and destination IP addresses used? What are the contents?
13. What is an IP address lease time?
14. How is a private address different from an ordinary IP address?
15. How does a NAT-enabled router get its address?
16. What does a NAT translation table contain?
17. How does a NAT-enabled router provide addresses to devices within the private network?
18. Which IPv4 header field does the traffic class field in IPv6 correspond to?
19. Which IPv4 header field does the next header field in IPv6 correspond to?
20. Which IPv4 header field does the hop limit field in IPv6 correspond to?
21. In IPv6, what happens if a router receives an IPv6 datagram that is too large? What message does it send back to the sender?
22. What fields are no longer in Ipv6?

Chapter 4.4 - Generalized Forwarding and SDN

1. What are some of the actions that can be performed in an SDN?
2. Can all fields in an IP header be matched? If not, what are some fields that cannot be matched?
3. What does it mean if a flow table entry has no action for a matched packet?

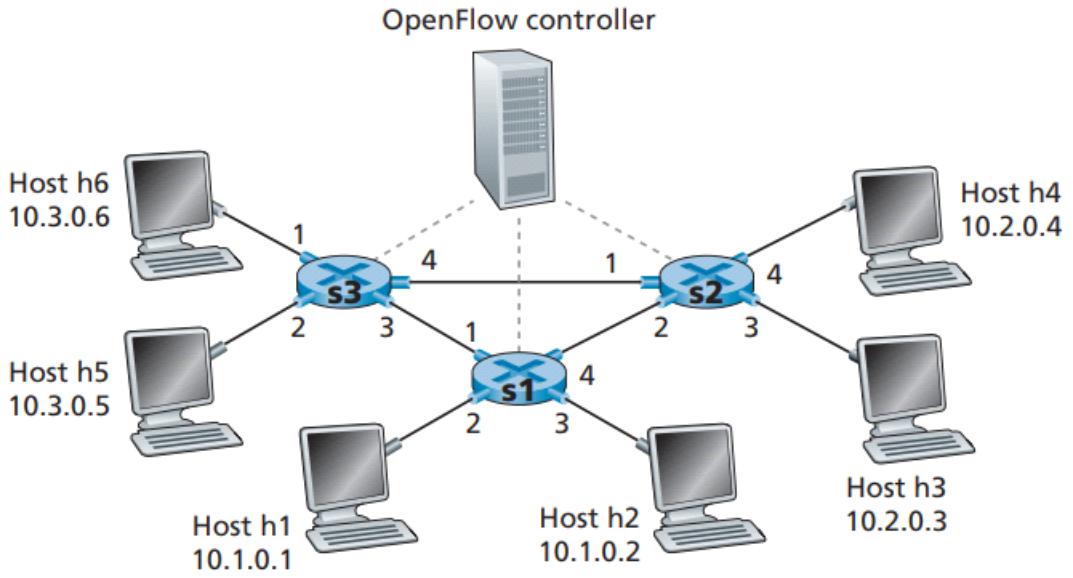


Figure 4.30 ♦ OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

4. Give the flow tables for the rule "packets from h5 or h6 destined to h3 or h4 are to be forwarded from s3 to s1, and then from s1 to s2"
5. Give the flow tables for the rules
    - Datagrams from h3 destined to 10.1.*.* are to be forwarded over the direct link between s2 and s1
    - Datagrams from h4 destined to 10.1.*.* are to be forwarded over the direct link between s2 and s3 (and then from s3 to s1)
6. Give the flow tables for the rules
    - s2 wants only to receive traffic sent from hosts attached to s3
    - If there are no other entries in s2's flow table, then only traffic from 10.3.*.* would be forwarded to the hosts attached to s2
7. What is P4?

Chapter 4 Review Problems

Wednesday, July 26, 2023 8:19 PM

4.6 Summary

In this chapter, we've covered the **data plane** functions of the network layer—the *per-router* functions that determine how packets arriving on one of a router's input links are forwarded to one of that router's output links. We began by taking a detailed look at the internal operations of a router, studying input and output port functionality and destination-based forwarding, a router's internal switching mechanism, packet queue management and more. We covered both traditional IP forwarding (where forwarding is based on a datagram's destination address) and generalized forwarding (where forwarding and other functions may be performed using values in several different fields in the datagram's header) and seen the versatility of the latter approach. We also studied the IPv4 and IPv6 protocols in detail, and Internet addressing, which we found to be much deeper, subtler, and more interesting than we might have expected. We completed our study of the network-layer data plane with a study of middleboxes, and a broad discussion of Internet architecture.

With our newfound understanding of the network-layer's data plane, we're now ready to dive into the network layer's control plane in Chapter 5!

Homework Problems and Questions

Chapter 4 Review Questions

SECTION 4.1

R1. Let's review some of the terminology used in this textbook. Recall that the name of a transport-layer packet is *segment* and that the name of a link-layer packet is *frame*. What is the name of a network-layer packet? Recall that both routers and link-layer switches are called *packet switches*. What is the fundamental difference between a router and link-layer switch?

R2. We noted that network layer functionality can be broadly divided into data plane functionality and control plane functionality. What are the main functions of the data plane? Of the control plane?

R3. We made a distinction between the forwarding function and the routing function performed in the network layer. What are the key differences between routing and forwarding?

R4. What is the role of the forwarding table within a router?

R5. We said that a network layer's service model "defines the characteristics of end-to-end transport of packets between sending and receiving hosts." What is the service model of the Internet's network layer? What guarantees are made by the Internet's service model regarding the host-to-host delivery of datagrams?

SECTION 4.2

R6. In Section 4.2, we saw that a router typically consists of input ports, output ports, a switching fabric and a routing processor. Which of these are implemented in

R1
- Network-layer packets are called datagrams
- Routers use IP addresses for forwarding
- Link-layer switches use MAC addresses for forwarding

R2
- Data plane - forwarding
- Control plane - routing (compute forwarding tables)

R3
- Routing is global, takes longer time (seconds)
- Forwarding is local, takes shorter time (nanoseconds)

R4
- We can use the forwarding table to look up and match an IP address, then forwards a packet onto the correct output port from which it will be sent to the next router

R5
- No reliable transfer, correctness guarantee, maximum delay, etc.
- "Best effort"

R6
- Input ports, output ports and switching fabrics are implemented in hardware and the routing processor is implemented in the software
- The data plane is implemented in hardware (for forwarding table lookup) and the control plane is implemented in software (for running routing algorithms?)

hardware and which are implemented in software? Why? Returning to the notion of the network layer's data plane and control plane, which are implemented in hardware and which are implemented in software? Why?

R7. How can the input ports of a high-speed router facilitate fast forwarding decisions?

R8. What is meant by destination-based forwarding? How does this differ from generalized forwarding (assuming you've read Section 4.4, which of the two approaches are adopted by Software-Defined Networking)?

R9. Suppose that an arriving packet matches two or more entries in a router's forwarding table. With traditional destination-based forwarding, what rule does a router apply to determine which of these rules should be applied to determine the output port to which the arriving packet should be switched?

R10. Switching in a router forwards data from an input port to an output port. What is the advantage of switching via an interconnection network over switching via memory and switching via bus?

R11. What is the role of a *packet scheduler* at the output port of a router?

R12. a. What is a drop-tail policy?

b. What are AQM algorithms?

c. Name one of the most widely studied and implemented AQM algorithms and explain how it works.

R13. What is HOL blocking? Does it occur in input ports or output ports?

R14. In Section 4.2, we studied FIFO, Priority, Round Robin (RR), and Weighted Fair Queuing (WFQ) packet scheduling disciplines? Which of these queuing disciplines ensure that all packets depart in the order in which they arrived?

R15. Give an example showing why a network operator might want one class of packets to be given priority over another class of packets.

R16. What is an essential different between RR and WFQ packet scheduling? Is there a case (*Hint:* Consider the WFQ weights) where RR and WFQ will behave exactly the same?

SECTION 4.3

R17. Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to some other upper-layer protocol?

R18. What field in the IP header can be used to ensure that a packet is forwarded through no more than *N* routers?

R19. Recall that we saw the Internet checksum being used in both transport-layer segment (in UDP and TCP headers, Figures 3.7 and 3.29 respectively) and in network-layer datagrams (IP header, Figure 4.17). Now consider a transport

layer segment encapsulated in an IP datagram. Are the checksums in the segment header and datagram header computed over any common bytes in the IP datagram? Explain your answer.

R20. When a large datagram is fragmented into multiple smaller datagrams, where are these smaller datagrams reassembled into a single larger datagram?

R21. How many IP addresses does a router have?

R22. What is the 32-bit binary equivalent of the IP address 202.3.14.25?

R23. Visit a host that uses DHCP to obtain its IP address, network mask, default router, and IP address of its local DNS server. List these values.

R24. Suppose there are four routers between a source host and a destination host. Ignoring fragmentation, an IP datagram sent from the source host to the destination host will travel over how many interfaces? How many forwarding tables will be indexed to move the datagram from the source to the destination?

R25. Suppose an application generates chunks of 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

R26. Suppose you purchase a wireless router and connect it to your cable modem. Also suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?

R27. What is meant by the term "route aggregation"? Why is it useful for a router to perform route aggregation?

R28. What is meant by a "plug-and-play" or "zeroconf" protocol?

R29. What is a private network address? Should a datagram with a private network address ever be present in the larger public Internet? Explain.

R30. Compare and contrast the IPv4 and the IPv6 header fields. Do they have any fields in common?

R31. It has been said that when IPv6 tunnels through IPv4 routers, IPv6 treats the IPv4 tunnels as link-layer protocols. Do you agree with this statement? Why or why not?

SECTION 4.4

R32. How does generalized forwarding differ from destination-based forwarding?

R33. What is the difference between a forwarding table that we encountered in destination-based forwarding in Section 4.1 and OpenFlow's flow table that we encountered in Section 4.4?

R7
- ~~Using TCAM?~~
- With a shadow copy, the forwarding lookup is made locally, at each input port, without invoking the centralized routing processor

R8
- ~~Destination forwarding only performs IP matching, while generalized forwarding can perform matching for header fields in the transport, network and link layers~~
- The SDN uses generalized forwarding
- Destination forwarding means that a datagram arriving at a router will be forwarded to an output interface based only on the final destination

R9
- The router uses the longest prefix matching rule?

R10
- Multiple packets can be forwarded at the same time (as long as they have different destination address)
- An interconnection network can forward packets in parallel as long as all the packets are being forwarded to different output ports

R11
- It selects a packet from the output queue and puts the packet on the link one at a time

R12
a. ~~The last packet in the queue gets dropped~~

a. Drop-tail: drop the arriving packet
b. AQM: active queue management; proactive packet-dropping and -marking policies
c. Random Early Detection (RED) is one of the most widely studied and implemented AQM algorithms

R13
- HOL blocking occurs in input ports when a packet cannot be forwarded because another packet is being forwarded, although they might not have the same output port

R14
- FIFO

R15
- A packet carrying network management information should receive priority over regular user traffic
- A real-time voice-over-IP packet might need to receive priority over non-real-time traffic such as email

R16
- In WFQ, the service time depends on the weight - the higher the weight, the longer the service time
- RR and WFQ are the same when all weights are the same?

R17
- It can use the TOS (or protocol?) field in the header
- The protocol field in the IP datagram specifies the transport layer protocol

R18
- Time-to-live (TTL) field

R19
- IP header checksum only computes the checksum of an IP packet's IP header fields
- It shares no common bytes with the IP datagram's transport-layer segment part

R20
- They are reassembled at the destination host

R21
- A router has an interface for each of its interfaces

R22
- No...

R23
- No...

R24
- ~~2 interfaces from the hosts and 8 interfaces in the routers~~
- ~~4 forwarding tables will be traversed?~~
- 8 interfaces
- 3 forwarding tables

R25
- 40/80=50%

R26
- Each PC can use DHCP to get an IP address. It can also be configured manually?

R28
- It means there is no need for a network administrator to manually configure the settings in order to connect the host into a network
- The protocol is able to automatically configure a host's network-related aspects in order to connect the host into a network

R29
- A private network address is only visible in a subnet?
- A private network address is only meaningful to devices within the network
- A datagram with a private network address should never be present in the larger public Internet, since the address is potentially used by many network devices in their own private networks

R30
- Source and destination addresses, TOS, TTL
- Payload length, next header

R31
- Yes, because in tunneling, IPv6 datagrams are encapsulated in an IPv4 datagram (the outermost header is from IPv4)

R33. What is the difference between a forwarding table that we encountered in destination-based forwarding in Section 4.1 and OpenFlow's flow table that we encountered in Section 4.4?

R34. What is meant by the "match plus action" operation of a router or switch? In the case of destination-based forwarding packet switch, what is matched and what is the action taken? In the case of an SDN, name three fields that can be matched, and three actions that can be taken.

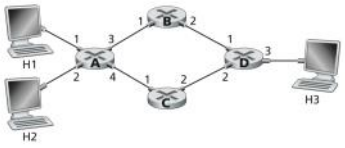
R35. Name three header fields in an IP datagram that can be "matched" in OpenFlow 1.0 generalized forwarding. What are three IP datagram header fields that *cannot* be "matched" in OpenFlow?

Problems

P1. Consider the network below.

a. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.

b. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (*Hint:* This is a trick question.)



P2. Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router.

a. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a shared bus?

b. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses switching via memory?

c. Suppose the two packets are to be forwarded to the same output port. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a crossbar?

R31
- Yes, because in tunneling, IPv6 datagrams are encapsulated in an IPv4 datagram (the outermost header is from IPv4)

R32
- Already discussed

R33
- A destination-based forwarding table only has IP addresses and an output port
- A flow table can have variable numbers of fields, also a set of counters and a set of actions

R34
- In a destination-based forwarding, IP addresses are matched and the action is forwarding to a specific port
- In an SDN, three fields that can be matched are source IP, destination IP and source MAC
- In an SDN, three actions that can be taken are forwarding, dropping and further processing (e.g. recomputing the forwarding table)
- "Match plus action" means that a router or switch tries to find a match between header values of a packet with some entry in a flow table, and perform operations on the packet based on the match

R35
- Can be matched: source IP, destination IP and source port number
- ~~Cannot be matched: source MAC, destination MAC and data link type~~
- Cannot be matched: TTL field, datagram length field and header checksum

Chapter 4 Practice Problems

Thursday, August 3, 2023

R34. What is meant by the "match plus action" operation of a router or switch? In the case of destination-based forwarding packet switch, what is matched and what is the action taken? In the case of an SDN, name three fields that can be matched, and three actions that can be taken.

R35. Name three header fields in an IP datagram that can be "matched" in Open-Flow 1.0 generalized forwarding. What are three IP datagram header fields that *cannot* be "matched" in OpenFlow?

Problems

P1. Consider the network below.

a. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.

b. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (*Hint:* This is a trick question.)

P2. Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router.

a. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a shared bus?

b. Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses switching via memory?

c. Suppose the two packets are to be forwarded to the same output port. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a crossbar?

P1.

a.

Destination IP	Interface
H3	3

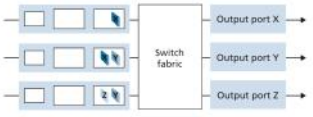
a. Not possible

P2.

a. No
b. No
c. ~~Yes~~ No because the two packets are sent to the same output at the same time

P3. In Section 4.2.4, it was said that if R_switch is N times faster than R_line, then only negligible queuing will occur at the input ports, even if all the packets are to be forwarded to the same output port. Now suppose that $R\_switch = R\_line$, but all packets are to be forwarded to different output ports. Let D be the time to transmit a packet. As a function of D, what is the maximum input queuing delay for a packet for the (a) memory, (b) bus, and (c) crossbar switching fabrics?

P4. Consider the switch shown below. Suppose that all datagrams have the same fixed length, that the switch operates in a slotted, synchronous manner, and that in one time slot a datagram can be transferred from an input port to an output port. The switch fabric is a crossbar so that at most one datagram can be transferred to a given output port in a time slot, but different output ports can receive datagrams from different input ports in a single time slot. What is the minimal number of time slots needed to transfer the packets shown from input ports to their output ports, assuming any input queue scheduling order you want (i.e., it need not have HOL blocking)? What is the largest number of slots needed, assuming the worst-case scheduling order you can devise, assuming that a non-empty input queue is never idle?



P5. Suppose that the WEQ scheduling policy is applied to a buffer that supports three classes, and suppose the weights are 0.5, 0.25, and 0.25 for the three classes.

a. Suppose that each class has a large number of packets in the buffer. In what sequence might the three classes be served in order to achieve the WFQ weights? (For round robin scheduling, a natural sequence is 123123123 . . .).

b. Suppose that classes 1 and 2 have a large number of packets in the buffer, and there are no class 3 packets in the buffer. In what sequence might the three classes be served in to achieve the WFQ weights?

? P3.

a. ~~D~~ *(n-1)D*
b. ~~D~~ *(n-1)D*
c. 0

? P4.

Minimum number of time slots: XYZ, then XY => 2 time slots
Maximum number of time slots (3):
t=1 X, X, Z
t=2 , X, Y
t=3 , , Y

P5.

a. 1231 1231 1231 ...
b. 121 121 121 ...

P6.

Time	1	2	3	4	5	6	7	8	9	10	11	12
Packet	1	2	3	4	6	5	7	8	9	10	11	12

Packet	Time of Arrival	Time to leave the queue	Delay
2	0	2	2
3	1	3	2
4	1	4	3
5	3	6	3
6	2	5	3
7	3	7	4
8	5	8	3
9	5	9	4
10	7	10	3
11	8	11	3
12	8	12	4

b.

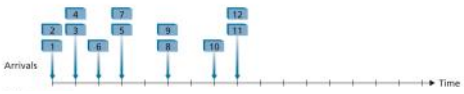
Time	1	2	3	4	5	6	7	8	9	10	11	12
Packet	1	3	5	7	9	2	4	11	6	8	10	12

Packet	Time of Arrival	Time to leave the queue	Delay
2	0	6	6
3	1	2	1
4	1	7	6
5	3	3	0
6	2	9	7
7	3	4	1
8	5	10	5
9	5	5	0
10	7	11	4
11	8	8	0
12	8	12	4

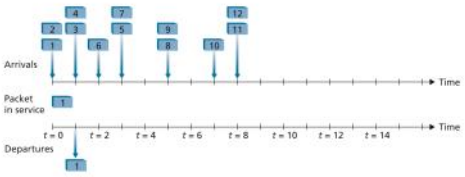
c.

Time	1	2	3	4	5	6	7	8	9	10	11	12

P6. Consider the figure below. Answer the following questions:



P6. Consider the figure below. Answer the following questions:



a. Assuming FIFO service, indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?

b. Now assume a priority service, and assume that odd-numbered packets are high priority, and even-numbered packets are low priority. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?

c. Now assume round robin service. Assume that packets 1, 2, 3, 6, 11, and 12 are from class 1, and packets 4, 5, 7, 8, 9, and 10 are from class 2. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and its departure? What is the average delay over all 12 packets?

d. Now assume weighted fair queuing (WFQ) service. Assume that odd-numbered packets are from class 1, and even-numbered packets are from class 2. Class 1 has a WFQ weight of 2, while class 2 has a WFQ weight of 1. Note that it may not be possible to achieve an idealized WFQ schedule as described in the text, so indicate why you have chosen the particular packet to go into service at each time slot. For each packet what is the delay between its arrival and its departure? What is the average delay over all 12 packets?

e. What do you notice about the average delay in all four cases (FIFO, RR, priority, and WFQ)?

P7. Consider again the figure for P6.

a. Assume a priority service, with packets 1, 4, 5, 6, and 11 being high-priority packets. The remaining packets are low priority. Indicate the slots in which packets 2 through 12 each leave the queue.

b. Now suppose that round robin service is used, with packets 1, 4, 5, 6, and 11 belonging to one class of traffic, and the remaining packets belonging to the second class of traffic. Indicate the slots in which packets 2 through 12 each leave the queue.

c. Now suppose that WFQ service is used, with packets 1, 4, 5, 6, and 11 belonging to one class of traffic, and the remaining packets belonging to the second class of traffic. Class 1 has a WFQ weight of 1, while class 2 has a WFQ weight of 2 (note that these weights are different than in the previous question). Indicate the slots in which packets 2 through 12 each leave the queue. See also the caveat in the question above regarding WFQ service.

P8. Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

b. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11001000 10010001 01010001 01010101
11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111

4	1	7	6
5	3	3	0
6	2	9	7
7	3	4	1
8	5	10	5
9	5	5	0
10	7	11	4
11	8	8	0
12	8	12	4

c.

Time	1	2	3	4	5	6	7	8	9	10	11	12
Packet	1	3	5	7	9	2	4	11	6	8	10	12

Packet	Time of Arrival	Time to leave the queue	Delay
2	0	6	6
3	1	2	1
4	1	7	6
5	3	3	0
6	2	9	7
7	3	4	1
8	5	10	5
9	5	5	0
10	7	11	4
11	8	8	0
12	8	12	4

P8.

a.

Prefix	Interface
11100000 00	0
11100000 00100000	1
11100000 01	2
Otherwise	3

Prefix	Interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 0	2
Otherwise	3

b. 11001000 10010001 01010001 01010101 => 3
11100001 01000000 11000011 00111100 => 2
11100001 10000000 00010001 01110111 => 3

P9. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match	Interface
00	0
010	1
011	2
10	2
11	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P10. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match	Interface
1	0
10	1
111	2
otherwise	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P11. Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

P12. In Section 4.2.2, an example forwarding table (using longest prefix matching) is given. Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

P13. In Problem P8, you are asked to provide a forwarding table (using longest prefix matching). Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

P14. Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

P15. Consider the topology shown in Figure 4.20. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

a. Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces. For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x – e.f.g.h/y.

b. Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

P16. Use the whois service at the American Registry for Internet Numbers (http://www.arin.net/whois) to determine the IP address blocks for three universities. Can the whois services be used to determine with certainty the geographical location of a specific IP address? Use www.maxmind.com to determine the locations of the Web servers at each of these universities.

P17. Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

P18. Consider the network setup in Figure 4.25. Suppose that the ISP instead assigns the router the address 24.34.101.225 and that the network address of the home network is 192.168.0/24.

a. Assign addresses to all interfaces in the home network.

b. Suppose each host has two ongoing TCP connections, all to port 80 at host 128.119.40.86. Provide the six corresponding entries in the NAT translation table.

P19. Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

a. Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.

b. If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.

P20. In this problem, we'll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

P21. Consider the SDN OpenFlow network shown in Figure 4.30. Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:

- any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts h1 or h2 should be forwarded over output port 2;

P9.

Address range	Interface
00000000 through 00111111	0
01000000 through 01011111	1
01100000 through 01111111	2
10000000 through 10111111	2
11000000 through 11111111	3

Interface 0: 64 addresses
Interface 1: 32 addresses
Interface 2: 32 + 64 = 96 addresses
Interface 3: 64 addresses

P10.

Address range	Interface
11000000 through 11011111	0
10000000 through 10111111	1
11100000 through 11111111	2
00000000 through 01111111	3

Interface 0: 32 addresses
Interface 1: 64 addresses
Interface 2: 32 addresses
Interface 3: 128 addresses

? P11.

Subnet 1: 223.1.17.0/26
~~Subnet 2: 223.1.17.64/25~~
Subnet 3: 223.1.17.192/28
Subnet 2: 223.1.17.128/25

P12.

Prefix	Interface
200.23.16.0/21	0
200.23.24.0/24	1
200.23.24.0/21	2
Otherwise	3

P13.

Prefix	Interface
224.0.0.0/10	0
224.32.0.0/16	1
224.0.0.0/8	2
225.0.0.0/9	2
Otherwise	3

P14.

One IP address that can be assigned is 128.119.40.128.
The last bytes of the four blocks should start with 0100, 0101, 0110 and 0111
~~128.119.40.64/26~~
~~128.119.40.80/26~~
~~128.119.40.96/26~~
~~128.119.40.102/26~~
128.119.40.64/28
128.119.40.80/28
128.119.40.96/28
128.119.40.102/28

P15.

a. Subnet A: 214.97.254.0/32 to 214.97.254.251/32 (last 2 bytes from 11111110 00000000 to 11111110 11111011)
Subnet B: 214.97.255.0/32 to 214.97.255.125/32 (last 2 bytes from 11111111 00000000 to 11111111 01111101)
Subnet C: 214.97.255.128/25 (last 2 bytes from 11111111 10000000 to 11111111 11111111)
Subnet D: 214.97.254.252/31 (last 2 bytes from 11111110 11111100 to 11111110 11111101)
Subnet E: 214.97.254.254/31(last 2 bytes from 11111110 11111110 to 11111110 11111111)
Subnet F: 214.97.255.126/31(last 2 bytes from 11111111 01111110 to 11111111 01111111)

b. Router 1

Subnet	Prefix	Interface
A	11010110 01100001 11111110	0
D	11010110 01100001 11111110 111111	1
F	11010110 01100001 11111111	2

Router 2

Subnet	Prefix	Interface
B	11010110 01100001 11111111	0
E	11010110 01100001 11111110	1
F	11010110 01100001 11111111 0111111	2

Router 3

Subnet	Prefix	Interface
C	11010110 01100001 11111111 1	0
D	11010110 01100001 11111110	1
E	11010110 01100001 11111111 0	2

P16.

No...

P17.

$$\frac{5 \times 10^6}{1480} = 3379$$

$$\frac{5 \times 10^6}{1460} = 3425$$

P18.

a. Host 1: 192.168.0.1
Host 2: 192.168.0.2
Host 3: 192.168.0.3
Router: 192.168.0.4

b.

WAN side	LAN side
24.34.101.225, 4000	192.168.0.1, 3345
24.34.101.225, 4001	192.168.0.1, 3346

a technique, discuss why.

P21. Consider the SDN OpenFlow network shown in Figure 4.30. Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:

- any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts h1 or h2 should be forwarded over output port 2;
- any datagrams arriving on input port 2 from hosts h1 or h2 that are destined to hosts h5 or h6 should be forwarded over output port 1;
- any arriving datagrams on input ports 1 or 2 and destined to hosts h3 or h4 should be delivered to the host specified;
- hosts h3 and h4 should be able to send datagrams to each other.

Specify the flow table entries in s2 that implement this forwarding behavior.

P22. Consider again the SDN OpenFlow network shown in Figure 4.30. Suppose that the desired forwarding behavior for datagrams arriving from hosts h3 or h4 at s2 is as follows:

- any datagrams arriving from host h3 and destined for h1, h2, h5 or h6 should be forwarded in a clockwise direction in the network;
- any datagrams arriving from host h4 and destined for h1, h2, h5 or h6 should be forwarded in a counter-clockwise direction in the network.

Specify the flow table entries in s2 that implement this forwarding behavior.

P23. Consider again the scenario from P21 above. Give the flow tables entries at packet switches s1 and s3, such that any arriving datagrams with a source address of h3 or h4 are routed to the destination hosts specified in the destination address field in the IP datagram. (*Hint:* Your forwarding table rules should include the cases that an arriving datagram is destined for a directly attached host or should be forwarded to a neighboring router for eventual host delivery there.)

P24. Consider again the SDN OpenFlow network shown in Figure 4.30. Suppose we want switch s2 to function as a firewall. Specify the flow table in s2 that implements the following firewall behaviors (specify a different flow table for each of the four firewalling behaviors below) for delivery of datagrams

destined to h3 and h4. You do not need to specify the forwarding behavior in s2 that forwards traffic to other routers.

- Only traffic arriving from hosts h1 and h6 should be delivered to hosts h3 or h4 (i.e., that arriving traffic from hosts h2 and h5 is blocked).
- Only TCP traffic is allowed to be delivered to hosts h3 or h4 (i.e., that UDP traffic is blocked).
- Only traffic destined to h3 is to be delivered (i.e., all traffic to h4 is blocked).
- Only UDP traffic from h1 and destined to h3 is to be delivered. All other traffic is blocked.

P25. Consider the Internet protocol stack in Figures 1.23 and 4.31. Would you consider the ICMP protocol to be a network-layer protocol or a transport-layer protocol? Justify your answer.

Wireshark Lab: IP

In the Web site for this textbook, www.pearsonglobaleditions.com, you'll find a Wireshark lab assignment that examines the operation of the IP protocol, and the IP datagram format in particular.

Router: 192.168.0.4

b.

WAN side	LAN side
24.34.101.225, 4000	192.168.0.1, 3345
24.34.101.225, 4001	192.168.0.1, 3346
24.34.101.225, 4002	192.168.0.2, 3345
24.34.101.225, 4003	192.168.0.2, 3346
24.34.101.225, 4004	192.168.0.3, 3345
24.34.101.225, 4005	192.168.0.3, 3346

P19.

a. Since all IP packets are sent outside, we can use a packet sniffer to record all IP packets generated by the hosts behind a N AT. We can group IP packets with consecutive IDs into a cluster. The number of clusters is the number of hosts behind the NAT
b. No, it would not be possible since there would not be clusters for the sniffed packets.

P20.

Without an application-specific NAT configuration, Arnold and Bernard will not know the port numbers that NAT uses to map WAN addresses to LAN addre sses.

P21.

Input Port	Source IP	Destination IP	Output Port
1	10.3.0.*	10.1.0.*	2
2	10.1.0.*	10.3.0.*	1
1	*	10.2.0.3	3
1	*	10.2.0.4	4
2	*	10.2.0.4	4
2	*	10.2.0.3	3
3	10.2.0.3	10.2.0.4	4
4	10.2.0.4	10.2.0.3	3

P22.

Input Port	Source IP	Destination IP	Output Port
3	10.2.0.3	10.1.0.1	2
3	10.2.0.3	10.1.0.2	2
3	10.2.0.3	10.3.0.5	2
3	10.2.0.3	10.3.0.6	2
4	10.2.0.4	10.2.0.4	1
4	10.2.0.4	10.2.0.3	1
4	10.2.0.4	10.3.0.5	1
4	10.2.0.4	10.3.0.6	1

Input Port	Source IP	Destination IP	Output Port
3	10.2.0.3	10.1.0.*	2
3	10.2.0.3	10.3.0.*	2
4	10.2.0.4	10.1.0.*	1
4	10.2.0.4	10.3.0.*	1

P23.

Switch 1's flow table

Source IP	Destination IP	Output Port
10.2.0.*	10.1.0.1	2
10.2.0.*	10.1.0.2	3
10.2.0.*	10.3.0.*	1

Switch 3's flow table

Source IP	Destination IP	Output Port
10.2.0.*	10.3.0.5	2
10.2.0.*	10.3.0.6	1
10.2.0.*	10.1.0.*	3

Chapter 4 Tricky Concepts

Monday, August 7, 2023

Network service model

- Services for individual datagrams
- Services for a flow of datagrams

Buffer management

- Drop (tail drop or priority)
- Marking (ECN, RED)

IP fragmentation/reassembly

IP addressing: CIDR

Chapter 5.1 - Introduction to the Control Plane

Wednesday, July 12, 2023

Introduction

- The forwarding table and the flow table were the principal elements that linked the network layer's data and control planes
- How are forwarding and flow tables computed, maintained and installed?
- There are two possible approaches
- Per-router control
    - A routing algorithm runs in each and every router
    - Both a forwarding and a routing function are contained within each router
    - Each router has a routing component that communicates with the routing components in other routers to compute the values for its forwarding table
    - Examples: OSPF and BGP protocols
- Logically centralized control
    - A logically centralized controller computes and distributes the forwarding tables to be used by each and every router
    - The controller interacts with a **control agent** (CA) in each of the routers via a well-defined protocol to configure and manage that router's flow table
    - Typically the CA has minimum functionality; its job is to communicate with the controller, and to do as the controller commands
- Key distinction between per-router control and logically centralized control:
    - Unlike the routing algorithms, the CAs do not directly interact with each other
    - Nor do they actively take part in computing the forwarding table

Chapter 5.2 - Routing Algorithms

Wednesday, July 12, 2023

Routing algorithms

- A **routing algorithm** determines good paths from senders to receivers that have the least cost.
- A **graph** $G = (N, E)$ is a set N of nodes and a collection E of edges, where each edge is a pair of nodes from N
- In networking, the nodes represent routers and the edges represent the physical links between the routers
- An edge may also have a value representing its cost (physical length, link speed, or the monetary cost, etc.)
- We denote $c(x, y)$ as the **cost** of the edge between nodes x and y
- If the (x, y) is not in E, we set $c(x, y) = \infty$
- We only consider **undirected graphs**
- The goal of a routing algorithm is to identify the **least-cost path** (aka **shortest path**) between two nodes

Types of routing algorithms

- Centralized routing algorithm
  - Computes the least-cost path using complete, global knowledge about the network
  - Takes the connectivity between all nodes and all link costs as inputs
  - The calculation itself can be run at one site (e.g. a logically centralized controller) or could be replicated in the routing component of each and every router
- Decentralized routing algorithm
  - The calculation of the least-cost path is carried out in an interactive, distributed manner by the routers
  - No node has complete information about the costs of all network links
  - Each node begins with only the knowledge of the cost of its own directly attached links
  - Through an iterative process of calculation and exchange of information with its neighbors, a node gradually calculates the least-cost path to a destination or a set of destinations
  - Probably more naturally suited to control planes where the routers interact directly with each other
- Static routing algorithm
  - Routes change very slowly over time
  - Often as a result of human intervention
- Dynamic routing algorithm
  - Change the routing paths as the network traffic loads or topology change
  - Can be run either periodically or in direct response to topology or link cost changes
  - More responsive to network changes
  - More susceptible to problems such as routing loops and route oscillation
- Load-sensitive algorithm
  - Link costs vary dynamically to reflect the current level of congestion in the link
  - If a high cost is associated with a link that is currently congested, a routing algorithm will tend to choose routes around such a congested link
- Load-insensitive algorithm
  - A link's cost does not explicitly reflect its current (or recent past) level of congestion

Link-State (LS) Routing Algorithm

- The network topology and all link costs are known, that is, available as input to the LS algorithm
- This is accomplished by having each node broadcast link-state packets to all other nodes in the network, with each link-state packet containing the identities and costs of its links
- The result of the node's broadcast is that all nodes have an identical and complete view of the network

- Each node can then run the LS algorithm and compute the same set of least-cost paths as every other node
- We will use Dijkstra's algorithm, which has the following properties
  - Iterative
  - After the kth iteration of the algorithm, the least-cost paths are known to k destination nodes
  - Among the least-cost paths to all destination nodes, these k paths will have the k smallest costs
  - $D(v)$: cost of the least-cost path
  - $p(v)$: previous node along the current least-cost path from the source to v
  - N': subset of nodes; v is in N' if the least-cost path from the source v is definitively known

LS Algorithm for Source Node u

```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v is a neighbor of u
5       then D(v) = c(u,v)
6     else D(v) = ∞
7
8  Loop
9   find w not in N' such that D(w) is a minimum
10  add w to N'
11  update D(v) for each neighbor v of w and not in N':
12        D(v) = min(D(v), D(w)+ c(w,v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

LS Algorithm: Example

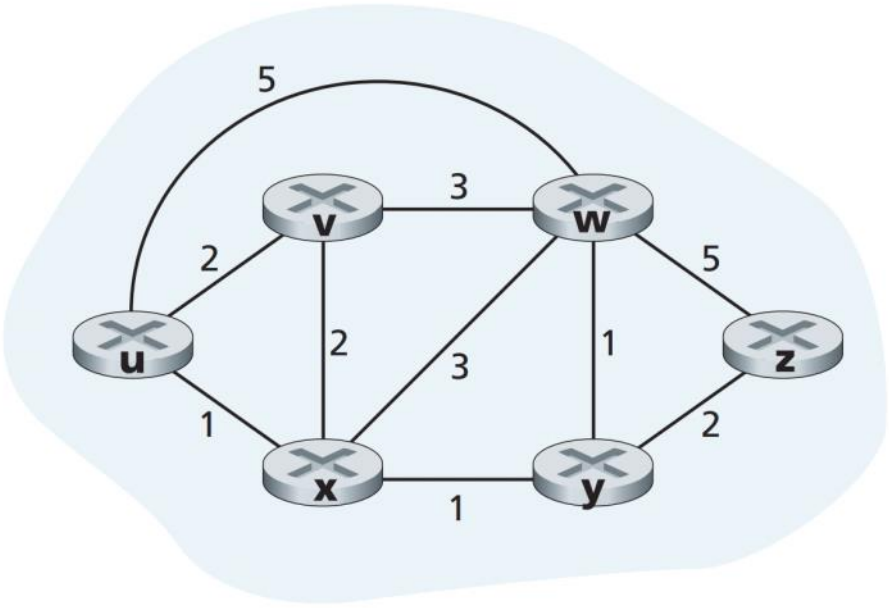
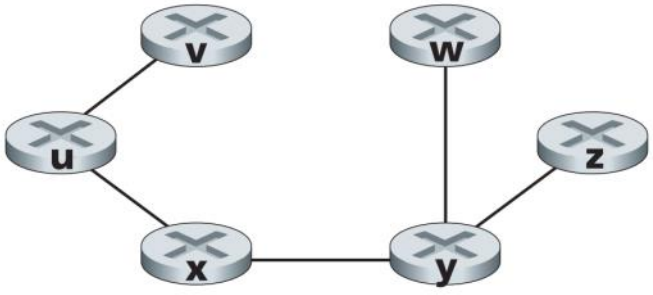


Figure 5.3 ♦ Abstract graph model of a computer network

step	*N′*	*D*(*v*), *p*(*v*)	*D*(*w*), *p*(*w*)	*D*(*x*), *p*(*x*)	*D*(*y*), *p*(*y*)	*D*(*z*), *p*(*z*)
0	u	2, u	5, u	1,u	∞	∞
1	ux	2, u	4, x		2, x	∞
2	uxy	2, u	3, y			4, y
3	uxyv		3, y			4, y
4	uxyvw					4, y
5	uxyvwz					

Table 5.1 ♦ Running the link-state algorithm on the network in Figure 5.3



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

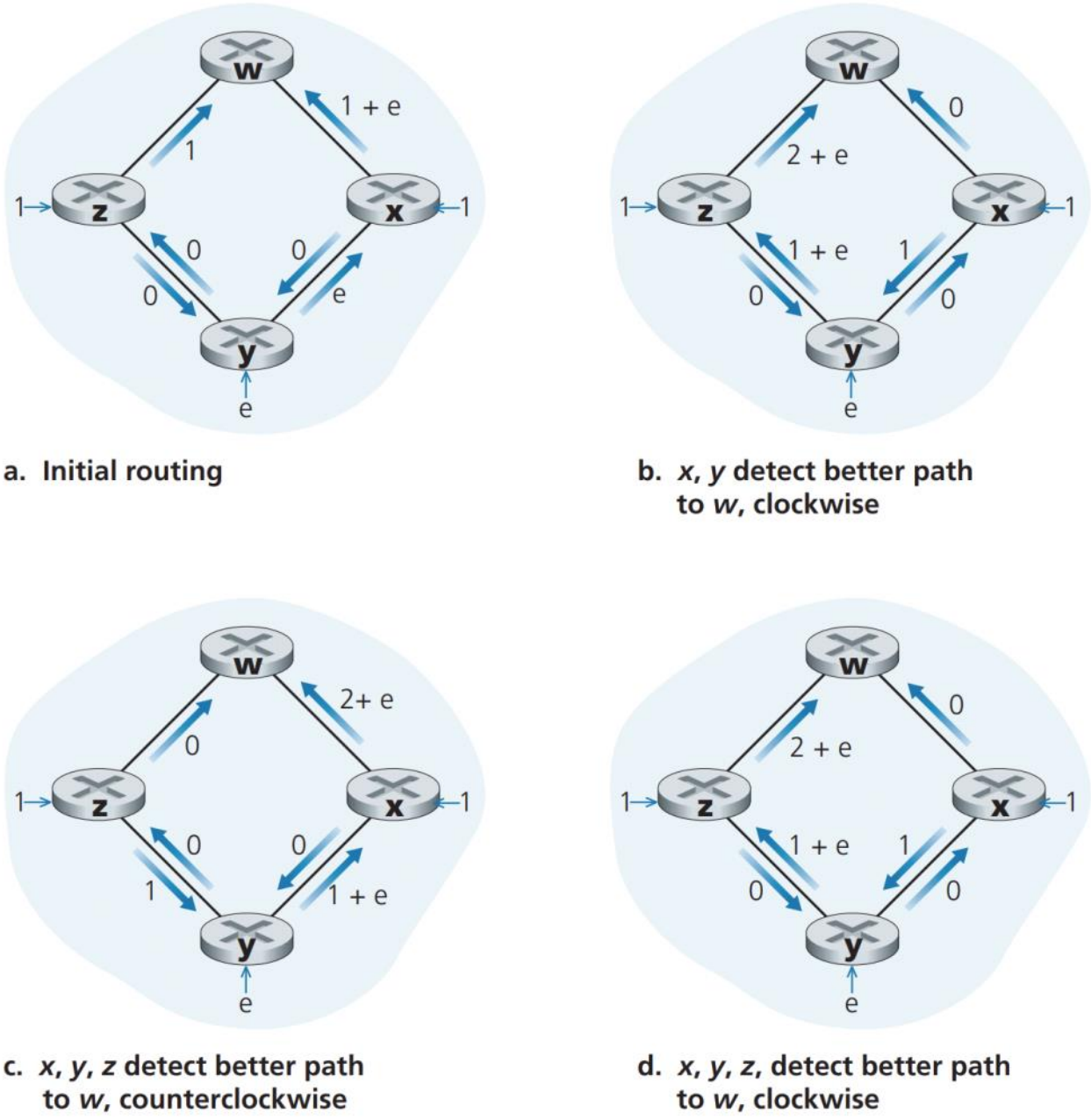
Figure 5.4 ♦ Least cost path and forwarding table for node u

LS Algorithm: Notes

- When the algorithm terminates, we have, for each node, its predecessor along the least-cost path from the source node
- For each predecessor, we also have its predecessor
- In this manner we can construct the entire path from the source to all destinations
- This implementation of the LS algorithm has worst-case complexity of $O(n^2)$
- Using the heap data structure, the complexity can be reduced to $O(n \log n)$

LS Algorithm: Oscillation

- Example (Figure 5.5)
  - In (a), node z originates a unit of traffic destined for w, node x also originates a unit of traffic destined for w, and node injects an amount of traffic equal to e, also destined for w
  - In (b), x, y and z all determine that their new least-cost paths to w are clockwise
  - In (c), x, y and z all determine that their new least-cost paths to w are counterclockwise
  - In (d), x, y and z all determine that their new least-cost paths to w are clockwise
  - …
  - This results in an oscillation
- This can occur in any algorithm that uses a congestion or delay-based link metric
- How to prevent such oscillations?
  - Mandate that link costs not depend on the amount of traffic carried - unacceptable since one goal of routing is to avoid highly congested links
  - Ensure that not all routers run the LS algorithm at the same time - more reasonable solution
  - However, routers in the Internet can still self-synchronize among themselves
  - One way to avoid such self-synchronization is for each router to randomize the time it sends out a link advertisement



a. Initial routing

b. *x*, *y* detect better path to *w*, clockwise

c. *x*, *y*, *z* detect better path to *w*, counterclockwise

d. *x*, *y*, *z*, detect better path to *w*, clockwise

Figure 5.5 ♦ Oscillations with congestion-sensitive routing

Distance-Vector (DV) Routing Algorithm

- The distance vector (DV) algorithm has the following properties
  - Distributed
    - Each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors
  - Iterative
    - This process continues on until no information is exchanged between neighbors
  - Asynchronous
    - Does not require all of the nodes to operate in lockstep with each other
- Bellman-Ford equation

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

- The solution to the Bellman-Ford equation provides the entries in node x's forwarding table
- Basic idea
  - Each node x begins with $D_x(y)$, an estimate of the cost of the least-cost path from itself to node y, for all nodes y in N
  - Each node x maintains the following routing information
    - For each neighbour v, the cost $c(x,v)$ from x to a directly attached neighbour, v
    - Node x's distance vector $\boldsymbol{D}_x = [D_x(y): y \in N]$ containing x's estimate of its cost to all destinations y in N
    - The distance vectors of each of its neighbours $\boldsymbol{D}_v = [D_v(y): y \in N]$ for each neighbour v of x
  - Each node sends a copy of its distance vector to each of its neighbours
  - When a node receives a new distance vector from any of its neighbours w, it saves w's distance vector, and uses the Bellman-Ford equation to update its own distance vector
  - If node x's distance vector changed as a result of this update step, it will send its updated distance vector to each of its neighbors

DV Algorithm

```
1  Initialization:
2    for all destinations y in N:
3       Dx(y)= c(x,y)/* if y is not a neighbor then c(x,y)= ∞ */
4    for each neighbor w
5       Dw(y) = ? for all destinations y in N
6    for each neighbor w
7       send distance vector  Dx = [Dx(y): y in N] to w
8
9  loop
10    wait  (until I see a link cost change to some neighbor w or
11             until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14        Dx(y) = minv{c(x,v) + Dv(y)}
15
16 if Dx(y) changed for any destination y
17       send distance vector Dx  = [Dx(y): y in N] to all neighbors
18
19 forever
```

DV algorithm: Notes

- The DV algorithm is decentralized, so it does not use global information such as a complete map of the network
- The only information a node will have is the costs of the links to its directly attached neighbours and information it receives from these neighbours
- DV-like algorithms are used in many routing protocols in practice, including the Internet's RIP and BGP

DV algorithm: Example

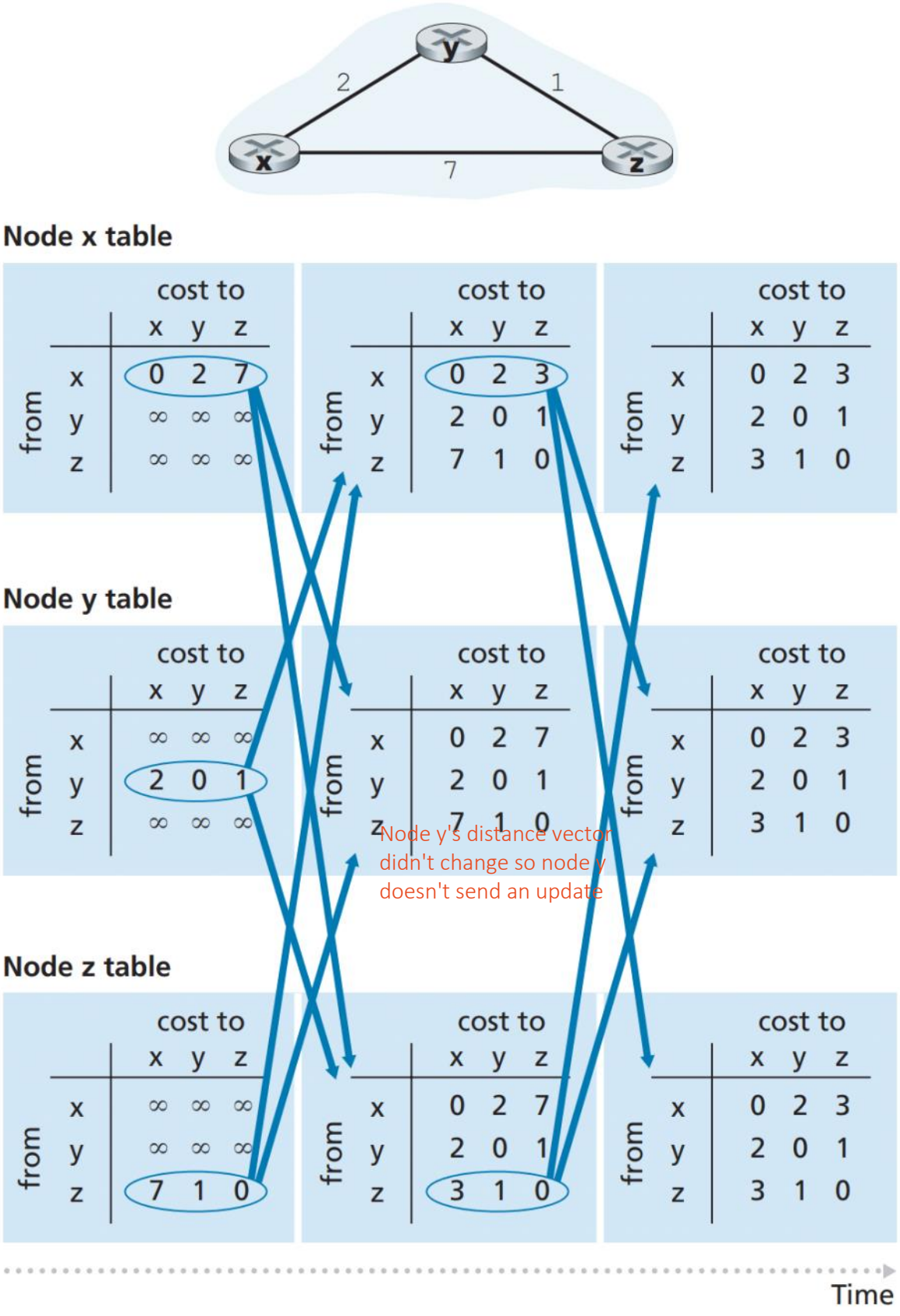


Figure 5.6 ♦ Distance-vector (DV) algorithm in operation

DV Algorithm: Link-Cost Changes and Link Failure

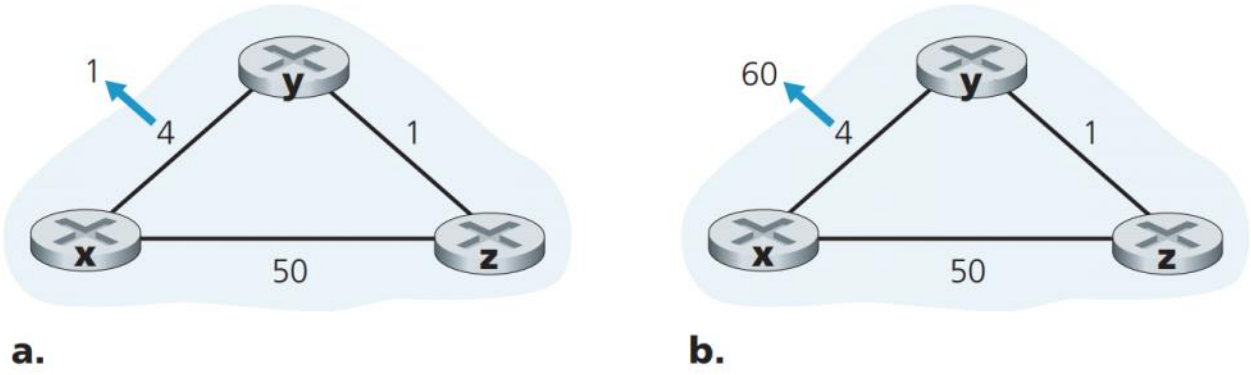


Figure 5.7 ♦ Changes in link cost

- Here we focus only on y' and z's distance table entries to destination x
- In Figure 5.7(a), suppose the link cost from y to x changes from 4 to 1
  - At t=0, node y detects the link-cost change, updates its distance vector, and informs its neighbors of this change
  - At t=1, node z receives the update from y and updates its table. It computes a new least cost to x (from 5 to 2) and sends its new distance vector to its neighbors
  - At t=2, node y receives z's update and updates its distance table. Its least costs do not change and hence does not send any message to z
  - The good news about the decreased cost between x and y has propagated quickly through the network
- In Figure 5.7(b), the link cost between x and y increases from 4 to 60
  - Initially, $D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$
  - At t=0, node y detects a link cost change, and computes a new minimum-cost path to x to be $\min\{60 + 0, 1 + 5\} = 6$
  - At t=1, from a global view, the new cost via z to x is wrong, however this is correct using the only information node y has
  - Now we have a routing loop - in order to get to x, y routes through z, and z routes through y
  - Since node y has computed a new minimum cost to x, it informs z of its new distance vector at t=1
  - When z receives y's new distance vector, indicating that y's minimum cost to x is 6, z knows it can get to y with a cost of 1 and hence computes a least cost to x of $\min\{50 + 0, 1 + 6\} = 7$ and informs y of its new distance vector at t=2
  - Node y then determines $D_y(x) = 8$ and sends z its distance vector
  - Node z then determines $D_z(x) = 9$ and sends y its distance vector
  - ...
- This loop will persist for 44 iterations until z eventually computes the cost of its path via y to be greater than 50
- At this point, z will determine that its least-cost path to x is via its direct connection to x. y will then route to x via z
- This problem is sometimes referred to as the count-to-infinity problem

DV Algorithm: Adding Poisoned Reverse

- The above problem can be avoided using a technique known as poisoned reverse

- If z routes through y to get to destination x, then z will advertise to y that its distance to x is infinity
- Since y believes that z has no path to x, y will never attempt to route to x via z, as long as z continues to route to x via y (and lies about doing so)
- However, poisoned reverse does not solve the general count-to-infinity problem

A Comparison of LS and DV Routing Algorithm

- Message complexity
  - LS requires each node to know the cost of each link in the network, which requires $O(nm)$ messages to be sent
  - In LS, whenever a link cost changes, the new link cost must be sent to all nodes
  - DV requires message exchanges between directly connected neighbours at each iteration
  - When link costs change, DV will propagate the results of the changed link cost only if the new link cost results in a changed least-cost path for one of the nodes attached to that link
- Speed of convergence
  - LS is an $O(n^2)$ algorithm
  - DV can converge slowly and can have routing loops while the algorithm is converging
  - DV also suffers from the count-to-infinity problem
- Robustness
  - What can happen if a router fails, misbehaves, or is sabotaged?
  - An LS node is computing only its own forwarding tables. Other nodes are performing similar calculations for themselves
  - Thus, route calculations are somewhat separated under LS, providing a degree of robustness
  - Under a DV, a node can advertise incorrect least-cost paths to any or all destinations
  - More generally, at each iteration, a node's calculation in DV is passed on to its neighbor and then indirectly to its neighbors' neighbor on the next iteration
  - An incorrect node calculation can be diffused through the entire network under DV

Chapter 5.3 - Intra-AS Routing in the Internet: OSPF

Thursday, July 20, 2023

Introduction

- The view of a homogenous set of routers all executing the same routing algorithm is simplistic for two reasons
    - Scale
        - As the number of routers becomes large, the overhead involved in communicating, computing and storing routing information becomes prohibitive
        - A DV algorithm that iterated among such a large number of routers would surely never converge
    - Administrative autonomy
        - An ISP generally desires to operate its network as it pleases or to hide aspects of its networks internal organization from the outside
        - Ideally, an organization should be able to operate and administer its network as it wishes, while still being able to connect its network to other outside networks
- Both of these problems can be solved by organizing routers into **autonomous systems (AS)** with each AS consisting of a group of routers that are under the same administrative control
- Often the routers in an ISP and the links that interconnect them, constitute a single AS
- An AS is identified by its globally unique autonomous system number (ASN), which are assigned by ICANN regional registries
- Routers within the same AS all run the same routing algorithm and have information about each other
- The routing algorithm within an AS is called an **intra-autonomous system routing protocol**

Open Shortest Path First (OSPF)

- OSPF is a link-state protocol that uses flooding of link-state information and a Dijkstra's least-cost path algorithm
- Each router constructs a complete topological map of the entire AS
- Each router then locally runs Dijkstra's shortest-path algorithm to determine a shortest-path tree to all subnets, with itself as the root node
- OSFP does not mandate a policy for how link weights are set (this is the job of the network administrator) but instead provides the mechanisms for determining least-cost path routing for the given set of link weights
- A router broadcasts routing information to all other routers in the AS
    - This happens whenever there is a change in a link's state
    - Also happens periodically (at least once every 30 minutes) even if the link's state has not changed
- OSPF advertisements are contained in OSPF messages that are carried directly by IP, so it must itself implement functionalities such as reliable data transfer and link-state broadcast
- Security
    - Exchanges between OSPF routers can be authenticated
    - With authentication, only trusted routers can participate in the OSPF protocol within an AS
    - Prevent malicious intruders from injecting incorrect information into router tables
- Multiple same-cost paths

    - When multiple paths to a destination have the same cost, OSPF allows multiple paths to be used
- Integrated support for unicast and multicast routing
    - Multicast OSPF provides for multicast routing
    - It uses the existing OSFP link database and adds a new type of link-state advertisement to the existing link-state broadcast mechanism
- Support for hierarchy within a single AS
    - An OSPF AS can be configured hierarchically into areas
    - Each area runs its own OSPF link-state routing algorithm, with each router in an area broadcasting its link state to all other routers in that area
    - Each area has one or more area border routers responsible for routing packets outside the area
    - Exactly one OSPF area in the AS is configured to be the backbone area, which routes traffic between the other areas in the AS
    - The backbone always contains all area border routers in the AS and may contain non-border routers as well
    - Packets must be first routed to an area border router (intra-area routing), then routed through the backbone to the area border router that is in the destination area, and then to the final destination

Chapter 5.4 - Routing Among the ISPs: BGP

Thursday, July 20, 2023

Introduction

- To route a packet across multiple ASs, we need an **inter-autonomous system routing protocol**
- Communicating ASs must run the same inter-AS routing protocol
- In the Internet, all ASs run the same inter-AS routing protocol, called the **Border Gateway Protocol**, more commonly known as **BGP**
- BGP is a decentralized and asynchronous protocol in the vein of distance-vector routing

The Role of BGP

- In BGP, packets are not routed to a specific destination address, but instead to CIDRized prefixes, which represent subnets or collections of subnets
- A destination may take the form 138.16.68/22, which for this example includes 1024 ($= 2^{10}$) IP addresses
- A router's forwarding table will have entries of the form (x, I) where x is a prefix and I is an interface number for one of the router's interfaces
- BGP provides each router a means to:
    - Obtain prefix reachability information from neighboring ASs
        - Each subnet can advertise its existence to the rest of the Internet so that all the routers in the Internet know about this subnet
    - Determine the "best" routes to the prefixes
        - A router may learn about two or more different routes to a specific prefix
        - The router will locally run a BGP route-selection procedure

Advertising BGP Route Information

- For each AS, each router is either a **gateway router** or an **internal router**
- Gateway router: a router on the edge of an AS that directly connects to one or more routers in other ASs
- Internal router: a router that connects only to hosts and routers within its own AS
- Example (Figure 5.9):
    - First, AS3 sends a BGP message to AS2 "AS3 x"
    - Then AS2 sends a BGP message to AS1 "AS2 AS3 x"
- Each AS will not only learn about the existence of x, but also learn about the path of AS that leads to x
- In BGP, pairs of routers exchange routing information over semi-permanent TCP connections using port 179
- Each such TCP connection, along with all the BGP messages, is called a **BGP connection**
- A BGP connection that spans two ASs is called an **external BGP (eBGP) connection**
- A BGP connection between routers in the same AS is called an **internal BGP (iBGP) connection**
- To propagate the reachability information, both iBGP and eBGP sessions are used
- Consider the same example (Figure 5.9) again
    - Gateway router 3a first sends an eBGP message "AS3 x" to gateway router 2c
    - Gateway router 2c then sends an iBGP message "AS3 x" to all other routers in AS2, including 2a
    - Gateway router 2a then sends an eBGP message "AS2 AS3 x" to gateway router 1c

- Gateway router 1c then sends an iBGP message "AS2 AS3 x" to all other routers in AS1, including 1a

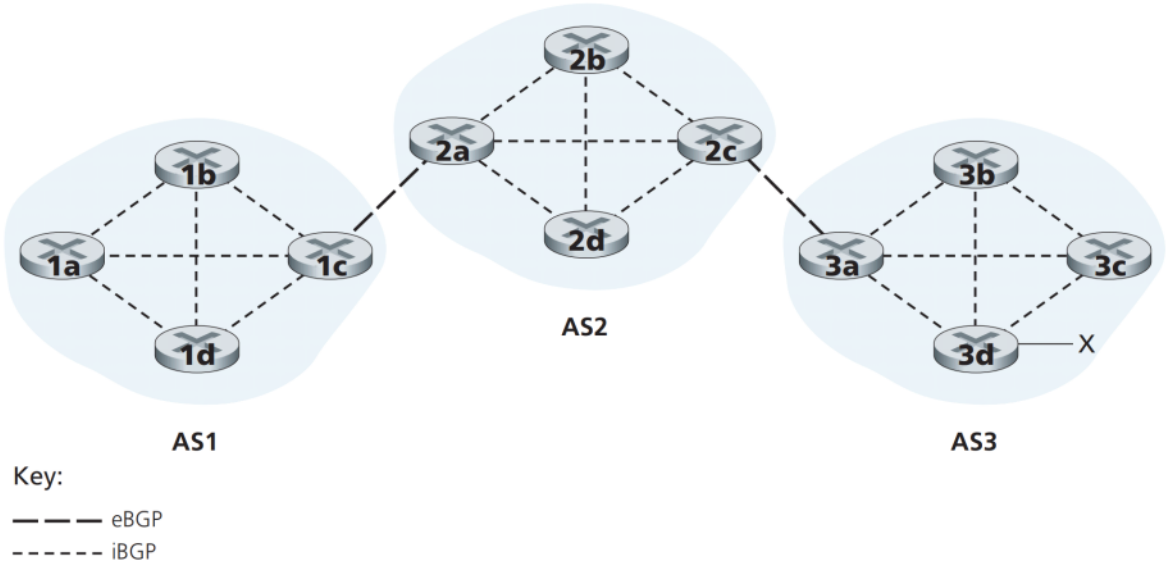


Figure 5.9 ♦ eBGP and iBGP connections

Determining the Best Route

- How does a router choose a path?
- When a router advertises a prefix across a BGP connection, it includes with the prefix several **attributes**
- A prefix along with its attributes is called a **route**
- Two of the more important attributes are AS-PATH and NEXT-HOP
- AS-PATH
  - Contains the list of ASs advertisement has passed
  - When a prefix is passed to an AS, the AS adds its ASN to the existing list in the AS-PATH
  - BGP routers also use the AS-PATH attribute to detect and prevent looping advertisements
  - If a router sees that its own AS is contained in the path list, it will reject the advertisement
  - Example (Figure 5.10): there are two routes from AS1 to subnet x
    - One uses the AS-PATH "AS2 AS3"
    - Another uses the AS-PATH "A3"
- NEXT-HOP
  - The IP address of the router interface that begins the AS-PATH
  - It is an IP address of a router that does not belong to the current AS
  - Example (Figure 5.10)
    - The NEXT-HOP attribute for the route "AS2 AS3 x" from AS1 to x is the IP address of the left interface on router 2a
    - The NEXT-HOP attribute for the route "AS3 x" from AS1 to x is the IP address of the left interface on router 2a

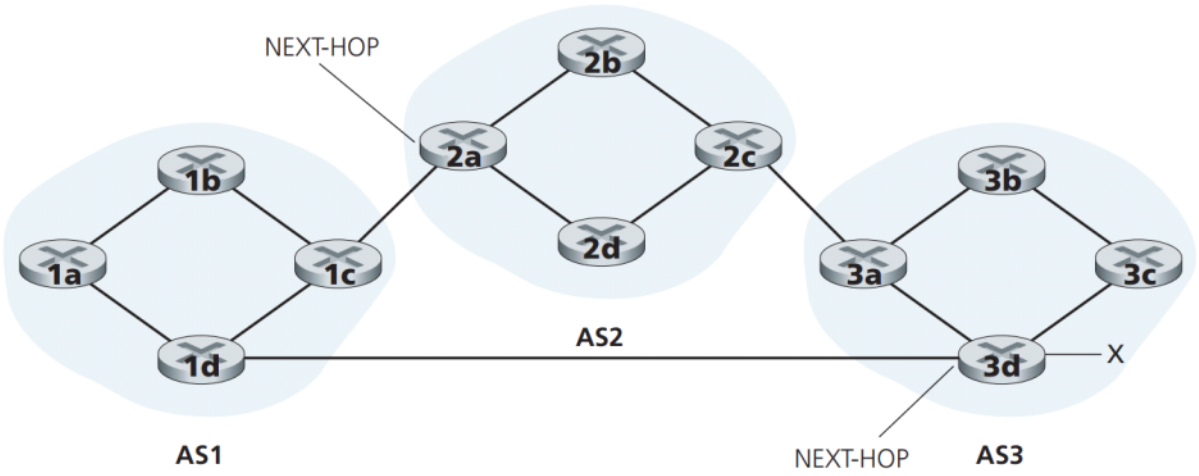


Figure 5.10 ♦ Network augmented with peering link between AS1 and AS3

Hot Potato Routing

- The route chosen is that route with the least cost to the NEXT-HOP router beginning that route
- Example (Figure 5.10)
  - Router 1b will consult its intra-AS routing information to find the least-cost intra-AS path to router 2a and the least-cost intra-AS path to the router 3d
  - Router 1b then select the route with the smallest cost
  - If the cost is defined as the number of links traversed, router 1b will select router 2a
  - Router 1b would then consult its forwarding table and find the interface I that is on the least-cost path to router 2a
  - It then adds (x, I) to its forwarding table
- When adding an outside-AS prefix into a forwarding table, both the inter-AS routing protocol (BGP) and the intra-AS routing protocol (e.g. OSPF) are used
- The idea behind hot-potato routing is to get packets out of its AS as quickly as possible without worrying about the cost of the remaining portions of the path outside of its AS
- Hot potato routing is thus a selfish algorithm that tries to reduce the cost in its own AS while ignoring the other components of the end-to-end costs outside its AS

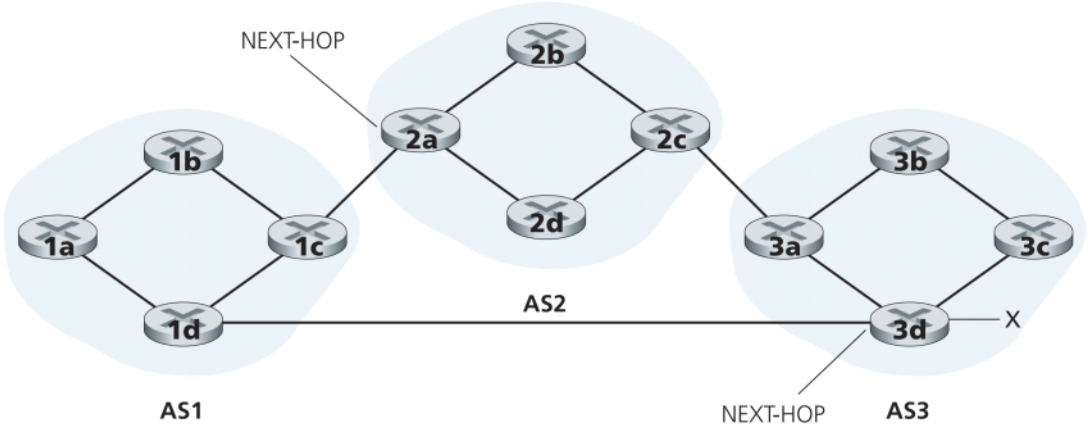


Figure 5.10 ♦ Network augmented with peering link between AS1 and AS3

Route-Selection Algorithm

- If there are two or more routes to the same prefix, then BGP sequentially invokes the following elimination rules until one route remains
  - A route is assigned a **local preference** value as one of its attributes (in addition to the AS-PATH and NEXT-HOP attributes)
  - The local preference could have been set by the router or could have been learned from another router in the same AS
    - The value is a policy decision that is left to the AS's network administrator
  - From the remaining routes, the route with the shortest AS-PATH is selected
  - From the remaining routes, hot potato routing is used, that is, the route with the closest NEXT-HOP router is selected
  - If more than one route still remains, the router uses BGP identifiers to select the route
- Example (Figure 5.10)
  - BGP will select the route that bypasses AS2, since that route has a shorter AS PATH
  - With the above route-selection algorithm, BGP is no longer a selfish algorithm since it first looks for routes with short AS paths (thereby likely reducing end-to-end delay)

Routing Policy (Example using Figure 5.13)

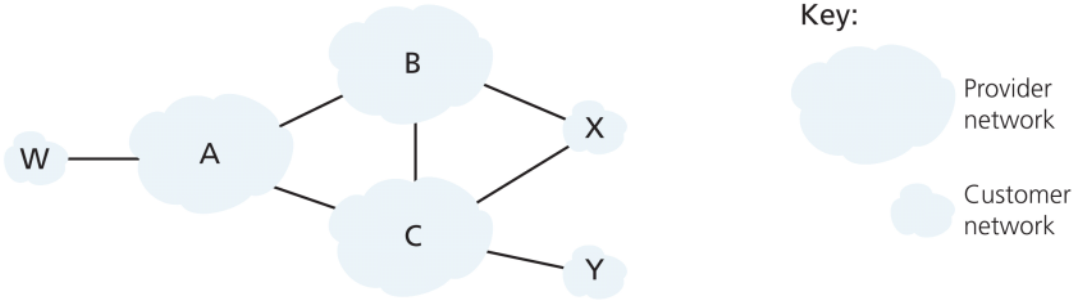


Figure 5.13 ♦ A simple BGP policy scenario

- A, B, C, W, X, Y are all ASs
- A, B, C are backbone provider networks
- W, X, Y are access ISPs
- All traffic entering an ISP access network must be destined for that network
- All traffic leaving an ISP network must have originated in that network
- X is a **multi-home access ISP**
- X will function as an access ISP network if it advertises (to its neighbors B and C) that it has no paths to any other destination except itself
  - Even though X may know of a path that reaches network Y (e.g. XCY), X will not advertise it to B

    - This makes sure that B will never forward traffic destined to Y (or C) via X
- Suppose that B has learned that A has a path AW to W
    - B can thus install the route AW into its routing information base
    - B also wants to advertise the path BAW to its customer, X
    - However, B might not advertise the path BAW to C
    - B might feel that it is A's and C's job to make sure C can route to/from A's customers via a direct connection between A and C

Chapter 5.5 - The SDN Control Plane

Tuesday, July 25, 2023

Introduction

- Four key characteristics of an SDN architecture can be identified
  - Flow-based forwarding
    - Packet forwarding by SDN-controlled switches can be based on any number of header field values in the transport-layer, network-layer or link-layer header
    - Packet forwarding rules are specified in a switch's flow table
    - It is job of the SDN control plane to compute, manage and install flow table entries in all switches
  - Separation of data plane and control plane
    - The data plane consists of the network's switches - relatively simple (but fast) devices that execute the "match plus action" rules in their flow tables
    - The control plane consists of servers and software that determine and manage the switches' flow tables
  - Network control functions: external to data-plane switches
    - SDN control plane is implemented in software
    - This software executes on servers that are both distinct and remote from the network's switches
  - A programmable network
    - The network is programmable through the network-control applications running in the control plane
    - These applications use the APIs provided by the SDN controller to specify and control the data plane in the network devices

The SDN Control Plane: SDN Controller and SDN Network-control Applications

- The SDN control plane divides broadly into two components
  - The SDN controller
  - The SDN network-control applications
- A controller's functionality can be broadly organized into three layers
  - Communication layer
    - Communicating between the controller and controlled network devices
  - Network-wide state-management layer
    - The ultimate control decisions made by the SDN control plane will require that the controller have up-to-date information about state of the networks' hosts, links, switches and other SDN-controlled devices
  - The interface to the network-control application layer
    - The controller interacts with network-control applications through its "northbound" interface
    - This API allows network-control applications to read/write network state and flow tables within the state-management layer
- The SDN controller can be considered to be "logically centralized", that is, the controller may be viewed externally as a single, monolithic service

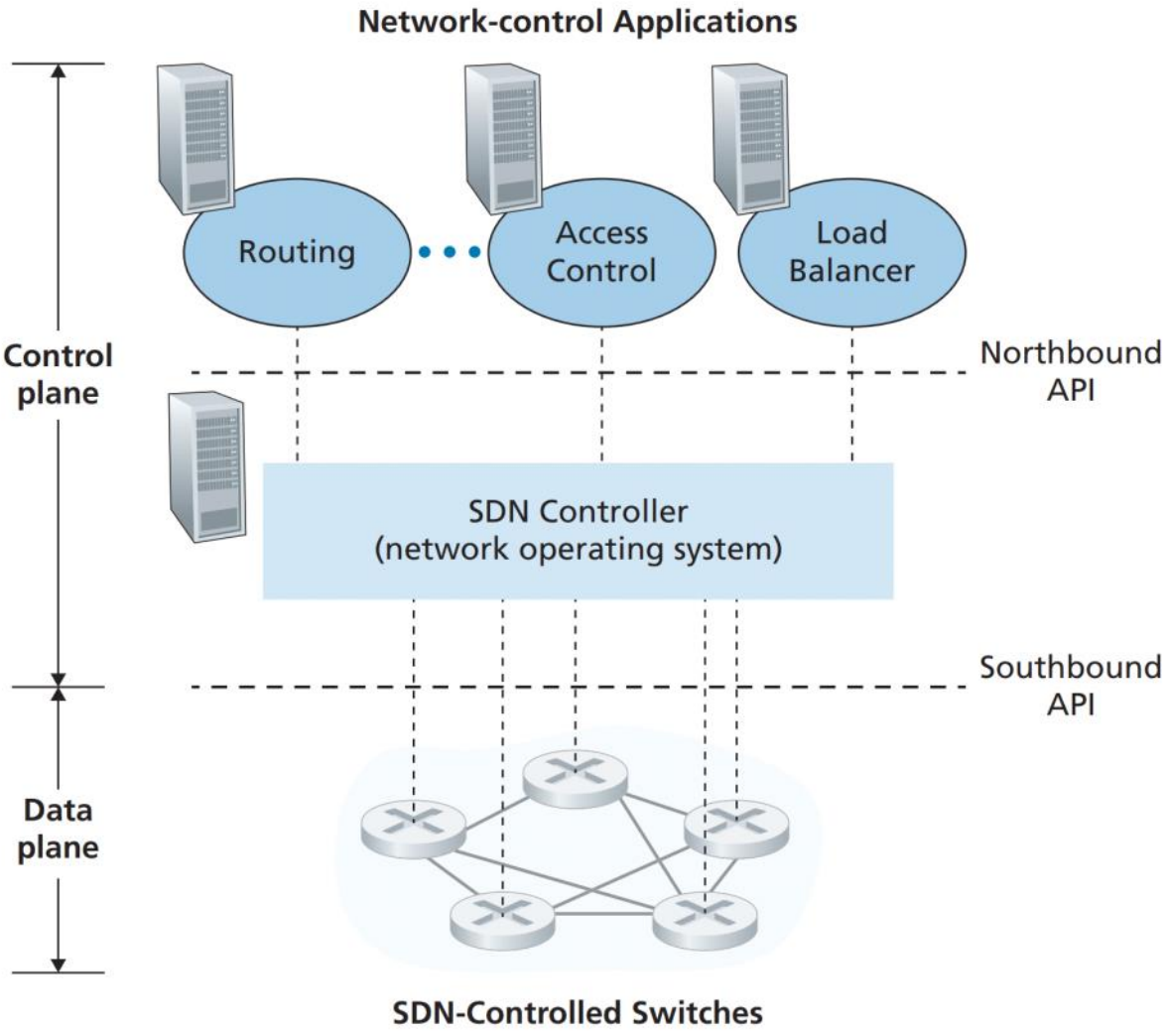


Figure 5.14 ♦ Components of the SDN architecture: SDN-controlled switches, the SDN controller, network-control applications

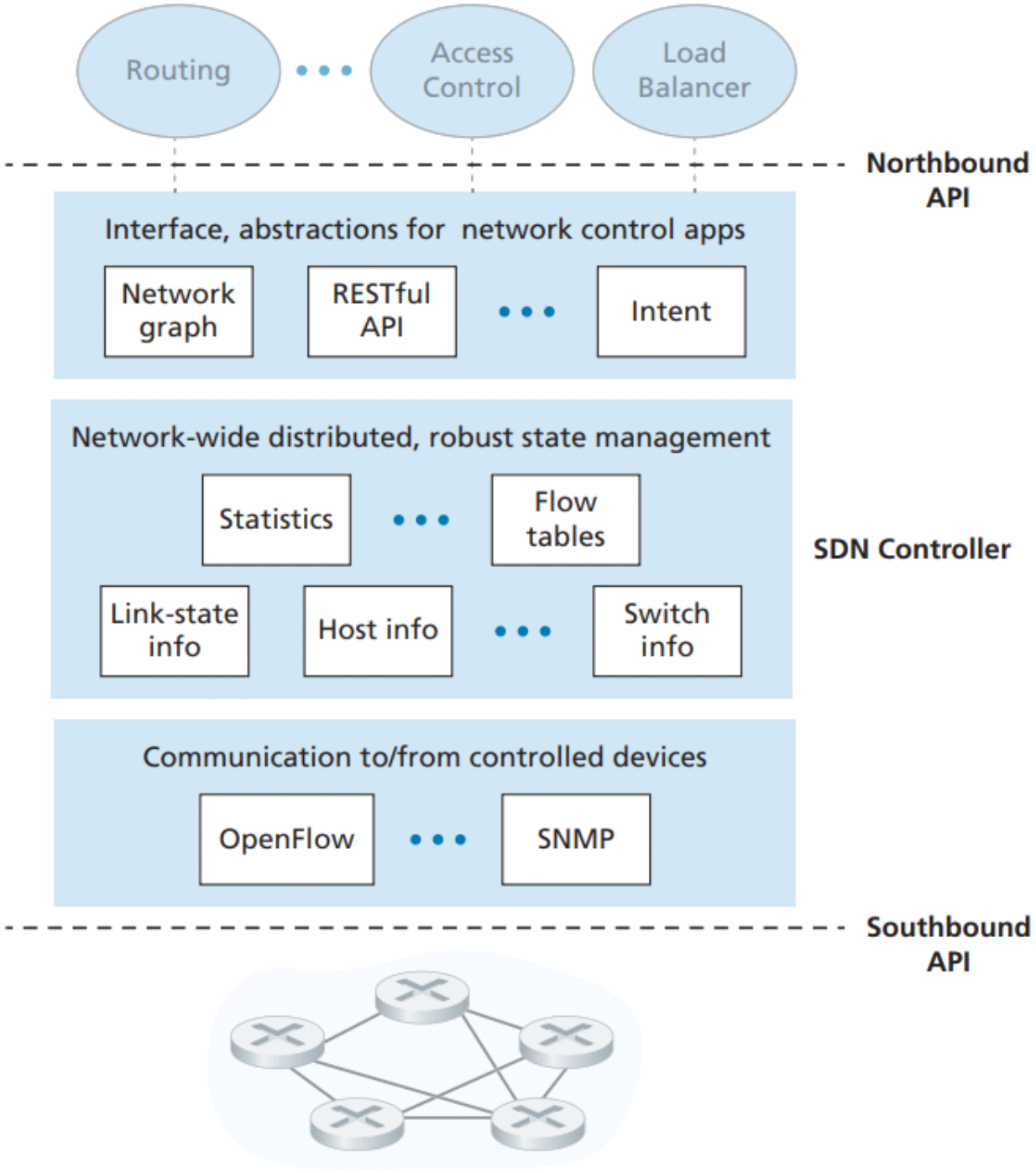


Figure 5.15 ♦ Components of an SDN controller

OpenFlow Protocol

- The OpenFlow operates between an SDN controller and an SDN-controlled switch or other device implementing the OpenFlow API
- Messages flowing from the controller to the controlled switch
  - Configuration: query and set a switch's configuration parameter
  - Modify-State: add/delete or modify entries in the flow table
  - Read-State: collect statistics and counter values from the switch's flow table and

ports
    - Send-Packet: send a specific packet out of a specified port at the controlled switch
- Messages flowing from the controlled switch to the controller
    - Flow-Removed: informs the controller that a flow table entry has been removed, for example by a timeout or as a result of a received modify-state message
    - Port-Status: inform the controller of a change in port status
    - Packet-in: send packets to the controller for additional processing

Chapter 5.6 - ICMP: The Internet Control Message Protocol

Tuesday, July 25, 2023

ICMP

- The **Internet Control Message Protocol (ICMP)** is used by hosts and routers to communicate network-layer information to each other
- The most typical use is for error reporting
- ICMP lies juts above IP, as ICMP messages are carried inside IP datagrams
    - ICMP messages are carried as IP payload, just as TCP or UDP segments are carried as IP payload
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram causing the error
    - Ping program sends an ICMP type 8 code 0 message to the specified host
    - The destination host sends back a type 0 code 0 ICMP echo reply
- Traceroute is also implemented with ICMP messages
    - Traceroute in the source sends a series of ordinary IP datagrams to the destination
    - The first has a TTL of 1, the second of 2, the third of 3
    - When the nth datagram arrives at the nth router, the nth router observes that the TTL of the datagram has just expired
    - The router discards the datagram and sends an ICMP warning message to the source (type 11 code 0) - containing the name of the router and its IP address
    - When this messages arrives back at the source, the source obtains the RTT from the timer and the name and IP of the nth router.
    - One of the datagrams will eventually make it to the destination host
    - The destination host sends a port unreachable ICMP message (type 3 code 3) back to the source
    - When the source host receives this ICMP message, it stops sending additional probe packets

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 5.19 ♦ ICMP message types

Chapter 5 Self Test

Sunday, August 6, 2023

Chapter 5.1 - Introduction to the Control Plane

1. What are some differences between per-router control and logically centralized control?

Chapter 5.2 - Routing Algorithms

1. What are some differences between centralized and decentralized routing algorithms?
2. What are some differences between static and dynamic routing algorithms?
3. In an Link-State (LS) algorithm, is the input the information about the entire network or just the neighbouring nodes?
4. Describe Dijkstra's algorithm.
5. What is the worst-case complexity of LS without using new data structure?
6. What is worst-case complexity of LS using a heap data structure?
7. Describe oscillation in Figure 5.5.
8. Why is the distance vector (DV) algorithm distributed, iterative and asynchronous?
9. In DV, what are the three types of routing information a node x maintains?
10. What are two real routing protocols that use DV algorithm?
11. In Figure 5.7, describe what happens to y and z's distance table entries to x for both scenarios.
12. For (b), how many iterations will this persist? Which node will break out of this loop?
13. What is poisoned reverse? How does the it solve the above problem?
14. Does the poisoned reverse technique solve the general count-to-infinity problem? If not, under what circumstances does it fail to solve the problem?
15. Compare LS and DV in terms of
    a. Message complexity
    b. Speed of convergence (time complexity)
    c. Robustness - what can happen if a router fails, misbehaves or is sabotaged?

Chapter 5.3 - OSPF

1. Why is the view of a homogenous set of routers all executing the same routing algorithm is simplistic?
2. What is an autonomous system (AS)?
3. Does OSPF use an LS or DV protocol?
4. Is OSPF an intra-AS routing protocol or inter-AS routing protocol?
5. Does the router use the network topology of the entire AS or just information from the neighbours?
6. When does a router broadcast routing information to other routers?
7. Does OSPF need to implement reliable data transfer itself. Why or why not?
8. What happens when the exchanges between OSPF routers can be authenticated?
9. How are boundary borders different from border routers? What are their functions?
10. What is the function of the backbone area?
11. True or False: a packet can be routed to the backbone and the destination area through only non-border routers.

Chapter 5.4 - BGP

1. Is BGP an intra-AS routing protocol or inter-AS routing protocol?
2. Does BGP use an LS or DV algorithm?
3. What is a prefix in BGP?
4. How many IP addresses are in 128.16.68/22?
5. What kind of entries are in a router's forwarding table?
6. What are the two things that BGP allow each router to determine?
7. What is the difference between a gateway router and an internal router?
8. In Figure 5.9, describe the process of AS3 advertising about the existence of prefix. What does AS3 send to AS2? What does AS2 send to AS1?
9. Repeat above using both eBGP and iBGP and specifying routers.
10. What is the AS-PATH attribute? What can it do?
11. What is the NEXT-HOP attribute?
12. What is hot potato routing? Why is it called a selfish algorithm?
13. Describe the BGP's route-selection algorithm (elimination rules from highest priority to lowest priority)

Chapter 5.5 - SDN Control Plane

1. List four characteristics of an SDN architecture
2. What are the functions of:
    a. Communication layer?
    b. Network-wide state-management layer?
    c. Network-control application layer?
3. What are the functions of the Northbound and the Southbound APIs?
4. Why can the SDN controller considered to be "logically centralized"
5. Does OpenFlow work in the Northbound or Southbound API?
6. For the Southbound API, what are some types of messages that flow from the controller?
7. For the Southbound API, what are some types of messages that flow into the controller?

Chapter 5.6 - ICMP

1. In which layer does ICMP reside?
2. What are contained in ICMP messages?
3. In traceroute, what ICMP message does the source receive when the nth datagram arrives at the nth router?
4. How does traceroute know when to stop? What ICMP message will it receive from the destination host?

Chapter 5 Review Problems

Friday, July 28, 2023

these algorithms find application in both per-router control and in SDN control. These algorithms are the basis for two widely deployed Internet routing protocols, OSPF and BGP, that we covered in Sections 5.3 and 5.4. We covered the SDN approach to the network-layer control plane in Section 5.5, investigating SDN network-control applications, the SDN controller, and the OpenFlow protocol for communicating between the controller and SDN-controlled devices. In Sections 5.6 and 5.7, we covered some of the nuts and bolts of managing an IP network: ICMP (the Internet Control Message Protocol) and network management using SNMP and NETCONF/YANG.

Having completed our study of the network layer, our journey now takes us one step further down the protocol stack, namely, to the link layer. Like the network layer, the link layer is part of each and every network-connected device. But we will see in the next chapter that the link layer has the much more localized task of moving packets between nodes on the same link or LAN. Although this task may appear on the surface to be rather simple compared with that of the network layer's tasks, we will see that the link layer involves a number of important and fascinating issues that can keep us busy for a long time.

Homework Problems and Questions

Chapter 5 Review Questions

SECTION 5.1

R1. What is meant by a control plane that is based on per-router control? In such cases, when we say the network control and data planes are implemented "monolithically," what do we mean?

R2. What is meant by a control plane that is based on logically centralized control? In such cases, are the data plane and the control plane implemented within the same device or in separate devices? Explain.

SECTION 5.2

R3. Compare and contrast the properties of a centralized and a distributed routing algorithm. Give an example of a routing protocol that takes a centralized and a decentralized approach.

R4. Compare and contrast static and dynamic routing algorithms.

R5. What is the "count to infinity" problem in distance vector routing?

R6. How is a least cost path calculated in a decentralized routing algorithm?

SECTIONS 5.3–5.4

R7. Why are different inter-AS and intra-AS protocols used in the Internet?

R8. True or false: When an OSPF route sends its link state information, it is sent only to those nodes directly attached neighbors. Explain.

R9. What is meant by an *area* in an OSPF autonomous system? Why was the concept of an area introduced?

R10. Define and contrast the following terms: *subnet, prefix*, and *BGP route*.

R11. How does BGP use the NEXT-HOP attribute? How does it use the AS-PATH attribute?

R12. Describe how a network administrator of an upper-tier ISP can implement policy when configuring BGP.

R13. True or false: When a BGP router receives an advertised path from its neighbor, it must add its own identity to the received path and then send that new path on to all of its neighbors. Explain.

SECTION 5.5

R14. Describe the main role of the communication layer, the network-wide state-management layer, and the network-control application layer in an SDN controller.

R15. Suppose you wanted to implement a new routing protocol in the SDN control plane. At which layer would you implement that protocol? Explain.

R16. What types of messages flow across an SDN controller's northbound and southbound APIs? Who is the recipient of these messages sent from the controller across the southbound interface, and who sends messages to the controller across the northbound interface?

R17. Describe the purpose of two types of OpenFlow messages (of your choosing) that are sent from a controlled device to the controller. Describe the purpose of two types of Openflow messages (of your choosing) that are send from the controller to a controlled device.

R18. What is the purpose of the service abstraction layer in the OpenDaylight SDN controller?

SECTIONS 5.6–5.7

R19. Names four different types of ICMP messages

R20. What two types of ICMP messages are received at the sending host executing the *Traceroute* program?

R21. Define the following terms in the context of SNMP: *managing server, managed device, network management agent and MIB.*

R22. What are the purposes of the SNMP *GetRequest* and *SetRequest* messages?

R23. What is the purpose of the SNMP trap message?

R1
- Per-router control means that there is a routing algorithm at each router
- Monolithic means the functions of both control plane and data plane are implemented in every switch
- Each router has a routing component that communicates with the routing components in other routers to compute the values for its forwarding table

R2
- ~~Logically centralized control means that the routers only perform the forwarding actions~~
- The data plane and control plane are implemented in separate devices
- Logically centralized control means that a logically central routing controller computes and distribute the forwarding table to be used by each and every router
- The control plane is implemented in a central server or multiple servers
- The data plane is implemented in each router

R3
- Centralized
  - Routers depend on the information of the entire network
  - Example: OSPF
- Distributed
  - Routers depend on the information of their neighbours only
  - Example: BGP

- Centralized
  - Uses complete, global knowledge about the network, including the connectivity between all nodes and all links' costs
  - The actual calculation can be run at one site or replicated in the routing component of each and every router
- Distributed
  - No node has the completed information about the costs of all links
  - A node gradually calculates the least-cost path through an iterative process of calculation and information exchange

R4
- Static
  - The least-cost routes are rarely recomputed
- Dynamic
  - The least-cost routes are recomputed whenever there is a change in the network

R5

R6
- A router sends information about itself to neighbouring routers
- It then receives information from the neighbouring routers
- Using this information, the router recomputes the least-cost paths
- This process is repeated if there is change to the computed least-cost paths or there is a change in the cost of one of the links

R7
- Because different companies/organizations might want to use different protocols
- Policy
- Scale
- Performance

R8
- False because OSPF is link-state algorithm, meaning it needs the information of the entire network

R9
- An area is a part of an AS in OSPF
- It is introduced because having too many routers in an AS can slow down the routing algorithm
- By dividing an AS into areas, the non-border routers only need to route to the border routers?
- An area refers to a set of routers, in which each router broadcasts its link state to all routers in the same set

R10
- A prefix is a part of the IP address. It is written in the form a.b.c.d/x
- ~~A subnet is a network where all devices have a common prefix?~~
- ~~A BGP route is a sequence of paths between the ASs to get from the source to the destination~~
- A subnet is a portion of a larger network that does not contain a router
- A BGP route is a prefix along with its attributes

R11
- The NEXT-HOP refers to the gateway router in the next AS on the AS-PATH
- The AS-PATH refers to the path consisting of names of ASs to reach a prefix.
- The AS-PATH can be used for cycle detection. If an AS finds its name in the AS-PATH, the packet will be dropped
- The NEXT-HOP attribute is used when a router configures its forwarding table

R12
- It can configure BGP so that when it receives an advertised path for a prefix that is not its own customer, it skips the step of adding its identity to the received path before advertising the path to its neighbours

R13
- False
- If the router does not want the neighbours learn that there is a path via itself, then it can skip the step of adding its own identity to the received path
- Example: when the prefix is not a customer of the current provider network, the provider network does not feel responsible for ensuring that the other provider network can reach this prefix

R14
- Communication layer - communicating with the controlled devices
- Network-wide state-management layer - providing up-to-date information about the state of the hosts, links, switches and other SDN-controlled devices; also maintaining a copy of the flow tables of the controlled devices
- Network-control application layer - routing protocols, load balancing, etc.

R15
- It would be implemented in the network-control application layer

R16
- The recipient of the messages sent across the southbound interface are the switches (controlled devices)
- The senders of the messages across the northbound interface are network control applications

R21. Define the following terms in the context of SNMP: *managing server, managed device, network management agent and MIB.*

R22. What are the purposes of the SNMP *GetRequest* and *SetRequest* messages?

R23. What is the purpose of the SNMP trap message?

- It would be implemented in the network-control application layer

R16

- The recipient of the messages sent across the southbound interface are the switches (controlled devices)
- The senders of the messages across the northbound interface are network control applications
- Types of message that flow across the southbound
  - Configuration
  - Modify-state
  - Read-state
  - Send-packet
- Types of message that flow across the northbound

R19

- Destination host unreachable
- Ping (request)
- Echo reply (to ping)
- Source quench (congestion control)

R20

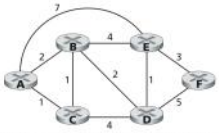
- TTL expired
- Destination port unreachable

Chapter 5 Practice Problems

Wednesday, August 2, 2023 10:16 AM

Problems

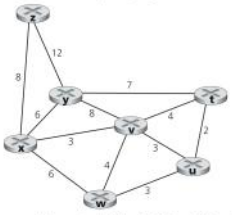
P1. Consider the figure below.



Enumerate all paths from A to D that do not contain any loops

P2. Repeat Problem P1 for paths from C to D, B to F, and C to F.

P3. Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.

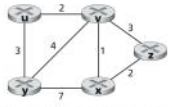


VideoNote
Dijkstra's algorithm: discussion and example

P4. Consider the network shown in Problem P3. Using Dijkstra's algorithm, and showing your work using a table similar to Table 5.1, do the following:

a. Compute the shortest path from t to all network nodes.
b. Compute the shortest path from u to all network nodes.
c. Compute the shortest path from v to all network nodes.
d. Compute the shortest path from w to all network nodes.
e. Compute the shortest path from y to all network nodes.
f. Compute the shortest path from z to all network nodes.

P5. Consider the network shown below. Assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.



P6. Consider a general topology (that is, not the specific network shown above) and a synchronous version of the distance-vector algorithm. Suppose that at each iteration, a node exchanges its distance vectors with its neighbors and receives their distance vectors. Assuming that the algorithm begins with each node knowing only the costs to its immediate neighbors, what is the maximum number of iterations required before the distributed algorithm converges? Justify your answer.

P7. Consider the network fragment shown below. x has only two attached neighbors, w and y. w has a minimum-cost path to destination u (illustrated with the dotted line through the remaining network) of 9, and y has a minimum-cost path to u of 11. The complete paths from w and y to u (and between w and y) are pictured with dotted lines, as they are irrelevant to the solution.

a. Give x's distance vector for destinations w, y, and u.
b. Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.
c. Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that x will *not* inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

P8. Consider the three-node topology shown in Figure 5.6. Rather than having the link costs shown in Figure 5.6, the link costs are $c(x,y) = 3$, $c(y,z) = 6$, $c(z,x) = 4$. Compute the distance tables after the initialization step and after each iteration of a synchronous version of the distance-vector algorithm (as we did in our earlier discussion of Figure 5.6).

P9. Can the *poisoned reverse* solve the general count-to-infinity problem? Justify your answer.

P10. Argue that for the distance-vector algorithm in Figure 5.6, each value in the distance vector $D(x)$ is non-increasing and will eventually stabilize in a finite

P3.

step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1,u	∞	∞
1	ux	2, u	4, x		2, x	∞
2	uxy	2, u	3, y			4, y
3	uxyv		3, y			4, y
4	uxyvw					4, y
5	uxyvwz					

Table 5.1 ♦ Running the link-state algorithm on the network in Figure 5.3

Step	N'	t	u	v	w	y	z
0	{x}			3, x	6, x	6, x	8, x
1	{x, v}	7, v	6, v	3, x	6, x	6, x	8, x
2	{x, v, u}	7, v	6, v	3, x	6, x	6, x	8, x
3	{x, v, u, w}	7, v	6, v	3, x	6, x	6, x	8, x
4	{x, v, u, w, y}	7, v	6, v	3, x	6, x	6, x	8, x
5	{x, v, u, w, y, t}	7, v	6, v	3, x	6, x	6, x	8, x
6	N	7, v	6, v	3, x	6, x	6, x	8, x

P4.

P5.

	To u	To v	To x	To y	To z
From v	∞	∞	∞	∞	∞
From x	∞	∞	∞	∞	∞
From z	∞	3	2	∞	0

	To u	To v	To x	To y	To z
From v	2	0	1	4	3
From x	∞	1	0	7	2
From z	5	3	2	7	0

	To u	To v	To x	To y	To z
From v	2	0	1	4	3
From x	3	1	0	5	2
From z	5	3	2	7	0

P6.

~~The maximum number of iterations required is the length of the longest path in the topology.~~

Let d be the diameter of the network, that is, the length of the longest path without loops between any two nodes in the network. After $d - 1$ iterations, all nodes will know the shortest path cost of d or fewer hops to all other nodes

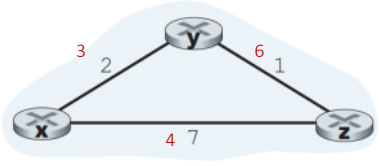
P7.

a.

w	5
y	4
u	15

b. Change $c(x, y)$ to 3
c. Change $c(x, y)$ to 5

P8.



Node x's table

	To x	To y	To z
From x	0	3	4
From y	∞	∞	∞
From z	∞	∞	∞

P10. Argue that for the distance-vector algorithm in Figure 5.6, each value in the distance vector $D(x)$ is non-increasing and will eventually stabilize in a finite number of steps.

P11. Consider Figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows: $c(x,y) = 4$, $c(x,z) = 50$, $c(y,w) = 1$, $c(z,w) = 1$, $c(y,z) = 3$. Suppose that poisoned reverse is used in the distance-vector routing algorithm.

a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?

b. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.

c. How do you modify $c(y,z)$ such that there is no count-to-infinity problem at all if $c(y,x)$ changes from 4 to 60?

P12. What is the message complexity of LS routing algorithm?

P13. Will a BGP router always choose the loop-free route with the shortest ASpath length? Justify your answer.

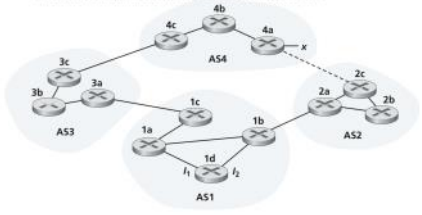
P14. Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is *no* physical link between AS2 and AS4.

a. Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?

b. Router 3a learns about x from which routing protocol?

c. Router 1c learns about x from which routing protocol?

d. Router 1d learns about x from which routing protocol?



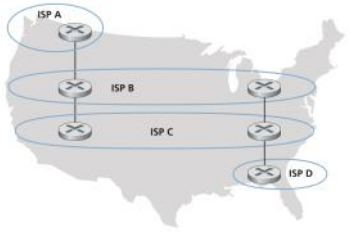
P15. Referring to the previous problem, once router 1d learns about x it will put an entry (x, I) in its forwarding table.

a. Will I be equal to I_1 or I_2 for this entry? Explain why in one sentence.

b. Now suppose that there is a physical link between AS2 and AS4, shown by the dotted line. Suppose router 1d learns that x is accessible via AS2 as well as via AS3. Will I be set to I_1 or I_2? Explain why in one sentence.

c. Now suppose there is another AS, called AS5, which lies on the path between AS2 and AS4 (not shown in diagram). Suppose router 1d learns that x is accessible via AS2 AS5 AS4 as well as via AS3 AS4. Will I be set to I_1 or I_2? Explain why in one sentence.

P16. Consider the following network. ISP B provides national backbone service to regional ISP A. ISP C provides national backbone service to regional ISP D. Each ISP consists of one AS. B and C peer with each other in two places using BGP. Consider traffic going from A to D. B would prefer to hand that traffic over to C on the West Coast (so that C would have to absorb the cost of carrying the traffic cross-country), while C would prefer to get the traffic via its East Coast peering point with B (so that B would have carried the traffic across the country). What BGP mechanism might C use, so that B would hand over A-to-D traffic at its East Coast peering point? To answer this question, you will need to dig into the BGP specification.



Node x's table

	To x	To y	To z
From x	0	3	4
From y	∞	∞	∞
From z	∞	∞	∞

	To x	To y	To z
From x	0	3	4
From y	3	0	6
From z	4	6	0

Node y's table

	To x	To y	To z
From x	∞	∞	∞
From y	3	0	6
From z	∞	∞	∞

	To x	To y	To z
From x	0	3	4
From y	3	0	6
From z	4	6	0

Node z's table

	To x	To y	To z
From x	∞	∞	∞
From y	∞	∞	∞
From z	4	6	0

	To x	To y	To z
From x	0	3	4
From y	3	0	6
From z	4	6	0

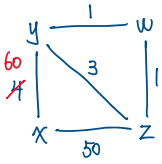
P9.

No, the poisoned reverse does not solve the general count-to-infinity problem. Loops containing three or more nodes will not be detected by poisoned reverse.

P10.

Updating of a node's distance vectors is based on the Bellman-Ford equation (i.e. only decreasing those values in its distance vector). Since the costs are finite, then eventually distance vectors will be stabilized in finite steps.

P11.



a. From w to x: 5
From y to x: 4
From z to x: 6

b. At t=0, x and y detect link cost change; y routes to x via w (1+5=6) and informs w that its cost to x is now infinity
At t=1, w routes to x via z (1+6=7); z routes to x via w (1+5=6). They tell each other that their costs to x are infinity
At t=2 , y routes to x via w (1+7=8); w routes to x via y; z routes to x via y (3+6=9)
...
There seems to be a loop?

P12.

The message complexity is $O(|N||E|)$

P13.

No, because there might be policies set by network administrators and they have higher priorities in the route-selection algorithm.

P14.

a. eBGP
b. ~~OSPF~~ iBGP
c. eBGP
d. ~~RIP~~ iBGP

P15.

a. I_1 because there is no path to AS4 via AS2
b. I_2 because although 1d can get to AS4 via AS2 or AS3, it will choose AS2 due to hot potato routing (get out of the AS as quickly as possible)
c. I_1 because the length of AS-PATH has a higher priority than hot potato routing

P17. In Figure 5.13, consider the path information that reaches stub networks W, X, and Y. Based on the information available at W and X, what are their respective views of the network topology? Justify your answer. The topology view at Y is shown below.

Figure 5.13 ♦ A simple BGP policy scenario

P18. Consider Figure 5.13. B would never forward traffic destined to Y via X based on BGP routing. But there are some very popular applications for which data packets go to X first and then flow to Y. Identify one such application, and describe how data packets follow a path not given by BGP routing.

P19. In Figure 5.13, suppose that there is another stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, and A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

P20. Suppose ASs X and Z are not directly connected but instead are connected by AS Y. Further suppose that X has a peering agreement with Y, and that Y has a peering agreement with Z. Finally, suppose that Z wants to transit all of Y's traffic but does not want to transit X's traffic. Does BGP allow Z to implement this policy?

X — Y — Z

P21. Consider the two ways in which communication occurs between a managing entity and a managed device: request-response mode and trapping. What are the pros and cons of these two approaches, in terms of (1) overhead, (2) notification time when exceptional events occur, and (3) robustness with respect to lost messages between the managing entity and the device?

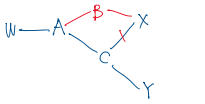
P22. In Section 5.7, we saw that it was preferable to transport SNMP messages in unreliable UDP datagrams. Why do you think the designers of SNMP chose UDP rather than TCP as the transport protocol of choice for SNMP?

Socket Programming Assignment 5: ICMP Ping

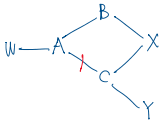
At the end of Chapter 2, there are four socket programming assignments. Here you will find a fifth assignment which employs ICMP, a protocol discussed in this chapter.

P17.

W's view of the network topology.



X's view of the network topology



P18.

One such application would be BitTorrent file sharing.

P19.

A should advertise AW to B, and AV to both B and C.
~~C receives AV, X and Y~~
C receives AV, BAW and BAV

P20.

No.

Since Z wants to transit Y's traffic, Z will send route advertisements to Y. However, Y can readvertise those routes to X. Therefore, there is nothing Z can do to prevent traffic from X to transit through Z.

Chapter 5 Tricky Concepts

Tuesday, August 8, 2023

1. What are RIP, EIGRP and OSPF?
   They are all intra-AS routing protocols
   RIP - classic DV
   EIGRP - DV based
   OSPF - LS
2. What is a BGP session?
   Two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection
3. What is a BGP route?
   It consists of prefix + attributes (AS-PATH, NEXT-HOP, …)
4. What is policy-based routing in BGP?
   Gateway router receiving route advertisement uses import policy to accept/decline path
   AS policy also determines whether to advertise path to other neighbouring ASs
5. What are some BGP messages?
   OPEN - opens TCP connection and authenticates sending BGP peer
   UPDATE - advertises new path (or withdraws old)
   KEEPALIVE - keeps connection alive in the absence of UPDATES
   NOTIFICATION - report errors or close connection
6. Why a logically centralized control plane?
   Easier network management, table-based forwarding allows programmable routers
7. How does OpenFlow exchange messages?
   TCP
8. What are three classes of OpenFlow messages?
   Controller-to-switch, asynchronous (switch to controller), symmetric (misc.)
9.

Chapter 6.1 - Introduction to the Link Layer

Tuesday, July 25, 2023

Introduction

- We will refer to any device that runs a link-layer protocol as a **node**
- We also refer to the communication channels that connect adjacent nodes along the communication path as **links**
- Over a given link, a transmitting node encapsulates the datagram in a **link-layer frame** and transmits the frame into the link

The Services Provided by the Link Layer

- Possible services that can be offered by a link-layer protocol
- Framing
    - Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link
- Link access
    - A **medium access control (MAC)** protocol specifies the rules by which a frame is transmitted onto the link
- Reliable delivery
    - Guarantees to move each network-layer datagram across the link without error
    - Often used for links that are prone to high error rates, such as wireless link, with the goal of correcting an error locally
    - Can be considered an unnecessary overhead for low bit-error links
- Error detection and correction
    - Many link-layer protocols provide a mechanism to detect such bit errors
    - The transmitting node includes error-detection bits in the frame, and the receiving node performs an error check
    - Error correction - a receiver not only detects when bit errors have occurred in the frame but also determines exactly where in the frame the errors have occurred (and then corrects these errors)

Where is the Link Layer implemented?

- For the most part, the link layer is implemented on a chip called **network adapter**, aka **network interface controller (NIC)**
- Much of a link-layer controller's functionality is implemented in hardware
- On the sending side, the controller
    - Encapsulates the datagram in a link-layer frame
    - Transmits the frame into the communication link, following the link-access protocol
    - Sets the error-detection bits in the frame header (if the link layer performs error detection)
- On the receiving side, the controller
    - Receives the entire frame
    - Extracts the network-layer datagram
    - Performs error detection (if the link layer performs error detection)
- The software components of the link layer implement higher-level link-layer functionalities
    - Assembles link-layer addressing information
    - Activates the controller hardware
- On the receiving side, link-layer software

    - Responds to controller interrupts
    - Handles error conditions
    - Passes datagrams up to the network layer
- Thus, the link layer is a combination of hardware and software

Chapter 6.3 - Multiple Access Links and Protocols

Tuesday, July 25, 2023

Introduction

- There are two types of network links: **point-to-point links** and **broadcast links**
- Point-to-point link
  - A single sender at one of the link
  - A single receiver at the other end of the link
  - Examples: point-to-point protocol (PPP) and high-level data link control (HDLC)
- Broadcast link
  - Multiple sending and receiving nodes all connected to the same, single, shared broadcast channel
  - The term broadcast is used because whenever one node transmits a frame, the channel broadcasts the frame and each of the other nodes receives a copy
  - Examples: Ethernet and wireless LANs
- How to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel - the **multiple access problem**
- There are protocols called **multiple access protocols** by which nodes regulate their transmission into the shared broadcast channel
- When more than two nodes transmit frames at the same time, all nodes receive multiple frames at the same time; the transmitted frames **collide** at all of the receivers
- When there is a collision, none of the receiving nodes can make any sense of any of the frames that were transmitted
- We can classify multiple access protocols into three categories:
  - Channel partitioning protocols
  - Random access protocols
  - Taking-turns protocols
- Ideally, a multiple access protocol for a channel of rate R bps should satisfy:
  - When only one node has data to send, it has a throughput of R bps
  - When M nodes have data to send, each of these nodes has a throughput of $\frac{R}{M}$ bps
  - The protocol is decentralized - there is no master node that represents a single point of failure for the network
  - The protocol is simple

Channel Partitioning Protocols

- Suppose the channel support N nodes and that the transmission rate of the channel is R bps
- **Time-division multiplexing (TDM)**
  - Divides time into **time frames** and further divides each time into N **time slots**
  - Each time slot is assigned to one of the nodes
  - Whenever a node has a packet to send, it transmits the packet's bits during its assigned time slot in the revolving TDM frame
  - TDM eliminates collisions and is perfectly fair
  - However, a node is limited to an average rate of $\frac{R}{N}$ bps even when it is the only node with packets to send
  - A node must always wait for its turn in the transmission sequence
- **Frequency-division multiplexing (FDM)**

    - Divides the R bps channel into different frequencies and assigns each frequency to one of the N nodes
    - Thus creates N smaller channels of $\frac{R}{N}$ bps out of the single, larger R bps channel
    - FDM avoids collisions and divides the bandwidth fairly among the N nodes
    - However, a node is limited to a bandwidth of $\frac{R}{N}$ bps, even when it is the only node with packets to send
- Code division multiple access (CDMA)
    - Assigns a different code to each node
    - Each node uses its unique code to encode the data bits it sends
    - If the codes are chosen carefully, different nodes can transmit simultaneously and have their respective receivers correctly receive a sender's encoded data bits

Random Access Protocols

- A transmitting node always transmits at the full rate of the channel
- When there is a collision, each node involved in the collision repeatedly retransmit its frame until its frame gets through without a collision
- It waits a random delay before retransmitting the frame

Carrier Sense Multiple Access (CSMA)

- Two important rules are carrier sensing and collision detection
- Carrier sensing
    - If a frame from another node is currently being transmitted, a node waits until it detects no transmissions for a short amount of time and then begins transmission
- Collision detection
    - If a node detects that another node is transmitting an interfering frame, it stops transmitting and waits a random amount of time before repeating the cycle
- These two rules are embodied in the family of carrier sense multiple access (CSMA) and CSMA with collision detection (CSMA/CD) protocols
- End-to-end channel propagation delay of a broadcast channel will play a crucial role in determining its performance
    - The longer the delay, the larger the chance that carrier-sensing node is not yet able to sense a transmission that has already begun at another node in the network

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

- The operation from the perspective of an adapter (in a node)
    1. Obtains a datagram from the network layer, prepares a link-layer frame and puts the frame adapter buffer
    2. If the adapter senses that the channel is idle, it starts to transmit the frame. Otherwise it waits until it senses no transmission and then starts to transmit the frame
    3. While transmitting, the adapter monitors for the presence of signal energy coming from other adapters
    4. If the adapter detects signal energy from other adapters while transmitting, it aborts the transmission
    5. After aborting, the adapter waits a random amount of time and then returns to step 2
- How to choose the amount of time to wait?
- We would like an interval that is short when the number of colliding nodes is small, and long when the number of colliding nodes is large

- The **binary exponential backoff** algorithm satisfies this
  - When transmitting a frame that has n collisions, a node chooses the value of K at trandom from $\{0,1,2,\ldots,2^n-1\}$
  - The more collisions experienced by a frame, the larger the interval from which K will be chosen
  - For Ethernet, the amount of time a node waits is $K \cdot 512$ bit times (i.e., K times the amount of time needed to send 512 bits into the Ethernet), and the maximum value that n can take is 10

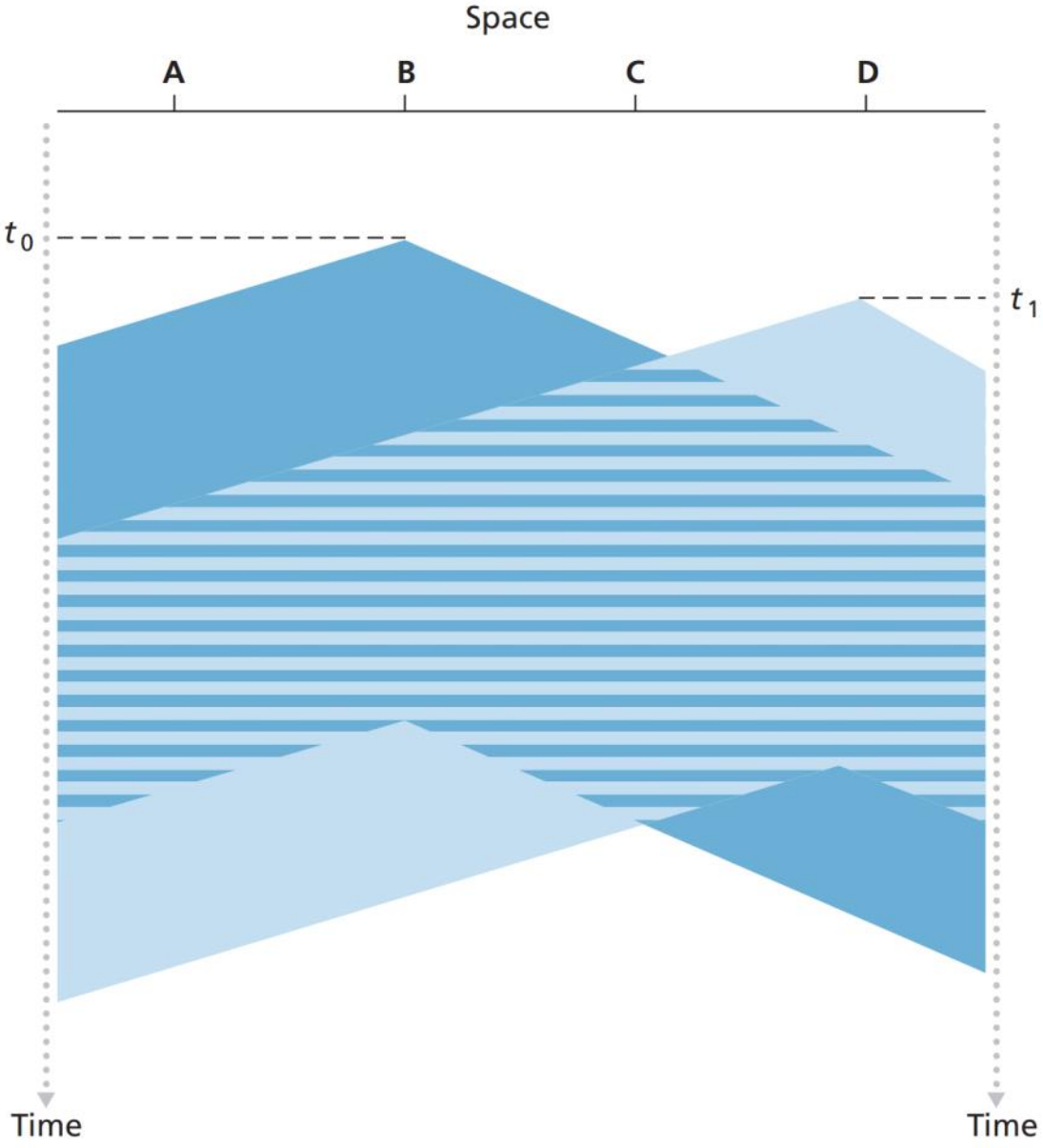


Figure 6.12 ♦ Space-time diagram of two CSMA nodes with colliding transmissions

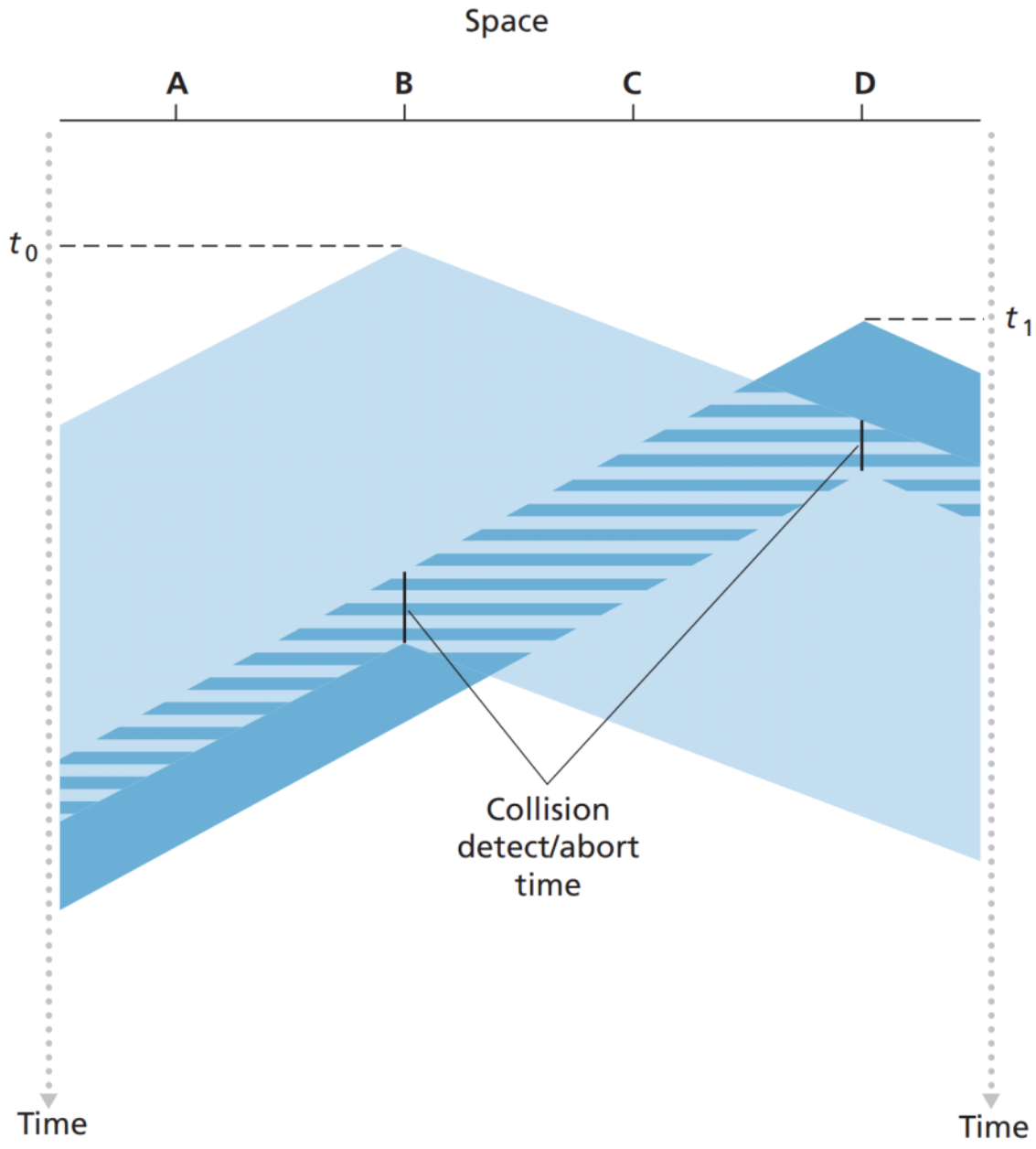


Figure 6.13 ♦ CSMA with collision detection

CSMA/CD Efficiency

- The **efficiency of CSMA/CD** is the long-run fraction of time during which frames are being transmitted on the channel without collisions when there is a large number of active nodes
- Let d_{prop} denote the maximum time it takes signal energy to propagate between any two adapters
- Let d_{trans} be the time to transmit a maximum-size frame
- An approximation is

$$\text{Efficiency} = \frac{1}{1 + 5d_{prop}/d_{trans}}$$

Taking-Turns Protocols

- Two of the more important protocols are **polling protocol** and **token-passing protocol**
- Polling protocol
    - Requires one of the nodes to be designated as a master
    - The master node polls each of the nodes in round-robin fashion
    - The polling protocol eliminates the collisions and empty slots that plague random access protocols
    - Advantages
        - Eliminates the collisions and empty slots
        - Much higher efficiency
    - Disadvantages
        - Introduces a polling delay - the amount of time required to notify a node that it can transmit
        - Single point of failure - If the master node fails, the entire channel fail
- Token-passing protocol
    - There is no master node
    - A small, special-purpose frame known as a **token** is exchanged among the nodes in some fixed order
    - A node holds onto the token only if it has some frames to transmit. Otherwise, it immediately forwards the token to the next node
    - Advantages
        - Decentralized
        - Highly efficient
    - Disadvantages
        - The failure of one node can crash the entire channel
        - If a node accidentally neglects to release the token, then some recovery procedure must be invoked to get the token back in circulation

Chapter 6.4 - Switched Local Area Networks

Tuesday, July 25, 2023

MAC Addresses

- It is not hosts and routers that have link-layer addresses but rather their adapters (network interfaces) that have link-layer addresses
- A link-layer address is called a **LAN address**, a **physical address**, or a **MAC address**
- For most LANs, the MAC address is 6 bytes (48 bits) long, giving 2^{48} possible MAC addresses
- No two adapters have the same address
- IEEE manages the MAC address space
- An adapter's MAC address has a flat structure and does not change no matter where the adapter goes
- When an adapter wants to send a frame to some destination adapter, the sending adapter inserts the destination adapter's MAC address into the frame and then sends the frame into the LAN
- Sometimes a sending adapter wants all the other adapters on the LAN to receive and process the frame it is about to send
    - In this case, the sending adapter inserts a special MAC **broadcast address** into the destination address field of the frame
    - For LANs that use 6-byte addresses, the broadcast address is FF-FF-FF-FF-FF-FF

Address Resolution Protocol (ARP)

- The **Address Resolution Protocol (ARP)** translates between network-layer addresses and link-layer addresses
- An ARP module in the sending host takes any IP address on the same LAN as input, and returns the corresponding MAC address
- ARP is analogous to DNS, which resolves host names to IP addresses
- However, DNS resolves host names for hosts anywhere in the Internet, whereas ARP resolves IP addresses only for hosts and router interfaces on the same subnet
- Each host and router has an **ARP table** in its memory, which contains mappings of IP addresses to MAC addresses
    - The TTL value indicates when each mapping will be deleted from the table
    - A table does not necessarily contain an entry for every host and router on the subnet since some may have never been in the table and others may have expired
- What if the ARP table does not have an entry for the destination?
    - The sender constructs a special packet called an **ARP packet**
    - This packet includes the sending and receiving IP and MAC addresses
    - The purpose is to query all other hosts and routers on the subnet to determine corresponding the MAC address
    - The adapter should send the packet to the MAC broadcast address, FF-FF-FF-FF-FF-FF
    - The frame containing the ARP query is received by all the other adapters on the subnet, and each passes the ARP packet to its ARP module
    - The one with a match of IP address sends back a response ARP packet with the desired mapping
- Thing to note about the ARP protocol
    - The query is sent within a broadcast frame, whereas the response ARP is sent within a standard frame

- ARP is **plug-and-play**; the table gets built automatically and does not need to be configured
- ARP is best considered a protocol that straddles the boundary between the link and network layers

Sending a datagram off the Subnet - Example (Figure 6.19)

- How does a host on Subnet 1 send a datagram to a host on Subnet 2?
- Suppose that host 111.111.111.111 wants to send an IP datagram to host 222.222.222.222
- If the MAC address is set to that of the adapter for 222.222.222.222, the destination address will not match the MAC address of any adapter on Subnet 1
  - No adapter on Subnet 1 will send the IP datagram up to its network layer
- The appropriate MAC address for the frame is the address of the adapter for router interface 111.111.111.110
  - This is obtained using ARP
- The router passes the frame to the network layer and determines the correct interface on which the datagram is to be forwarded using a forwarding table
- This interface then passes the datagram to its adapter, which encapsulates the datagram in a new frame and sends the frame into Subnet 2
- This time, the destination MAC address is the MAC address of the ultimate destination, and can be obtained using ARP

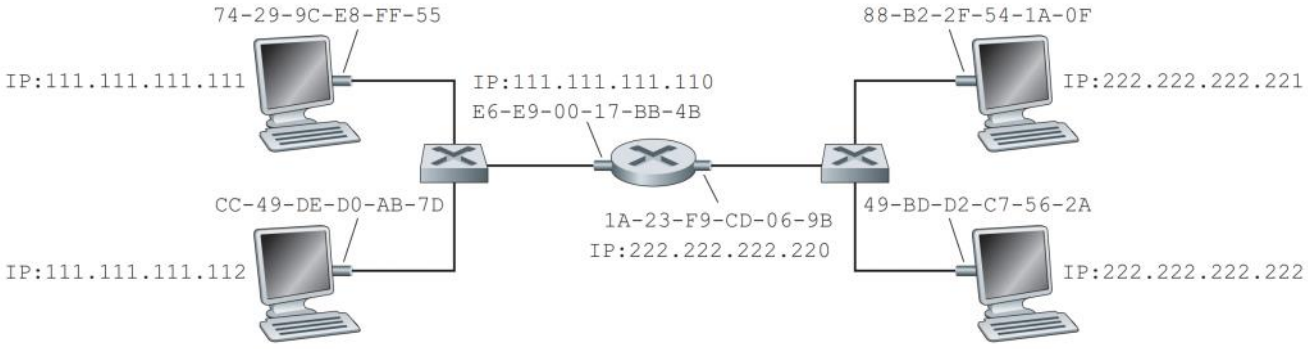


Figure 6.19 ♦ Two subnets interconnected by a router

Ethernet

- By far the most prevalent wired LAN technology
- The first widely deployed high-speed LAN
- Token ring, FDDI and ATM were more complex and expensive than Ethernet
- The original Ethernet LAN used a coaxial bus to interconnect the nodes
  - A broadcast LAN
  - All transmitted frames travel to and are processed by all adapters connected to the bus
- Later replaced with a hub-based star topology
  - The hosts and routers are directly connected to a hub with twisted-pair copper wire
  - A **hub** is a physical-layer device that acts on individual bits rather than frames
  - Ethernet with a hub-based star topology is also a broadcast LAN
- Later replaced with a switch
  - Collision-less

    - Operates only up through layer 2

Ethernet Frame Structure

- Data field (46 to 1500 bytes)
    - Carries the IP datagram
    - The **maximum transmission unit (MTU)** of Ethernet is 1500 bytes
    - If the IP exceeds 1500 bytes, then the host has to fragment the datagram
    - The minimum size of the data field is 46 bytes
    - If the IP datagram is less than 46 bytes, it has to be stuffed to 46 bytes
    - The network layer uses the length field in the IP datagram header to remove the stuffing
- Destination address (6 bytes)
    - MAC address of the destination adapter
- Source address (6 bytes)
    - MAC address of the adapter that transmits the frame onto the LAN
- Type field (2 bytes)
    - Permits Ethernet to multiplex network-layer protocol
    - Analogous to the protocol field in the network-layer datagram and the port number fields in the transport-layer segment
    - They all serve to glue a protocol at one layer to a protocol at the layer above
- Cyclic redundancy check (CRC) (4 bytes)
    - Allow the receiving adapter to detect bit errors in the frame
- Preamble (8 bytes)
    - The Ethernet frame begins with an 80-byte preamble field
    - Each of the first 7 bytes has a value of 10101010; the last byte is 10101011
    - The first 7 bytes of the preamble serve to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock
    - The last 2 bits of the 8th byte alert the receiving adapter that that "important stuff" is about to come
- Ethernet is **connectionless** and **unreliable**
- The receiving adapter discards a frame if it fails the CRC check without sending an acknowledgment
- The lack of reliable transport helps to make Ethernet simple and cheap

Ethernet Technologies

- Ethernet comes in many different flavors, with acronyms such as 10BASE-T, 10BASE-2, 100BASE-T, 1000BASE-LX, 10GBASE-T and 40GBASE-T
- The first part refers to the speed of the standard
- BASE refers to the baseband Ethernet - the physical media only carries Ethernet traffic
- The final part refers to the physical media itself
    - Generally, a "T" refers to twisted-pair copper wires
- The early types of coaxial cables are limited in length to 500 meters
- Longer runs could be obtained by using a **repeater -** a physical-layer device that receives a signal on the input side and regenerates the signal on the output side
- In most installations today, nodes are connected to a switch via point-to-point segments made of twisted-pair coper wires or fiber-optic cables
- Gigabit Ethernet is an extension that offers a raw data rate of 40000 Mbps. The standard
    - Uses a standard Ethernet frame format and is backward compatible with 10BASE-T

and 100BASE-T technologies
    - Allows for point-to-point links and shared broadcast channels
    - Uses CSMA/CD for shared broadcast channels
    - Allows for full-duplex operation at 40 Gbps and in both directions for point-to-point channels

Link-Layer Switches

- The role of a switch is to receive incoming link-layer frames and forward them onto outgoing links
- The switch itself is **transparent** to the hosts and routers in the subnet
    - A host/router addresses a frame to another host/router (rather than addressing the frame to the switch) and sends the frame in to the LAN
    - It is unaware that a switch will be receiving the frame and forwarding it

Forwarding and Filtering

- **Filtering** is the switch function that determines whether a frame should be forwarded to some interface or should just be dropped
- **Forwarding** is the switch function that determines the interfaces to which a frame should be directed, and then moves the frame to those interfaces
- They are done with a **switch table**, with each entry containing
    - A MAC address
    - The switch interface that leads toward that MAC address
    - The time at which the entry was placed in the table
- Switches forward packets based on MAC addresses rather than on IP addresses
- A traditional switch table is constructed in a very different manner from a router's forwarding table
- There are three possible cases upon the arrival of a frame with destination address DD-DD-DD-DD-DD-DD on interface x
    - There is no entry in the table for the DD-DD-DD-DD-DD-DD
        - Forwards copies of the frame to all interfaces except for interface x
        - This is called broadcasting
    - There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface x
        - No need to forward the frame to any other interface since the frame is coming from a LAN segment that contains adapter DD-DD-DD-DD-DD-DD
        - The switch filters by discarding the frame
    - There is an entry in the table, associating DD-DD-DD-DD-DD-DD with interface $y \neq x$
        - The frame will be forwarded to the LAN segment attached to interface y

Self-Learning

- A switch table is built automatically, dynamically, and autonomously
- In other words, switches are **self-learning**
    - The switch table is initially empty
    - For each incoming frame, the switch stores
        - The source MAC address
        - The interface from which the frame arrived
        - The current time
    - The switch deletes an address in the table if no frames are received from that source address after some period of time (the **aging time**)

        - If a PC is replaced by another PC (with a different adapter), the old MAC address will eventually be purged from the switch table
- Switches are **plug-and-play devices** because they require no intervention from a network administrator or user
- Switches are also **full-duplex** because any switch interface can send and receive at the same time

Properties of Link-Layer Switching

- Elimination of collisions
    - Significant performance improvement over LANs with broadcast links
- Heterogeneous links
    - Different links can operate at different speeds and can run over different media
- Management
    - Example: if an adapter malfunctions, a switch can detect the problem and internally disconnect this adapter

Switches vs. Routers

- Switches
    - Advantages
        - Plug-and-play
        - Relatively high filtering and forwarding rate
        - Process frames only up through layer 2
    - Disadvantages
        - To prevent the cycling of broadcast frames, the active topology is restricted to a spanning tree
        - A large switched network requires large ARP tables and generates substantial ARP traffic
        - Susceptible to broadcast storms
- Routers
    - Advantages
        - Network addressing is often hierarchical; packets do not normally cycle through routers
        - A rich topology that includes, for example, multiple active links between Europe and North America
        - Firewall protection against layer-2 broadcast storms
    - Disadvantages
        - Not plug-and-play
        - The IP addresses need to be configured
        - Routers often have a larger per-packet processing time since they have to process up through layer 3
- Small networks consisting of a few hundred hosts have a few LAN segments - switch suffice
    - Localize traffic
    - Increase aggregate throughput
    - No configuration of IP addresses
- Larger networks consisting of thousands of hosts typically include routers within the network
    - More robust isolation of traffic
    - Control broadcast storms

- More "intelligent" routes among the hosts in the network

Chapter 6 Self Test

Sunday, August 6, 2023

Chapter 6.1 - Introduction to the Link Layer

1. What is framing in the link layer?
2. Why is reliable delivery rarely used for low bit-error links?
3. Where is the link layer usually implemented? Software or hardware?

Chapter 6.3 - Multiple Access Links and Protocols

1. What is the difference between point-to-point link and broadcast link? Give some examples of each.
2. What does collision mean?
3. What are the four characteristics that a multiple access protocol for a channel of rate R bps should satisfy?
4. Does TDM have collisions? Is it fair?
5. For TDM, What is the average throughput of a node when it is the only node with packets to send?
6. Answer Q4 for FDM.
7. Answer Q5 for FDM.
8. How does code division multiple access (CDMA) work?
9. What is carrier sensing and collision detection in CSMA?
10. List the steps for CSMA.
11. In the binary exponential backoff algorithm, how is K determined when there are n collisions?
12. For Ethernet, how long does a node wait?
13. Describe the polling protocol. What are some advantages and disadvantages?
14. Describe the token-passing protocol. What are some advantages and disadvantages?

Chapter 6.4 - Switched Local Area Networks

1. True or False: the hosts and routers have link-layer addresses themselves
2. How many possible MAC addresses are here?
3. Does an adapter's MAC address change when its location changes?
4. How does an adapter send a frame to all the other adapters on the LAN? What destination address does it use?
5. What does an ARP module do?
6. What are some similarities and differences between ARP and DNS?
7. List the steps a switch need to perform to learn the MAC address of a host/router that is not in the ARP table. What are the source and destination IP/MAC addresses in each message?
8. Is the query sent within a broadcast frame or standard frame?
9. Is the response sent within a broadcast frame or standard frame?
10. Why is ARP called a plug-and-play protocol?
11. In which layer does ARP reside?
12. In Figure 6.19, describe how the host with IP address 111.111.111.111 sends a datagram to the host with address IP 222.222.222.222

13. What does the type field in the Ethernet frame header do?
14. What does the CRC field in the Ethernet frame header do?
15. What is the value of each of the first 7 bytes in the preamble? What is the value of the last byte? What are their purposes?
16. Why is the switch said to be transparent to the hosts and routers?
17. What kind of entries are there in a switch table?
18. Suppose a frame with destination MAC address D arrives on interface x. What actions do the switch table take in each of the following cases:
    a. There is no entry in the table for the address D.
    b. There is an entry in the table associating the address D with x.
    c. There is an entry in the table, associating the address D with interface $y \neq x$.
19. Why are switches called plug-and-play devices?
20. List three properties of link-layer switching.
21. List some differences between switches and routers.

Chapter 6 Review Problems

Sunday, July 30, 2023

of network-layer routing to interconnect these local nodes. We also learned how multiple virtual LANs can be created on a single physical LAN infrastructure.

We ended our study of the link layer by focusing on how MPLS networks provide link-layer services when they interconnect IP routers and an overview of the network designs for today's massive data centers. We wrapped up this chapter (and indeed the first five chapters) by identifying the many protocols that are needed to fetch a simple Web page. Having covered the link layer, *our journey down the protocol stack is now over*! Certainly, the physical layer lies below the link layer, but the details of the physical layer are probably best left for another course (e.g., in communication theory, rather than computer networking). We have, however, touched upon several aspects of the physical layer in this chapter and in Chapter 1 (our discussion of physical media in Section 1.2). We'll consider the physical layer again when we study wireless link characteristics in the next chapter.

Although our journey down the protocol stack is over, our study of computer networking is not yet at an end. In the following three chapters, we cover wireless networking, network security, and multimedia networking. These four topics do not fit conveniently into any one layer; indeed, each topic crosscuts many layers. Understanding these topics (billed as advanced topics in some networking texts) thus requires a firm foundation in all layers of the protocol stack—a foundation that our study of the link layer has now completed!

Homework Problems and Questions

Chapter 6 Review Questions

SECTIONS 6.1–6.2

R1. What is framing in link layer?

R2. If all the links in the Internet were to provide reliable delivery service, would the TCP reliable delivery service be redundant? Why or why not?

R3. Name three error-detection strategies employed by link layer.

SECTION 6.3

R4. Suppose two nodes start to transmit at the same time a packet of length L over a broadcast channel of rate R. Denote the propagation delay between the two nodes as d_{prop}. Will there be a collision if $d_{prop} < L/R$? Why or why not?

R5. In Section 6.3, we listed four desirable characteristics of a broadcast channel. Which of these characteristics does slotted ALOHA have? Which of these characteristics does token passing have?

R1

- Framing is taking a network-layer datagram and encapsulating it into a frame and transmitting the frame over a link to the adjacent node

R2

- No, it would not be redundant because the network layer (specifically, IP) does not provide reliable data transfer
- With IP, datagrams in the same TCP connection can take different routes in the network, and therefore arrive out of order

R4

- ~~There will be no collision~~
- There will be a collision since while a node is transmitting, it will start to receive a packet from the other node

R5

- The four characteristics are
  - Throughput of R bps if there is one node sending packets
  - Throughput of $\frac{R}{M}$ bps if there are M nodes sending packets
  - Decentralized
  - Simple
- Token passing satisfies all characteristics

R6. In CSMA/CD, after the fifth collision, what is the probability that a node chooses $K = 4$? The result $K = 4$ corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

R7. While TDM and FDM assign time slots and frequencies, CDMA assigns a different code to each node. Explain the basic principle in which CDMA works.

R8. Why does collision occur in CSMA, if all nodes perform carrier sensing before transmission?

SECTION 6.4

R9. How big is the MAC address space? The IPv4 address space? The IPv6 address space?

R10. Suppose nodes A, B, and C each attach to the same broadcast LAN (through their adapters). If A sends thousands of IP datagrams to B with each encapsulating frame addressed to the MAC address of B, will C's adapter process these frames? If so, will C's adapter pass the IP datagrams in these frames to the network layer C? How would your answers change if A sends frames with the MAC broadcast address?

R11. IEEE manages the MAC address space, allocating chunks of it to companies manufacturing network adapters. The first half of the bits of the addresses in these chunks are fixed, ensuring that the address space is unique. How long will a chunk last for a company manufacturing 1,000,000 network adapters per year?

R12. For the network in Figure 6.19, the router has two ARP modules, each with its own ARP table. Is it possible that the same MAC address appears in both tables?

R13. What is a hub used for?

R14. Consider Figure 6.15. How many subnetworks are there, in the addressing sense of Section 4.3?

R15. Each host and router has an ARP table in its memory. What are the contents of this table?

R16. The Ethernet frame begins with an 8-byte preamble field. The purpose of the first 7 bytes is to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock. What are the contents of the 8 bytes? What is the purpose of the last byte?

Problems

R6

- There will be $2^5 = 32$ possible values for K, thus the probability is $\frac{1}{32}$
- 1 sec to send 10 Mb $\Rightarrow \frac{512}{10^6}$ sec to send 512 bits $\Rightarrow 4 \cdot \frac{512}{10^6}$ sec

R7

- Each node uses a different code to encode the packets
- If the codes are chosen properly, the receivers will be able to receive the corresponding frames correctly

R8

- In the beginning, suppose that no node is transmitting
- Then suppose all nodes perform carrier sensing and sense no transmission
- All nodes will start transmitting, believing there are no other nodes transmitting at the same time

R9

- The MAC address space is 2^{48}
- The IPv4 address space is 2^{32}
- The IPv6 address space is 2^{128}

R10

- ~~If the destination address is the MAC address of B, C's adapter will not process these frames and pass the IP datagrams to the network layer~~
- If the destination is the MAC broadcast address, C's adapter will process these frames and pass the IP datagrams to the network layer
- If the destination address is the MAC address of B, C's adapter will process these frames but will not pass the IP datagrams to the network layer

R11

- $\frac{2^{24}}{1000000} = 16.8\ yrs$

R12

- ~~Yes, since the left interface for the Subnet 1 and the right interface is for Subnet 2~~
- No, it is impossible because each LAN has its own distinct set of adapters attached to it, with each adapter having a unique LAN address

R13

- The hub can be used as a shared, broadcast channel

first 7 bytes is to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock. What are the contents of the 8 bytes? What is the purpose of the last byte?

Problems

P1. Suppose the information content of a packet is the bit pattern 1010 0111 0101 1001 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme? Your answer should be such that a minimum-length checksum field is used.

- No, it is impossible because each LAN has its own distinct set of adapters attached to it, with each adapter having a unique LAN address

R13

- The hub can be used as a shared, broadcast channel

R14

- ~~There is one subnet~~
- There are two subnets (one internal and one external)

R15

- ARP table stores the mapping between IP addresses and MAC addresses

R16

- The first 7 bytes are 10101010
- The last byte is 10101011, the purpose of which is to tell the receiver that "important stuff" (actual data) is about the come

Chapter 6 Practice Problems

Monday, July 31, 2023

R6. In CSMA/CD, after the fifth collision, what is the probability that a node chooses $K = 4$? The result $K = 4$ corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

R7. While TDM and FDM assign time slots and frequencies, CDMA assigns a different code to each node. Explain the basic principle in which CDMA works.

R8. Why does collision occur in CSMA, if all nodes perform carrier sensing before transmission?

SECTION 6.4

R9. How big is the MAC address space? The IPv4 address space? The IPv6 address space?

R10. Suppose nodes A, B, and C each attach to the same broadcast LAN (through their adapters). If A sends thousands of IP datagrams to B with each encapsulating frame addressed to the MAC address of B, will C's adapter process these frames? If so, will C's adapter pass the IP datagrams in these frames to the network layer C? How would your answers change if A sends frames with the MAC broadcast address?

R11. IEEE manages the MAC address space, allocating chunks of it to companies manufacturing network adapters. The first half of the bits of the addresses in these chunks are fixed, ensuring that the address space is unique. How long will a chunk last for a company manufacturing 1,000,000 network adapters per year?

R12. For the network in Figure 6.19, the router has two ARP modules, each with its own ARP table. Is it possible that the same MAC address appears in both tables?

R13. What is a hub used for?

R14. Consider Figure 6.15. How many subnetworks are there, in the addressing sense of Section 4.3?

R15. Each host and router has an ARP table in its memory. What are the contents of this table?

R16. The Ethernet frame begins with an 8-byte preamble field. The purpose of the first 7 bytes is to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock. What are the contents of the 8 bytes? What is the purpose of the last byte?

Problems

P1. Suppose the information content of a packet is the bit pattern 1010 0111 0101 1001 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme? Your answer should be such that a minimum-length checksum field is used.

P2. For the two-dimensional parity check matrix below, show that:
   a. a single-bit error that can be corrected.
   b. a double-bit error that can be detected, but not corrected.

   0101
   1010
   0101
   1010

P3. Suppose the information portion of a packet contains six bytes consisting of the 8-bit unsigned binary ASCII representation of string "CHKSUM"; compute the Internet checksum for this data.

P4. Compute the Internet checksum for each of the following:
   a. the binary representation of the numbers 1 through 6.
   b. the ASCII representation of the letters C through H (uppercase).
   c. the ASCII representation of the letters c through h (lowercase).

P5. Consider the generator, G = 1001, and suppose that D has the value 11000111010. What is the value of R?

P6. Rework the previous problem, but suppose that D has the value
   a. 01101010101.
   b. 11111010101.
   c. 10001100001.

P7. In this problem, we explore some of the properties of the CRC. For the generator G (= 1001) given in Section 6.2.3, answer the following questions.
   a. Why can it detect any single bit error in data D?
   b. Can the above G detect any odd number of bit errors? Why?

P8. In Section 6.3, we provided an outline of the derivation of the efficiency of slotted ALOHA. In this problem we'll complete the derivation.
   a. Recall that when there are N active nodes, the efficiency of slotted ALOHA is $Np(1-p)^{N-1}$. Find the value of p that maximizes this expression.
   b. Using the value of p found in (a), find the efficiency of slotted ALOHA by letting N approach infinity. *Hint*: $(1-1/N)^N$ approaches $1/e$ as N approaches infinity.

P9. Show that the maximum efficiency of pure ALOHA is $1/(2e)$. *Note*: This problem is easy if you have completed the problem above!

P10. Consider two nodes, A and B, that use the slotted ALOHA protocol to contend for a channel. Suppose node A has more data to transmit than node B, and node A's retransmission probability p_A is greater than node B's retransmission probability, p_B.
   a. Provide a formula for node A's average throughput. What is the total efficiency of the protocol with these two nodes?
   b. If $p_A = 2p_B$, is node A's average throughput twice as large as that of node B? Why or why not? If not, how can you choose p_A and p_B to make that happen?
   c. In general, suppose there are N nodes, among which node A has retransmission probability $2p$ and all other nodes have retransmission probability p. Provide expressions to compute the average throughputs of node A and of any other node.

P11. Suppose four active nodes—nodes A, B, C and D—are competing for access to a channel using slotted ALOHA. Assume each node has an infinite number of packets to send. Each node attempts to transmit in each slot with probability p. The first slot is numbered slot 1, the second slot is numbered slot 2, and so on.
   a. What is the probability that node A succeeds for the first time in slot 4?
   b. What is the probability that some node (either A, B, C or D) succeeds in slot 5?
   c. What is the probability that the first success occurs in slot 4?
   d. What is the efficiency of this four-node system?

P12. Graph the efficiency of slotted ALOHA and pure ALOHA as a function of p for the following values of N:
   a. $N = 10$.
   b. $N = 30$.
   c. $N = 50$.

P13. Consider a broadcast channel with N nodes and a transmission rate of R bps. Suppose the broadcast channel uses polling (with an additional polling node) for multiple access. Suppose the amount of time from when a node completes transmission until the subsequent node is permitted to transmit (that is, the polling delay) is d_{poll}. Suppose that within a polling round, a given node is allowed to transmit at most Q bits. What is the maximum throughput of the broadcast channel?

P14. Consider three LANs interconnected by two routers, as shown in Figure 6.33.
   a. Assign IP addresses to all of the interfaces. For Subnet 1 use addresses of the form 192.168.1.xxx; for Subnet 2 uses addresses of the form 192.168.2.xxx; and for Subnet 3 use addresses of the form 192.168.3.xxx.
   b. Assign MAC addresses to all of the adapters.

P13

$$\frac{NQ}{N(d_{poll}+\frac{Q}{R})} = \frac{Q}{d_{poll}+\frac{Q}{R}} = \frac{Q}{d_{poll}} + R$$

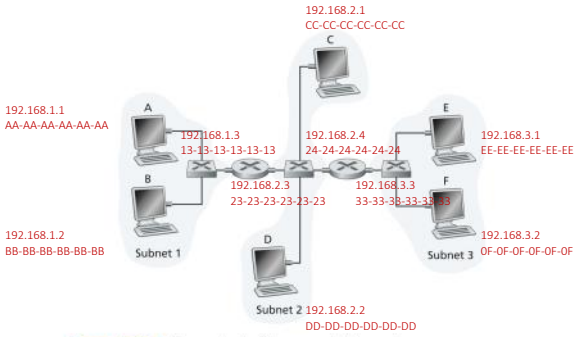


Figure 6.33 • Three subnets, interconnected by routers

c. Consider sending an IP datagram from Host E to Host B. Suppose all of the ARP tables are up to date. Enumerate all the steps, as done for the single-router example in Section 6.4.1.

d. Repeat (c), now assuming that the ARP table in the sending host is empty (and the other tables are up to date).

P15. Consider Figure 6.33. Now we replace the router between subnets 1 and 2 with a switch S1, and label the router between subnets 2 and 3 as R1.

a. Consider sending an IP datagram from Host E to Host F. Will Host E ask router R1 to help forward the datagram? Why? In the Ethernet frame containing the IP datagram, what are the source and destination IP and MAC addresses?

b. Suppose E would like to send an IP datagram to B, and assume that E's ARP cache does not contain B's MAC address. Will E perform an ARP query to find B's MAC address? Why? In the Ethernet frame (containing the IP datagram destined to B) that is delivered to router R1, what are the source and destination IP and MAC addresses?

c. Suppose Host A would like to send an IP datagram to Host B, and neither A's ARP cache contains B's MAC address nor does B's ARP cache contain A's MAC address. Further suppose that the switch S1's forwarding table contains entries for Host B and router R1 only. Thus, A will broadcast an ARP request message. What actions will switch S1 perform once it receives the ARP request message? Will router R1 also receive this ARP request message? If

so, will R1 forward the message to Subnet 3? Once Host B receives this ARP request message, it will send back to Host A an ARP response message. But will it send an ARP query message to ask for A's MAC address? Why? What will switch S1 do once it receives an ARP response message from Host B?

P16. Consider the previous problem, but suppose now that the router between subnets 2 and 3 is replaced by a switch. Answer questions (a)–(c) in the previous problem in this new context.

P17. Recall that with the CSMA/CD protocol, the network adapter waits $K \cdot 512$ bit times after a collision, where K is drawn randomly. For $K = 115$, how long does the adapter wait until returning to Step 2 for:

a. a 10 Mbps broadcast channel?

b. a 100 Mbps broadcast channel?

P18. Suppose nodes A and B are on the same 12 Mbps broadcast channel, and the propagation delay between the two nodes is 316 bit times. Suppose CSMA/CD and Ethernet packets are used for this broadcast channel. Suppose node A begins transmitting a frame and, before it finishes, node B begins transmitting a frame. Can A finish transmitting before it detects that B has transmitted? Why or why not? If the answer is yes, then A incorrectly believes that its frame was successful transmitted without a collision. *Hint:* Suppose at time $t = 0$ bits, A begins transmitting a frame. In the worst case, Atransmits a minimum-sized frame of 512 + 64 bit times. So A would finish transmitting the frame at $t = 512 + 64$ bit times. Thus, the answer is no, if B's signal reaches A before bit time $t = 512 + 64$ bits. In the worst case, when does B's signal reach A?

P19. Suppose nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 245 bit times. Suppose A and B send Ethernet frames at the same time, the frames collide, and then A and B choose different values of K in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at $t = 0$ bit times. They both detect collisions at $t = 245$ t bit times. Suppose $K_A = 0$ and $K_B = 1$. At what time does B schedule its retransmission? At what time does A begin transmission? (*Note*: The nodes must wait for an idle channel after returning to Step 2—see protocol.) At what time does A's signal reach B? Does B refrain from transmitting at its scheduled time?

P20. In this problem, you will derive the efficiency of a CSMA/CD-like multiple access protocol. In this protocol, time is slotted and all adapters are synchronized to the slots. Unlike slotted ALOHA, however, the length of a slot (in seconds) is much less than a frame time (the time to transmit a frame). Let S be the length of a slot. Suppose all frames are of constant length $L = kRS$, where R is the transmission rate of the channel and k is a large integer.

P14c.

Router/Switch	Src IP	Dst IP	Src MAC	Dst MAC	Action
Switch 3	192.168.3.1	192.168.1.2	EE-EE-EE-EE-EE-EE	33-33-33-33-33-33	Send to Router 2
Router 2	192.168.3.1	192.168.1.2	=> 24-24-24-24-24-24	=> 23-23-23-23-23-23	Forward Port 1
Switch 2	192.168.3.1	192.168.1.2	24-24-24-24-24-24	23-23-23-23-23-23	Send to Router 1
Router 1	192.168.3.1	192.168.1.2	=> 13-13-13-13-13-13	=> BB-BB-BB-BB-BB-BB	Forward Port 1
Switch 1	192.168.3.1	192.168.1.2	13-13-13-13-13-13	BB-BB-BB-BB-BB-BB	Send to Host B

P14d.

Router/Switch	Src IP	Dst IP	Src MAC	Dst MAC	Action
Router 2	192.168.3.1	192.168.3.3	EE-EE-EE-EE-EE-EE	FF-FF-FF-FF-FF-FF	Send back an ARP response
Switch 3	192.168.3.1	192.168.1.2	EE-EE-EE-EE-EE-EE	33-33-33-33-33-33	Send to Router 2
Router 2	192.168.3.1	192.168.1.2	=> 24-24-24-24-24-24	=> 23-23-23-23-23-23	Forward Port 1
Switch 2	192.168.3.1	192.168.1.2	24-24-24-24-24-24	23-23-23-23-23-23	Send to Router 1
Router 1	192.168.3.1	192.168.1.2	=> 13-13-13-13-13-13	=> BB-BB-BB-BB-BB-BB	Forward Port 1
Switch 1	192.168.3.1	192.168.1.2	13-13-13-13-13-13	BB-BB-BB-BB-BB-BB	Send to Host B

P15.

a. No, E will not ask Router 1 to help forward the datagram, because E and F are in the same subnet. The source and destination IP are 192.168.3.1 and 192.168.3.2. The source and destination MAC are EE-EE-EE-EE-EE-EE and 0F-0F-0F-0F-0F-0F.

b. No, E will not perform an ARP query to find B's MAC address since they are in different subnets. The source and destination IP are 192.168.3.1 and 192.168.1.2. The source and destination MAC are EE-EE-EE-EE-EE-EE and 33-33-33-33-33-33.

c. Switch 1 will broadcast the request message to Host B and Router 1. It learns that A resides on Subnet 1, so it updates its forwarding table to include an entry for Host A.

Router 1 will receive this message but will not forward to Subnet 3.

Host B will not ask for A's MAC address because it will add the entry to the ARP table.

~~Once Switch 1 receives an ARP response message from Host B, it will store the mapping between B's MAC address and the corresponding interface, and send the ARP response message to A.~~
Once Switch 1 receives B's response message, it will add an entry for B in its forwarding table, and then drop the received frame since A is on the same interface as B (i.e. A and B are on the same LAN segment)

P16.

a. No, E will not ask Router 1 to help forward the datagram, because E and F are in the same subnet. The source and destination IP are 192.168.3.1 and 192.168.3.2. The source and destination MAC are EE-EE-EE-EE-EE-EE and 0F-0F-0F-0F-0F-0F.

b. Yes, E will send an ARP query to find B's MAC address since they are in the same subnet. The source and destination IP are 192.168.3.1 and 192.168.1.2. The source and destination MAC are EE-EE-EE-EE-EE-EE and the broadcast address FF-FF-FF-FF-FF-FF.

c. Switch 1 will broadcast the request message to Host B and Router 1. It learns that A resides on Subnet 1, so it updates its forwarding table to include an entry for Host A.

~~Router 1 will receive this message but will not forward to Subnet 3.~~
Switch 2 also receives this ARP request message, and S2 will broadcast this query packet to all its interfaces

Host B will not ask for A's MAC address because it will add the entry to the ARP table.

~~Once Switch 1 receives an ARP response message from Host B, it will store the mapping between B's MAC address and the corresponding interface, and send the ARP response message to A.~~
Once Switch 1 receives B's response message, it will add an entry for B in its forwarding table, and then drop the received frame since A is on the same interface as B (i.e. A and B are on the same LAN segment)

P17

a. $K \cdot 512 \text{ bit times} = 115 \cdot \frac{512 \text{ bit}}{10^7 \text{ bps}} = 5.9 \times 10^{-3} \text{ s}$

b. $K \cdot 512 \text{ bit times} = 115 \cdot \frac{512 \text{ bit}}{10^8 \text{ bps}} = 5.9 \times 10^{-4} \text{ s}$

P19.

A and B begin transmission at 0
A and B detect collision at t=245
A start retransmitting at t=245
A finish transmitting at t=245+245=490
B start retransmitting t=245+512=757

A and B detect collision and send jam signal at t=245
A and B finish sending jam signal at t=245+48=293
A start retransmitting at t=293
A finish retransmitting at t=293+245=538
B start retransmitting at t=293+512=805
B finishes retransmitting at t=805+245=1050

Suppose there are N nodes, each with an infinite number of frames to send. We also assume that $d_{prop} < S$, so that all nodes can detect a collision before the end of a slot time. The protocol is as follows:

- If, for a given slot, no node has possession of the channel, all nodes contend for the channel; in particular, each node transmits in the slot with probability p. If exactly one node transmits in the slot, that node takes possession of the channel for the subsequent $k - 1$ slots and transmits its entire frame.
- If some node has possession of the channel, all other nodes refrain from transmitting until the node that possesses the channel has finished transmitting its frame. Once this node has transmitted its frame, all nodes contend for the channel.

Note that the channel alternates between two states: the productive state, which lasts exactly k slots, and the nonproductive state, which lasts for a random number of slots. Clearly, the channel efficiency is the ratio of $k/(k + x)$, where x is the expected number of consecutive unproductive slots.

a. For fixed N and p, determine the efficiency of this protocol.

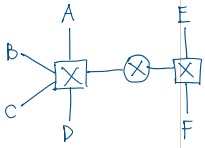
b. For fixed N, determine the p that maximizes the efficiency.

c. Using the p (which is a function of N) found in (b), determine the efficiency as N approaches infinity.

d. Show that this efficiency approaches 1 as the frame length becomes large.

P21. Consider Figure 6.33 in problem P14. Provide MAC addresses and IP addresses for the interfaces at Host A, both routers, and Host F. Suppose Host A sends a datagram to Host F. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted *(i)* from A to the left router, *(ii)* from the left router to the right router, *(iii)* from the right router to F. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time.

P22. Suppose now that the leftmost router in Figure 6.33 is replaced by a switch. Hosts A, B, C, and D and the right router are all star-connected into this switch. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted *(i)* from A to the switch, *(ii)* from the switch to the right router, *(iii)* from the right router to F. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time.



P23. Consider Figure 5.15. Suppose that all links are 120 Mbps. What is the maximum total aggregate throughput that can be achieved among 12 hosts (4 in each department) and 2 servers in this network? You can assume that any host or server can send to any other host or server. Why?

P24. Suppose the three departmental switches in Figure 5.15 are replaced by hubs. All links are 120 Mbps. Now answer the questions posed in Problem P23.

P21.

i) Src MAC: AA-AA-AA-AA-AA-AA
Dst MAC: 13-13-13-13-13-13

ii) Src MAC: 23-23-23-23-23-23
Dst MAC: 24-24-24-24-24-24

iii) Src MAC: 33-33-33-33-33-33
Dst MAC: 0F-0F-0F-0F-0F-0F

Src IP: 192.168.1.1
Dst IP: 192.168.3.2

P22.

i) Src MAC: AA-AA-AA-AA-AA-AA
Dst MAC: 24-24-24-24-24-24

ii) Src: MAC: AA-AA-AA-AA-AA-AA
Dst MAC: 24-24-24-24-24-24

iii) Src MAC: 33-33-33-33-33-33
Dst MAC: 0F-0F-0F-0F-0F-0F

P25. Suppose that *all* the switches in Figure 5.15 are replaced by hubs. All links are 120 Mbps. Now answer the questions posed in Problem P23.

P26. Let's consider the operation of a learning switch in the context of a network in which 6 nodes labeled A through F are star connected into an Ethernet switch. Suppose that *(i)* B sends a frame to E, *(ii)* E replies with a frame to B, *(iii)* A sends a frame to B, *(iv)* B replies with a frame to A. The switch table is initially empty. Show the state of the switch table before and after each of these events. For each of these events, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.

P27. In this problem, we explore the use of small packets for Voice-over-IP applications. One of the drawbacks of a small packet size is that a large fraction of link bandwidth is consumed by overhead bytes. To this end, suppose that the packet consists of P bytes and 5 bytes of header.

a. Consider sending a digitally encoded voice source directly. Suppose the source is encoded at a constant rate of 128 kbps. Assume each packet is entirely filled before the source sends the packet into the network. The time required to fill a packet is the **packetization delay**. In terms of L, determine the packetization delay in milliseconds.

b. Packetization delays greater than 20 msec can cause a noticeable and unpleasant echo. Determine the packetization delay for $L = 1{,}500$ bytes (roughly corresponding to a maximum-sized Ethernet packet) and for $L = 50$ (corresponding to an ATM packet).

c. Calculate the store-and-forward delay at a single switch for a link rate of $R = 622$ Mbps for $L = 1{,}500$ bytes, and for $L = 50$ bytes.

d. Comment on the advantages of using a small packet size.

P28. Consider the single switch VLAN in Figure 6.25, and assume an external router is connected to switch port 1. Assign IP addresses to the EE and CS hosts and router interface. Trace the steps taken at both the network layer and the link layer to transfer an IP datagram from an EE host to a CS host (*Hint:* Reread the discussion of Figure 6.19 in the text).

P29. Consider the MPLS network shown in Figure 6.29, and suppose that routers R5 and R6 are now MPLS enabled. Suppose that we want to perform traffic engineering so that packets from R6 destined for A are switched to A via R6-R4-R3-R1, and packets from R5 destined for A are switched via R5-R4-R2-R1. Show the MPLS tables in R5 and R6, as well as the modified table in R4, that would make this possible.

P30. Consider again the same scenario as in the previous problem, but suppose that packets from R6 destined for D are switched via R6-R4-R3, while packets from R5 destined to D are switched via R4-R2-R1-R3. Show the MPLS tables in all routers that would make this possible.

P31. In this problem, you will put together much of what you have learned about Internet protocols. Suppose you walk into a room, connect to Ethernet, and

P26.

i)

B's MAC	B's interface

ii)

B's MAC	B's interface
E's MAC	E's interface

iii)

B's MAC	B's interface
E's MAC	E's interface
A's MAC	E's interface

iv) Same as above

Action	Switch Table State	Links packet is forwarded to
(i) B sends a frame to E	Adds (B's mapping, B's interface)	A, C, D, E, F
(ii) E replies with a frame to B	Adds (E's mapping, E's interface)	B
(iii) A sends a frame to B	Adds (A's mapping, A's interface)	B
(iv) B sends a frame to A	No change	A

want to download a Web page. What are all the protocol steps that take place, starting from powering on your PC to getting the Web page? Assume there is nothing in our DNS or browser caches when you power on your PC. (*Hint:* The steps include the use of Ethernet, DHCP, ARP, DNS, TCP, and HTTP protocols.) Explicitly indicate in your steps how you obtain the IP and MAC addresses of a gateway router.

P32. Consider the data center network with hierarchical topology in Figure 6.30. Suppose now there are 80 pairs of flows, with ten flows between the first and ninth rack, ten flows between the second and tenth rack, and so on. Further suppose that all links in the network are 10 Gbps, except for the links between hosts and TOR switches, which are 1 Gbps.

a. Each flow has the same data rate; determine the maximum rate of a flow.

b. For the same traffic pattern, determine the maximum rate of a flow for the highly interconnected topology in Figure 6.31.

c. Now suppose there is a similar traffic pattern, but involving 20 hosts on each rack and 160 pairs of flows. Determine the maximum flow rates for the two topologies.

P33. Consider the hierarchical network in Figure 6.30 and suppose that the data center needs to support e-mail and video distribution among other applications. Suppose four racks of servers are reserved for e-mail and four racks are reserved for video. For each of the applications, all four racks must lie below a single tier-2 switch since the tier-2 to tier-1 links do not have sufficient bandwidth to support the intra-application traffic. For the e-mail application, suppose that for 99.9 percent of the time only three racks are used, and that the video application has identical usage patterns.

a. For what fraction of time does the e-mail application need to use a fourth rack? How about for the video application?

b. Assuming e-mail usage and video usage are independent, for what fraction of time do (equivalently, what is the probability that) both applications need their fourth rack?

c. Suppose that it is acceptable for an application to have a shortage of servers for 0.001 percent of time or less (causing rare periods of performance degradation for users). Discuss how the topology in Figure 6.31 can be used so that only seven racks are collectively assigned to the two applications (assuming that the topology can support all the traffic).

Wireshark Labs: 802.11 Ethernet

At the Companion Website for this textbook, http://www.pearsonglobaleditions.com, you'll find a Wireshark lab that examines the operation of the IEEE 802.3 protocol and the Wireshark frame format. A second Wireshark lab examines packet traces taken in a home network scenario.

Chapter 6 Tricky Concepts

Tuesday, August 8, 2023

1. Which organization manages MAC addresses?
   IEEE
2. What does soft state mean in the context of ARP tables?
   It refers to information that can expire unless refreshed.
3. What are some differences between a bus and star (for Ethernet)?
   In a bus, all nodes are in the same collision domain. The nodes can collide with each other.
   In a star, there is an active switch in the center. Each node runs a separate Ethernet protocol and does not collide with each other. This is used today.
4. Consider an acronym 10BASE-T. What does each part mean?
   10 stands for 10 Mbps.
   BASE refers to the baseband Ethernet
   T refers to twisted-pair copper wires
5. In a star switch, can two pairs of hosts transmit simultaneously without collisions?
   Yes
6. In a star switch, can two pairs of hosts transmit simultaneously without collisions?
   Yes