Thursday, August 31, 2023 11:42 PM

APPLICATION LAYER

2.1 Network Apps

Network apps run on hosts
& communicate over network

structure of network apps
- client-server
- peer-to-peer

Client Server

Server
- waits to be contacted
- always-on host
- permanent IP address
- server farms for scaling

Client
- initiate contact w/ server
- may be intermittently connected
- may have dynamic IP address
- do not communicate directly with each other

P2P no "Server"
- not always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
- self-scalability - new peers bring new service capacity, as well as new service demands
- peers are intermittently connected & change IP addresses
- complex management

e.g. most popular file-sharing today! (Torrent, etc.)

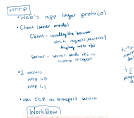
Apps need
- Data integrity
  - some apps (e.g. file transfer, web transactions) require 100% reliable data transfer
  - other apps (e.g. audio) can tolerate some loss
- Throughput
  - some apps (multimedia) require minimum amount of throughput to be effective
  - other apps ("elastic apps") make use of whatever throughput they get
- Timing
  - some apps (e.g. Internet telephony, interactive games) require low delay to be effective
- Security
  - encryption, data integrity, ...

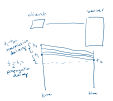
2.2 HTTP Web

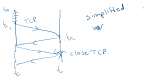
Web Page
- consists of objects
- each object can be stored on different Web servers
- object can be addressed by a URL

HTTP: Web's application layer protocol
- client/server model
  - client: browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - server: Web server sends (using HTTP protocol) objects in response to requests

uses TCP as transport service







2 Types of HTTP connections

Non-persistent HTTP (ver 1.0)
- at most 1 obj sent over TCP connection
  - then connection closed
- downloading multiple objs requires multiple connections
  - port 80 default for web

Persistent HTTP (ver 1.1)
- multiple objs can be sent over single TCP connection

RTT = time for packet to travel from client to server and go back

- 1 RTT to establish TCP connection
- 1 RTT for HTTP request and first few bytes of HTTP response to return
- file transmission time
- Non-persistent HTTP response time = 2 x RTT + file transmission time

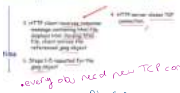
Non-persistent HTTP requires 2 RTT per obj
+ OS overhead for each TCP connection

browser after opening parallel TCP connections to fetch referenced objs

Persistent HTTP
- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over same TCP connection
- client can send requests as soon as it encounters a referenced obj (persistent w/ pipelining)

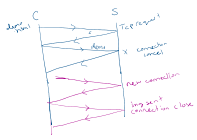
Non-persistent HTTP Example

Non-persistent HTTP Example

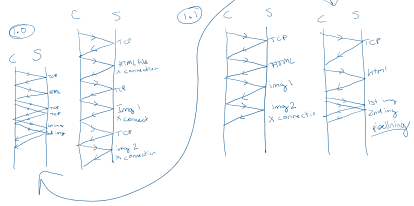


- every obj need new TCP connection

Timeline Diagram



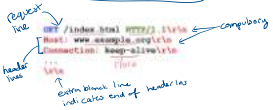
Timeline (HTML file + 2 imgs)



2.3 HTTP Request Message

2 types of msgs: request, response

Request Msg



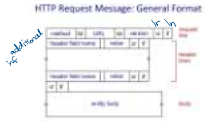
HTTP Request Method Types

HTTP/1.0:
- GET
- POST
  - web page often includes form input
  - input is uploaded to server in entity body
- HEAD
  - asks server to leave requested object out of response

HTTP/1.1:
- GET, POST, HEAD
- PUT
  - uploads file in entity body to path specified in URL field
- DELETE
  - deletes file specified in the URL field

optional

HTTP Request Message: General Format



Response Msg

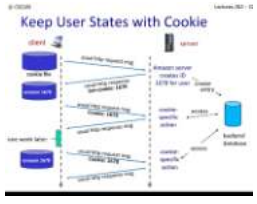
Example HTTP Response Message

Cookies

HTTP designed to be stateless
- server maintains no inf. about past client requests

- But for some sit. need to maintain states over multiple transactions
  - shopping cart

- Cookie - http msg carry "state"
  1. cookie header field of HTTP request/response msg
  2. cookie file kept on user's host managed by user's browser
  3. back-end database at Web site

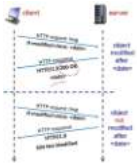


Conditional GET

- We wouldn't want to send obj if the client's cache has the up-to-date cached version

Cache - specific date of cached copy in HTTP request
  if-modified-since: <date>

server - response contains no obj if cached copy is up to date
  HTTP/1.0 304 not modified

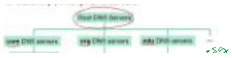


*saves bandwidth

2.4 DNS - Domain Name Sys

- 2 ways to identify host
  - Hostname (string) www.ex.org
  - IP [illegible] device connecting to internet

- DNS stores RR in distributed databases in many name servers



Local DNS server
- doesn't strictly belong to hierarchy
- gives non authoritative answer
- faster

- 2 ways to identify host
  - Hostname (string) www.ex.org
  - IP address (every device connecting to internet has 1 (93.184.216.34))
    32 bit
- DNS translates between hostname & IP address
  - client must carry out a DNS query to determine IP address corresponding to the server name (hostname) prior to connection

  mapping need not be one to one
- mapping between host names & IP addresses are stored as resource records RR

  format = (name, value, type, ttl)

Types

* type A address
  - name is hostname
  - value is IP address
* type CNAME
  - name is alias name for canonical name
  - value is canonical name
* type NS (name server)
  - name is domain

    ex nus.edu.sg
  - value is hostname of authoritative name server for this domain
* type MX mail exchange email
  - value is name of mail server associated w. name

dig a www. -- . org +short
- get IP address

dig cname
- get canonical name

www
blog
luminus
nus.edu.sg
- domain

all servers

- DNS stores RR in [illegible] in many name servers

Ex. if a client wants IP address for facebook.com,
1. client queries root server to find .com DNS server
2. client queries .com DNS server to get facebook.com DNS server
3. client queries facebook.com DNS server to get IP address for

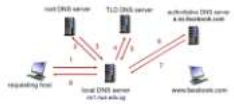
- Root servers return list of authoritive name servers for the appropriate top-lvl domain (TLD)

  13 root name servers worldwide
- TLD server
  com, edu ... UK, sg
- Authoritative servers
  - organization's own DNS server, providing authoritative hostname to IP mappings for organization's named hosts
  - can be maintained by org or service provider

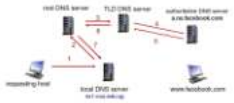
- doesn't strictly belong to hierarchy
- gives non authoritative answer
- faster

DNS caching
- once a name server learns mapping it caches mapping
  - cached entries may be out-of-date
  - may expire after some time (ttl)
  - if name host changes IP adress, may not be known internet wide until all TTLs expire
  - runs over UDP

DNS Name Resolution



- This is known as iterative query



- This is known as recursive query

2.7 Socket Programming

Processes

- program running within a host

  within same host, 2 processes communicate using inter-process communication

  processes in diff hosts communicate by exchanging msg (accord to protocol)
- IP address identifys host

  32 bit integer
- A process is identified by IP address and port number

  16 bit integer

  ↳ port number: 80 for HTTP server
  25 for SMTP server
- IANA coordinates assignment of port #

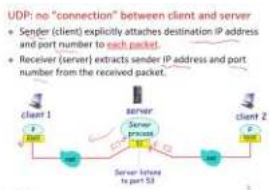
Sockets

- software interface between app processes & transport layer protocols

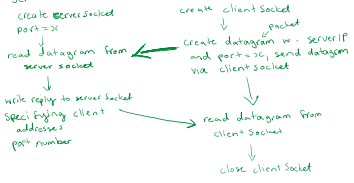
  process sends/receives msg to/from its socket

- apps treat internet as black box send & recieve msg through sockets
- 2 types of sockets
  - TCP - reliable, byte stream-oriented
  - UDP - unreliable datagram socket

UDP

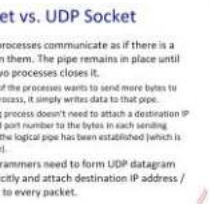


Workflow



TCP

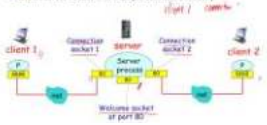
Workflow



packets explicitly and attach destination IP address / port number to every packet.

Socket Programming with *TCP*

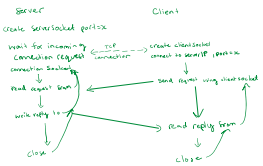
- When client creates socket, client TCP establishes a connection to server TCP.
- When contacted by client, server TCP creates a new socket for server process to communicate with that client.
  - allows server to talk with multiple clients individually.



listen()

accept()

Workflow



Transport layer services

Monday, September 4, 2023 2:06 PM

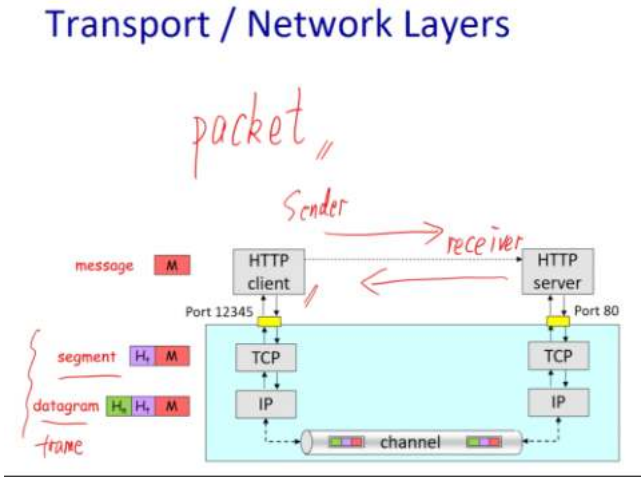
Transport layer protocols run in hosts

- Connectionless Transport - UDP
  - User Datagram protocol
  - Add very little service on top of IP
    - Multiplexing at sender
  - Checksum calculated by UDP/TCP

Reliable data transfer

- We need a reliable transport service that guarantee packet delivery and correctness. And deliver packets in same order sent
- HTTP run over TCP
- DNS run over UDP

- Sender side: breaks app message into segments, and passes to network layer (IP layer)
- Receiver side: reassembles segments into message, passes it to app layer
- Packet switches (routers in between) - only check destination IP address to decide routing
  - Routers only know IP
  - They don't run TCP/UDP

- Packet
  - At transport layer, a packet is called a segment
  - At network layer, its called datagram
  - 

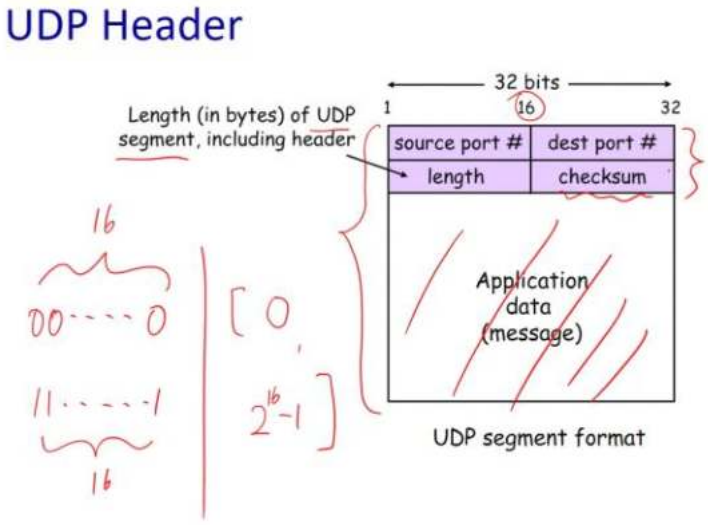
UDP

- Adds very little service on top of IP
- Multiplexing at sender: UDP gathers data from processes, forms packets and passes to IP
- De-multiplexing at receiver: UDP receives puackts from lower layer and dispatches them to right processes according to port number
- Checksum -

- UDP is unreliable
- Never tries to retransmit

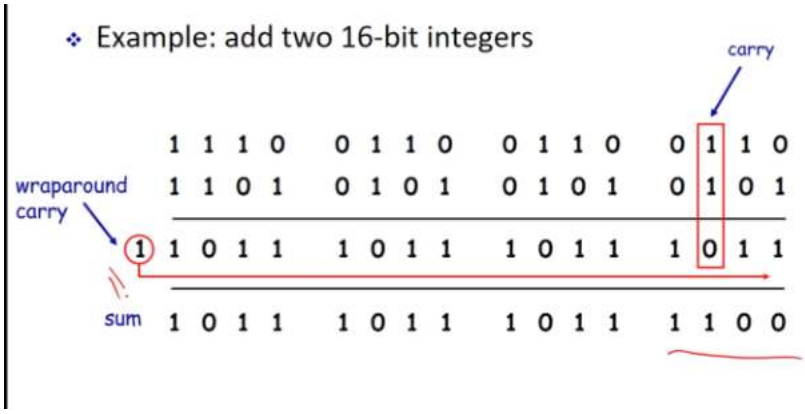
- When UDP receiver receives a UDP segmentL
  - Check destination port number in segment
  - Directs UDP segment to the socket with that port #

- IP datagrams (from diff sources) with same destination port number will be directed to **same** UDP socket at destination
- 1-to-many

- UDP Header
  - 

- UDP Checksum
  - To detect errors
  - Sender: computes checksum value and puts it into UDP checksum field
  - Receiver: computes checksum and checks if computed value equals checksum field value
    - If not, error detected
    - If equal, no error detected
      - But doesn't mean really error free
  - How to compute
    - Treat UDP segment as sequence of 16 bit integer
    - Apply binary addition on every 16 bit integer
      3. Carry (if any) from the most significant bit will be added to the result.
      4. Compute 1's complement to get UDP checksum.
    -

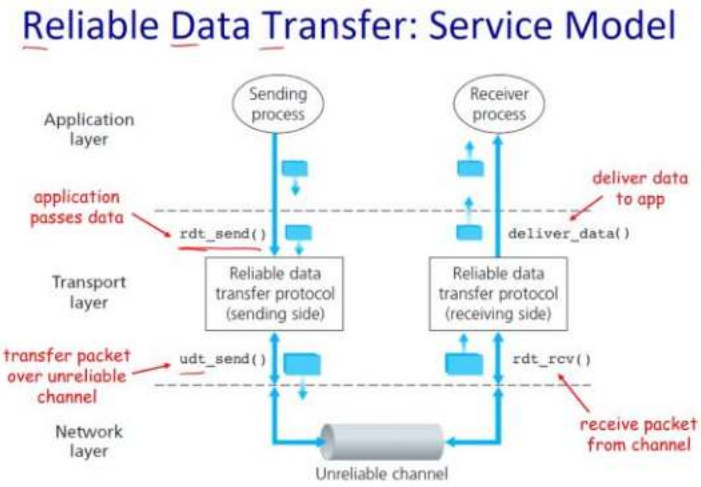
x	y	x ⊕ y	carry
0	0	0	-
0	1	1	-
1	0	1	-
1	1	0	1

    - 

    - In udp checksum, many 16 bit integers to add
    - Once you get final sum, do 1s complement

- Flip every bit

Principles of Reliable Data Transfer

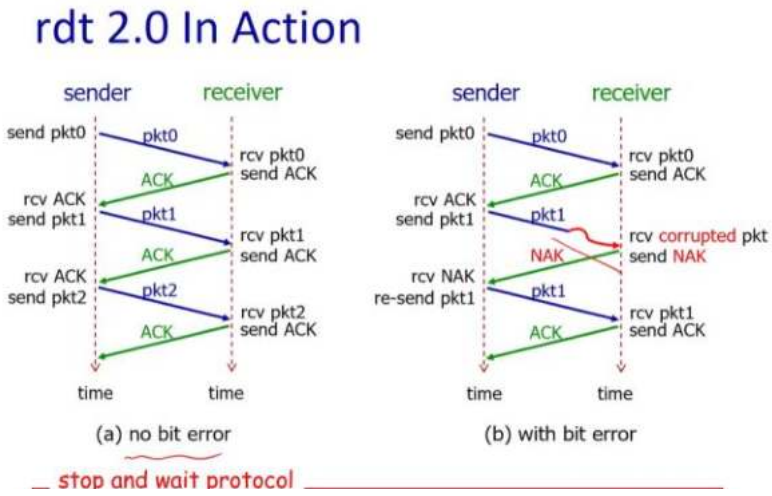
- Transport layer resides on end hosts and provides process-to-process communication
- Network layer provides host-to-host, best-effort and unreliable communication
- End-to-end transport service should
  - Guarantee packets delivery and correctness
- 

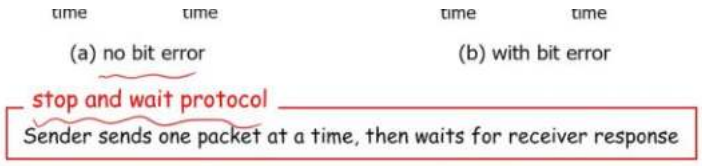
Finite State Machine (FSM)

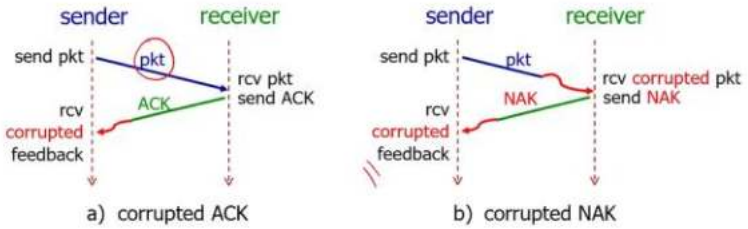
Rdt 1.0 - Perfectly reliable channel

- Assume underlying channel is perfectly reliable
- Separate FSMs for sender, receiver
  - Sender sends data into perfect channel
  - Receiver reads data from perfect channel
- No error detection needed

Rdt 2.0 -channel with bit errors

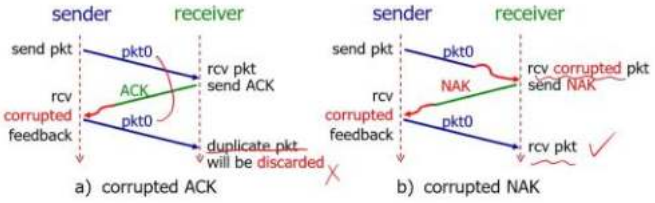
- Underlying channel may flip bits in packet
  - Otherwise perfect
- Receiver may use checksum to detect bit errors
- If bit error detected,
  - Acknowledgements (ACKs) receiver explicitly tells sender that packet received is OK
  - Negative Acknowledgements (NAKs) receiver explicitly tells sender that packet has errors
    - Sender then retransmits packet on receipt of NAK
- 

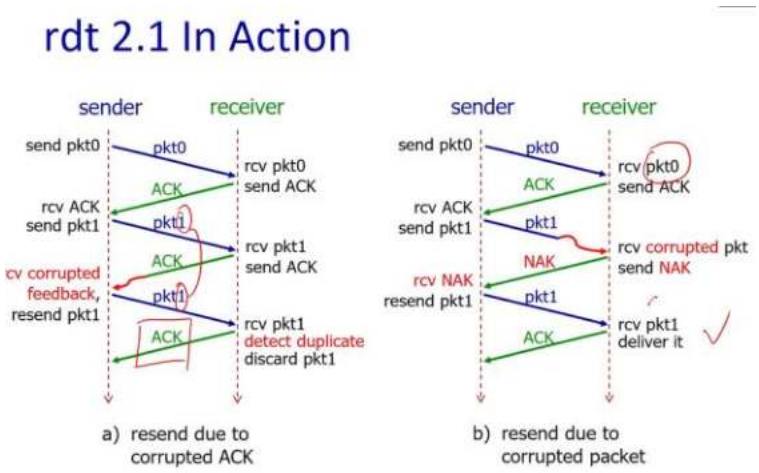


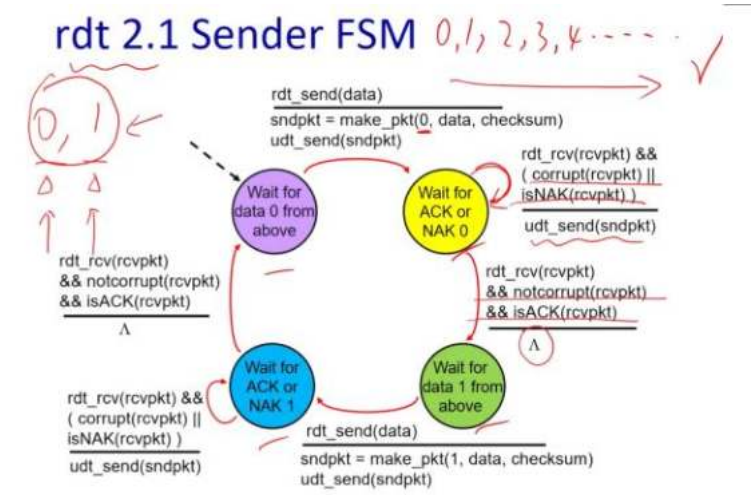
- Has a fatal flaw
  - If ACK/NAK is corrupted,
  - 

  - Is sender receives corrupted ACK, they don't know which one is the correct scenario
  - So sender just retransmits the packet when they receive garbled ACK or NAK
    - But this can cause retransmission nad a duplicate packet which will not be detected in scenario A

Rdt 2.1 - rdt 2 + packet sequence number

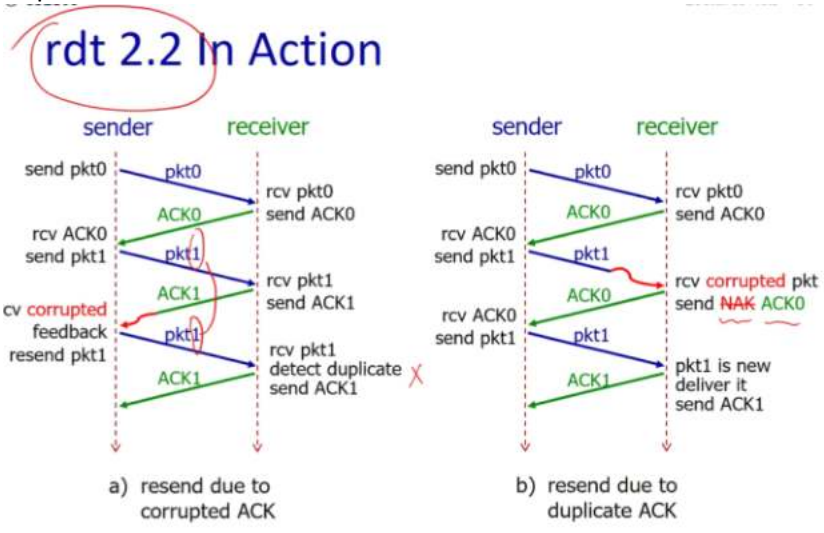
- In order to handle duplicate packets from previous,
  - Sender retransmits current packet if ACK/NAK is garbled
  - Sender adds sequence number to each packet
  - Receiver discards (doesn't deliver up) duplicate packet
  - rdt 2.1: rdt 2.0 + Packet Seq. #
    - To handle duplicates:
      - Sender retransmits current packet if ACK/NAK is garbled.
      - Sender adds *sequence number* to each packet.
      - Receiver discards (doesn't deliver up) duplicate packet.
    - This gives rise to protocol rdt 2.1.
  - 

- 

- 

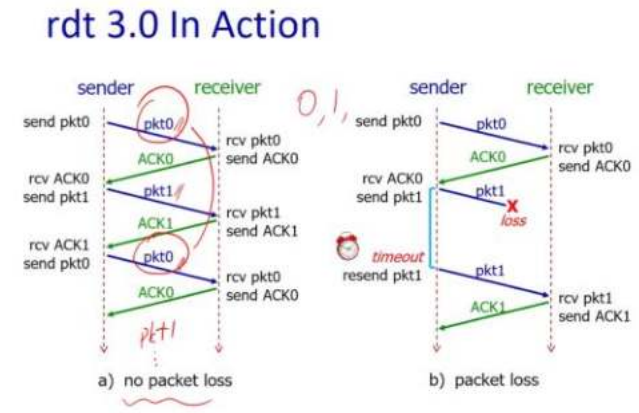
  - 2 sequence number 0,1 are sufficient

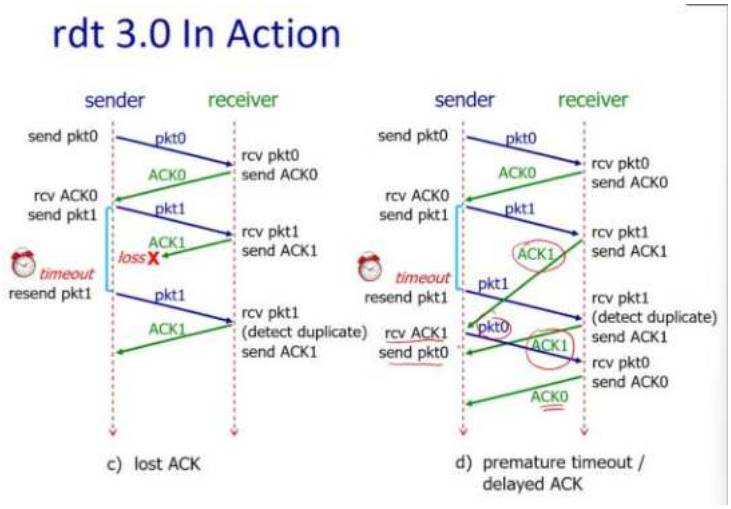
Rdt 2.2 - NAK-free protocol

- Use ACKs only
- Instead of sending NAK, receiver sends ACK for the last packet received OK
  - Must explicitly include sequence number of packet being ACKed
- Duplicate ACKs at sender results in same action as NAK : retransmit current packet
- 

Rdt 3.0 - channel with errors and loss

- Assume underlying channel
  - May flip bits, lose packets, incur arbitrarily long packet delay
  - But will not reorder packets
- How to detect packet loss
- Sender waits "resonable" amount of time for ACK
  - If not ACK received until timeout, assume lost
- Its possible that the packet was just delayed not lost
  - Then the timeout triggers retransmission
  - Retransmission generates duplicates, but receiver can use sequence number to detect it





RDT Summary

rdt Version	Scenario	Features Used
1.0	no error	nothing
2.0	data Bit Error	checksum, ACK/NAK
2.1	data Bit Error ACK/NAK Bit Error	checksum, ACK/NAK, sequence Number
2.2	Same as 2.1	NAK free
3.0	data Bit Error ACK/NAK Bit Error packet Loss	checksum, ACK/NAK, sequence Number, timeout/re-transmission

Tutorial_2_qns

Thursday, September 7, 2023 8:18 AM

Tutorial_2_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 2** Question paper

To students:

Due to time constraint, not all the questions will be discussed in class. Your tutor has the discretion to choose the questions to discuss (or you may request your tutor to discuss certain questions). Please go through the rest questions after class.

1. Consider the following HTTP request message sent by a browser.

   GET /index.html HTTP/1.1

   Host: www.example.org

   Connection: keep-alive

   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36

   Accept-Encoding: gzip, deflate

   …

   a) What is the URL of the document requested by this browser? www.example.org/index.html
   b) What version of HTTP is this browser running? 1.1
   c) Does the browser request a non-persistent or a persistent connection?
   d) What is the IP address of the host on which the browser is running? ~~71.0.3578.98~~ not shown

2. The text below shows the header of the response message sent from the server in reply to the HTTP GET message in Q1 above. Answer the following questions.

   HTTP/1.1 200 OK

   Content-Encoding: gzip

   Content-Type: text/html; charset=UTF-8

   Date: Wed, 23 Jan 2019 13:50:31 GMT

Page 1 of 4

Http wouldn't know ip address
User agent - is about browser

Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT

Connection: Keep-Alive

Content-Length: 606

...

a) Was the server able to successfully find the document or not? yes

b) What time did the server send the HTTP response message? 23 Jan 13:50:31

c) How many bytes are there in the document being returned? 606

d) Did the server agree to a persistent connection? yes

3. True or false?

a) A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages. F, 1 obj per request

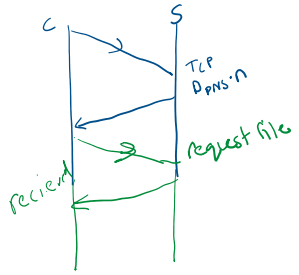
b) Two distinct Web pages (for example, www.mit.edu/research.html and www.mit.edu/students.html) can be sent over the same persistent connection. ~~F~~ T, on same server

Same sub domain- same server

c) The **Date:** header in the HTTP response message indicates when the object in the response was last modified. F

d) HTTP response messages never have an empty message body. False,

4. **[Modified from KR, Chapter 2, P7]** Suppose within your Web browser, you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address.
Suppose that ***n*** DNS servers are visited before your host receives the IP address from DNS; visiting them incurs an RTT of D_{DNS} per DNS server.
Further suppose that the Web page associated with the link contains ***m*** very small objects (in addition to the HTML page). Suppose the HTTP running is non-persistent and non-parallel. Let D_{Web} denote the RTT between the local host and the server of each object.
Assuming zero transmission time of each object, how much time elapses from when the client clicks on the link until the client receives all the objects?



$n D_{DNS} + 2 \times D_{Web} + m \times 2 \times D_{Web}$

map hostname to IP address

obj sent 1 by 1

5. **[Modified from KR, Chapter 2, P8]** Referring to the previous question, suppose that three DNS servers are visited. Further, the HTML file references five very small objects on the same server. Neglecting transmission delay, how much time elapses with:
   a) Non-persistent HTTP with no parallel TCP connections? $3 \cdot D_{DNS} + 2 D_{web} + 10 D_{web}$
   b) Non-persistent HTTP with the browser configured for five parallel connections? $3 \cdot D_{DNS} + 2 D_{web} + 2 D_{web}$
   c) Persistent HTTP with pipelining?

confused conceptually w. difference

3Ddns +5Dweb
Nonpersistnet http with pipelining ?

6. Do you know what is DNS cache poisoning? Search online for a real example. Gov. intentionally do this to restrict some websites rogue DNS records introduced into DNS resolver's cache causing name server to return incorrect IP address

7. Wireshark Introduction
   Wireshark is a tool for observing the messages exchanged between executing protocol entities. It observes messages being sent and received by applications and protocols running on your computer.
   Download and install the Wireshark software:
   - Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer.

   Taking Wireshark for a test run:
   1. Start up your web browser.
   2. Start up the Wireshark software.
   3. To begin packet capture, select the Capture pull down menu and select Interfaces.
   4. Click on Start for the interface on which you want to begin packet capture.
   5. While Wireshark is running, enter the URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html and have that page displayed in your browser.
   6. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities!
   7. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply. This will cause only HTTP message to be displayed in the packet-listing window.

   Congratulations! You've now completed the Wireshark introduction.

8. Wireshark: HTTP GET/response interaction

   Let's begin our exploration of HTTP by downloading a very simple HTML file, and contains no embedded objects. Do the following:

   1. Start up your web browser.
   2. Start up the Wireshark packet sniffer. Enter "http" in the display-filter-specification window and begin Wireshark packet capture.
   3. Enter the following to your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html.
   4. Stop Wireshark packet capture.

   Now answer the following questions:

   1. What is the status code returned from the server to your browser?
   2. When was the HTML file that you are retrieving last modified at the server?

304 not modified

Tutorial_3_qns

Thursday, September 14, 2023 11:48 PM

Tutorial_3_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 3** Question paper

To students:

Please be reminded that submission deadline of **assignment 1** is **Monday (25 Sep 2023) 2359**.

1. Launch your browser and open its network diagnostic tool (e.g. press F12 if you use Chrome on Windows, or Cmd + Opt + I for Mac). Then click the "Network" tab to observe network communication.
   Copy-and-paste the following URL in the address bar of your browser:
   http://tiny.cc/atupaz

   Enter your choice and press the "Submit" button.

   a) Look at the entry named "formResponse". What is the HTTP request method issued? POST

   b) Briefly explain when HTTP POST and GET methods are used.
   send data to server receive data

2. **[KR, Chapter 2, P21]** Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e., not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain. yes, can check if query has been cached

3. **[Modified from KR, Chapter 2, P31]** You are given 4 programs: **TCPEchoServer.py**, **TCPEchoClient.py**, **UDPEchoServer.py** and **UDPEchoClient.py**.

   a) Suppose you run **TCPEchoClient** before you run **TCPEchoServer**. What happens? Why?

   b) Suppose you run **UDPEchoClient** before you run **UDPEchoServer**. What happens? Why? waits for response indefinately

4. **[KR, Chapter 3, R7]** Suppose a process in Host C has a UDP socket with port number 6,789. Suppose both Host A and Host B each sends a UDP segment to Host C with destination port number 6,789. Will both of these segments be directed to the same

Page 1 of 2

Port Socket → port IP tuple

send msg

BUC – IP address
UNIT no port

yes

socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

yes but post would be diff

5. **[Modified from KR, Chapter 3, P4]**

   a) Suppose you have the following 2 bytes: **01011100** and **01100101**. What is the 1s complement of the sum of these 2 bytes?

   sum: 11000001

   1s 00111110

   b) Suppose you have the following 2 bytes: **11011010** and **01100101**. What is the 1s complement of the sum of these 2 bytes?

   sum 00111111 → 1s 11000000

   (Note: UDP and TCP use 16-bit words in computing their checksums. For simplicity you are asked to consider 8-bit checksums in this problem).

- each router adds everything including checksum resulting in 11... if nothing is wrong

6. **[Modified from KR, Chapter 3, P5]** Suppose that UDP receiver computes the checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? You may use Q5 as an example to explain.

7. **[KR, Chapter 3, R9]** In our `rdt` protocols, why did we need to introduce sequence numbers?

8. Do you have any question on Assignment 1 to clear?

Monday, September 18, 2023 9:24 AM

- Network layer is unreliable as it transfers data. Transport layer dispatches data
- Timeout is to deal with packet loss
- Rdt 3.0 works but has horrible performance

Performance of rdt 3.0

- rdt 3.0 works, but performance stinks.
- Example: packet size = 8000 bits, link rate = 1 Gbps:

$$d_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 0.008 \text{ msec}$$

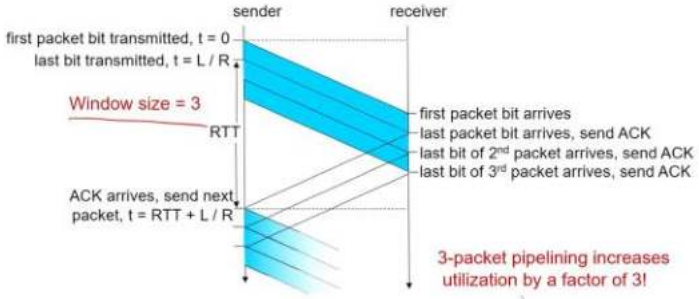
- If RTT = 30 msec, sender sends 8000 bits every 30.008 msec.

$$\text{throughput} = \frac{L}{RTT + d_{trans}} = \frac{8000}{30.008} = 267 \text{ kbps}$$

- U_{sender}: *utilization* – fraction of time sender is busy sending

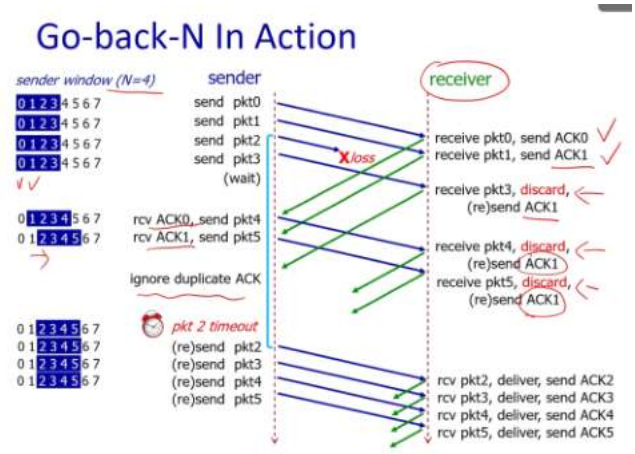
$$U_{sender} = \frac{d_{trans}}{RTT + d_{trans}} = \frac{0.008}{30 + 0.008} = 0.00027$$

Pipelining: Increased Utilization



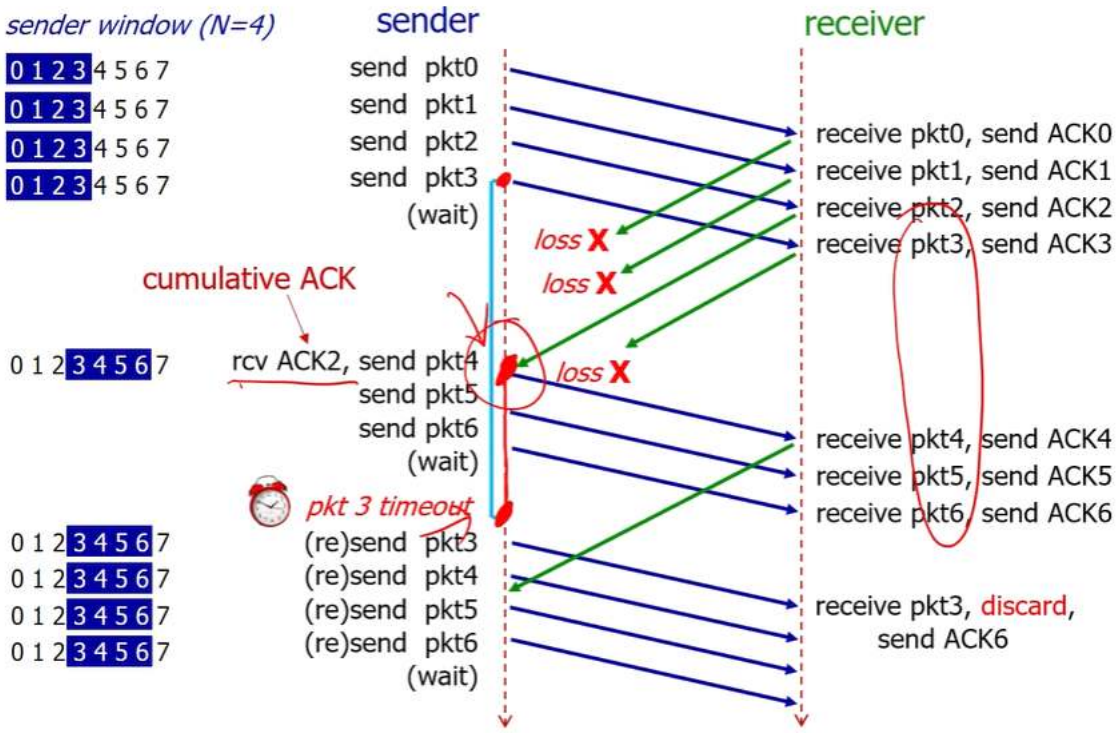
$$U_{sender} = \frac{3 * L / R}{RTT + L/R} = \frac{0.024}{30 + 0.008} = 0.00081$$

- Pipelining allows sender to send multiple, inflight yet to be acknowledged packets
  - But range of sequence numbers must increase
  - Buffering at sender/receiver
- Generic forms of pipelined protocols
  - Go-Back-N (GBN)
  - Selective repeat (SR)
  - Assumptions are packets won't be reordered
- Go-back-N
  - Stubborn receiver
  - Any packets after lost packet are discarded

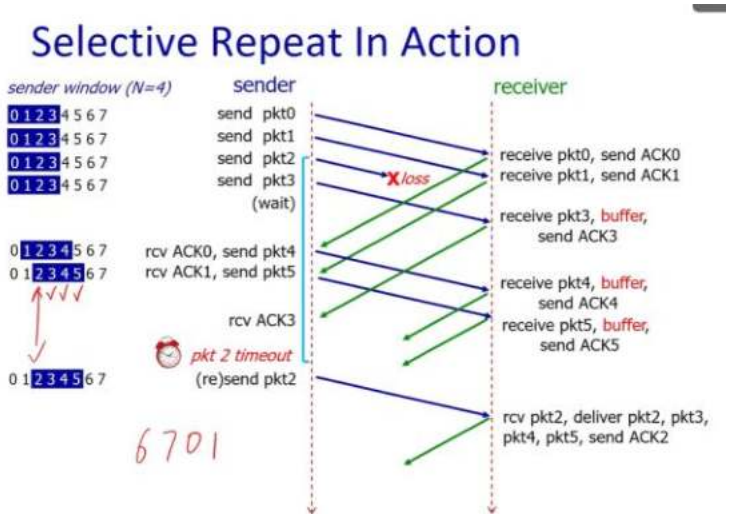
- 

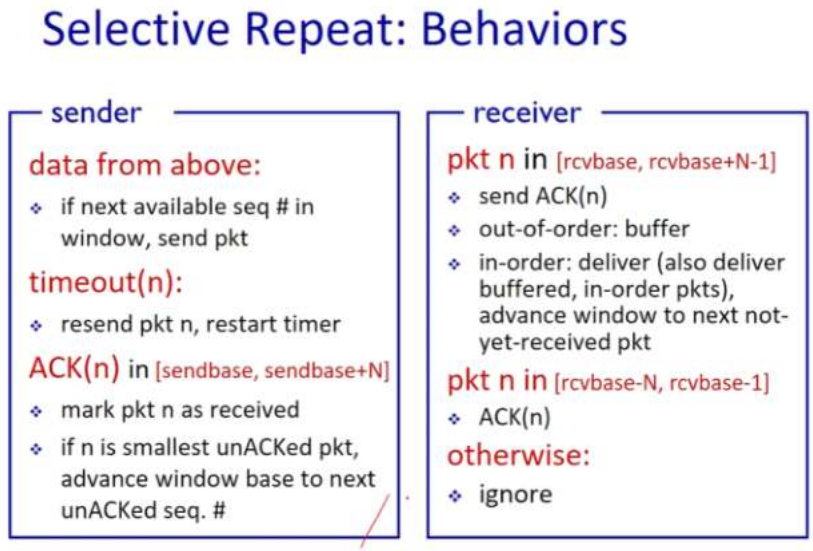
  - Sender can have up to N unACKed packets in pipeline
  - N is window size
  - Insert k-bits sequence number in packet header
  - Use a sliding window to keep track of unACKed packets
  - Keep a timer for the oldest unACKed packet
  - Timeout(n) retransmit packet n and all subsequent packets in the window
  - Receiver only accepts ACK packets that arrive in order
  - Discards out of order packets and ACK the last in-order seq. #
    - Cumulative ACK - means all packets up to m are received "ACK m"

Go-back-N In Action

- 

    - If timeout value is too small, premature timeout may happen trigger unnessary transmission
    - But very large timeout will make system slow in responding to packet loss/corruption
- Selective repeat
  - Receiver individually acknowledges all correctly received packets
  - Buffers out-of-order packets for eventual in order delivery to upper layer
  - Sender maintains timer for each unACKed packet
    - When timer expires, retransmit only that unACKed packet

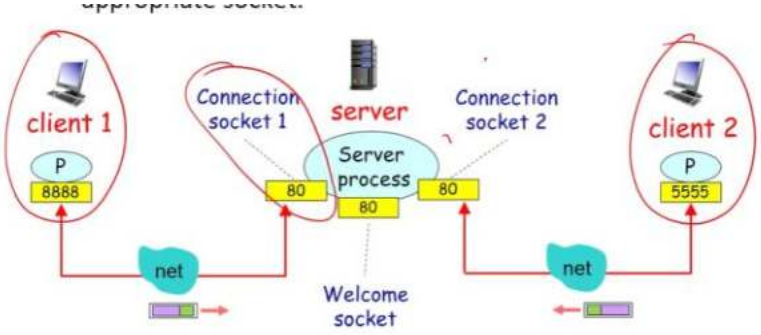
- 

- 

TCP - transport control protocol

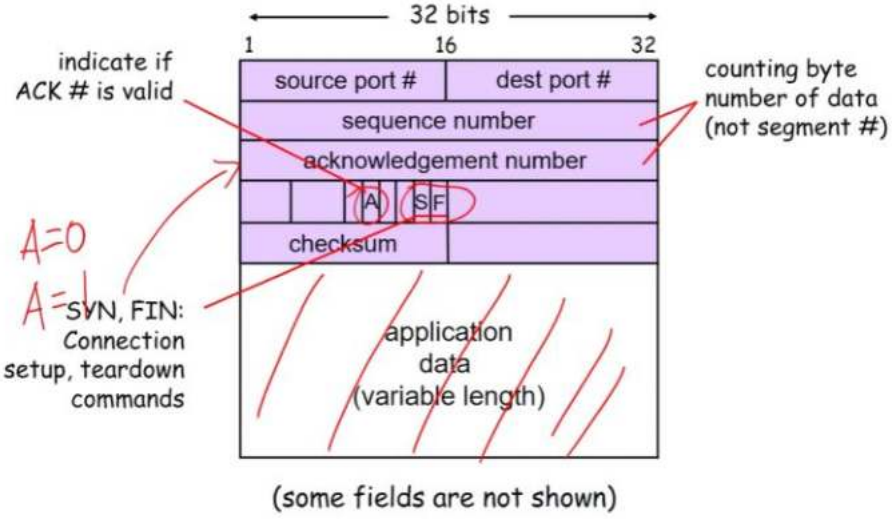
- Unlike UDP, TCP is complex and descripted in tens of RFC
- Point-to-point - meaning there is 1 sender and 1 receiver
- Connection oriented - handshaking (exchange of control messages) before sending app data
  - Establish TCP connection before actually sending anything
  - Purpose of handshake is to share sequence numbers
  - SYN packets consume 1 sequence #
  - Pure ACK consume no sequence #
- Full duplex service
  - Bi-directional data flow in same connection
    - Both parties can send data to each other
- Reliable, inorder byte stream
  - Uses sequence #s to label bytes

- Connection-oriented De-mux
  - TCP connection (socket) is identified by 4-tuple
  - (srcIPAddr, srcPort, destIPAddr, destPort)
  - Receiver uses all four values to direct a segment to the appropriate socket

- 

- TCP can send and receive buffers
  - 2 buffers created after handshaking at any side
  - Maximum segment size (MSS) around 1460 bytes
    - MSS refers the max amout amount of application data in TCP segment does not include data in TCP header
    - TCP header itself is another 20 bytes
    - So actual size is 1460 + 20
  - App passes data to TCP and TCP forms packets In view of MSS

TCP Header

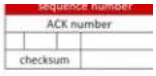
- 

- A field if 0, means there is application data. If a = 1, then it's a feedback packet

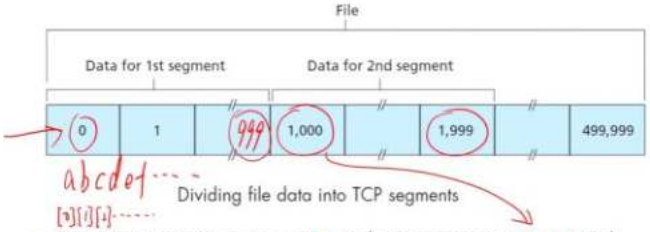
TCP Sequence Number

- "byte number" of first byte of data in a segment

TCP Sequence Number

- 

- "Byte number" of the first byte of data in a segment.
- Example: send a file of 500,000 bytes; MSS is 1,000 bytes.

- 

- Seq. # of 1st TCP segment: 0, 2nd TCP segment: 1,000, 3rd TCP segment: 2,000, 4th TCP segment: 3,000, etc.

TCP ACK Number

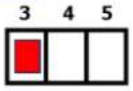
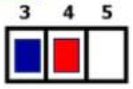
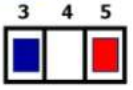
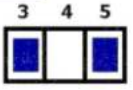
- Seq # of the next byte of data expected by receiver.

Sequence number of a segment	Amount of data carried	Corresponding ACK number
0	1,000	1,000
1,000	1,000	2,000
2,000	1,000	3,000
3,000	1,000	4,000
...	...	...

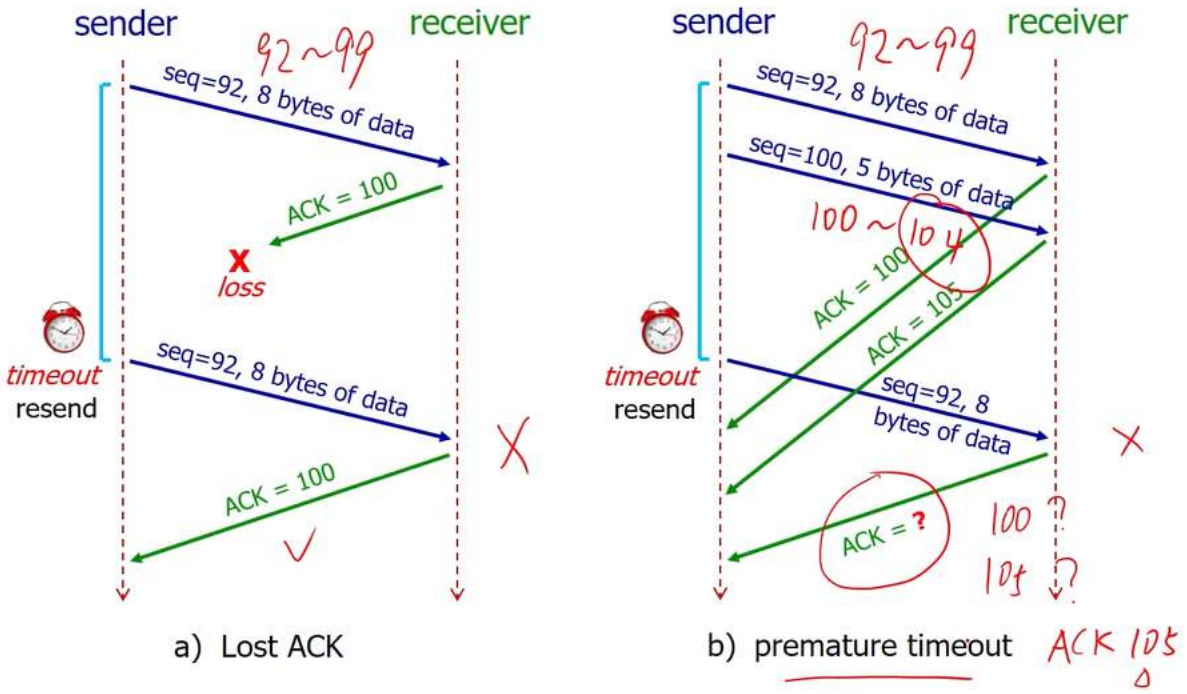
- TCP ACKs up to the first missing byte in the stream (cumulative ACK).
  - Note: TCP spec doesn't say how receiver should handle out-of-order segments - it's up to implementer.

Initial sequence number is an offset applieed to every data packet

TCP ACK Generation [RFC 2581]

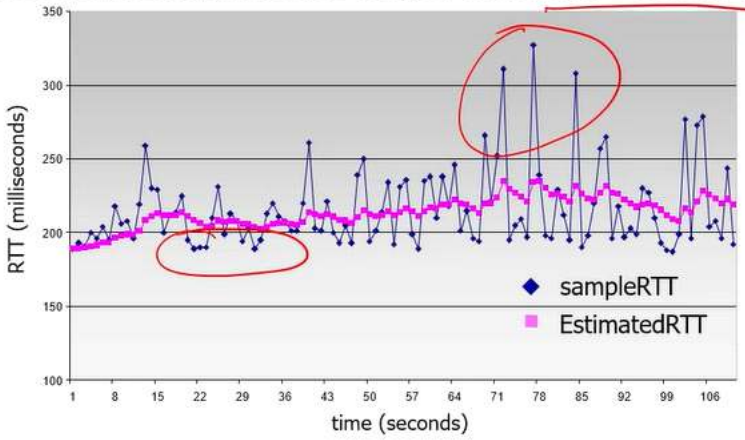
Event at TCP receiver	*TCP receiver action*	
Arrival of in-order segment with expected seq #. All data up to expected seq # already ACKed	Delayed ACK: wait up to 500ms for next segment. If no next segment, send ACK	
Arrival of in-order segment with expected seq #. One other segment has ACK pending	Immediately send single cumulative ACK, ACKing both in-order segments	
Arrival of out-of-order segment higher-than-expect seq. # (gap detected)	Immediately send *duplicate ACK*, indicating seq. # of next expected byte	
Arrival of segment that partially or completely fills gap	Immediately send ACK, provided that segment starts at lower end of gap	

TCP Timeout / Retransmission



TCP Timeout Value

- How does TCP set appropriate timeout value?
  - too short timeout: premature timeout and unnecessary retransmissions.
  - too long timeout: slow reaction to segment loss.
  - Timeout interval must be longer than RTT – but RTT varies!



TCP Timeout Value

- TCP computes (and keeps updating) timeout interval based on estimated RTT.

$$\texttt{EstimatedRTT} = (1-\alpha)*\texttt{EstimatedRTT} + \alpha*\texttt{SampleRTT}$$

(typical value of α : 0.125)

$$\texttt{DevRTT} = (1-\beta)*\texttt{DevRTT} + \beta*|\texttt{SampleRTT}-\texttt{EstimatedRTT}|$$

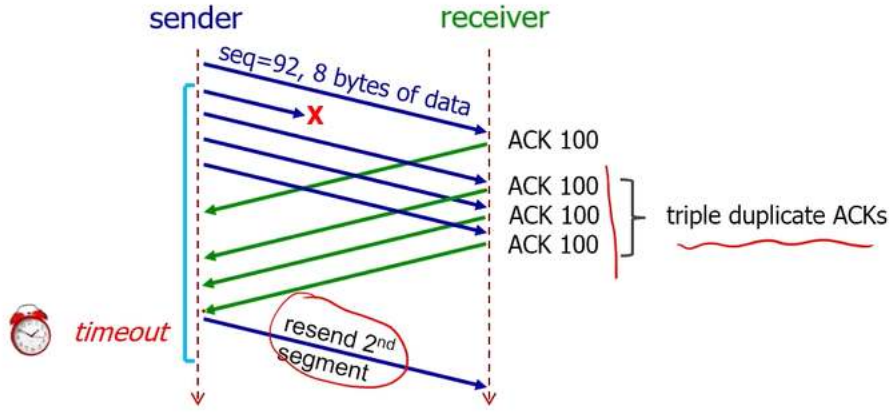
(typical value of β : 0.25)

$$\texttt{TimeoutInterval} = \texttt{EstimatedRTT} + 4*\texttt{DevRTT}$$

"safety margin"

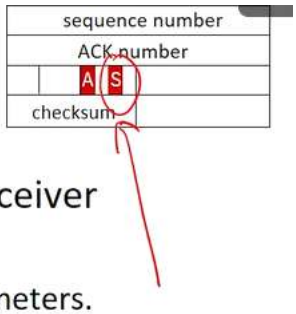
TCP Fast Retransmission [RFC 2001]

- Timeout period is often relatively long.
  - long delay before resending lost packet
- Fast retransmission:
  - **Event:** If sender receives 4 ACKs for the same segment, it supposes that segment is lost.
  - **Action:** resend segment (even before timer expires).



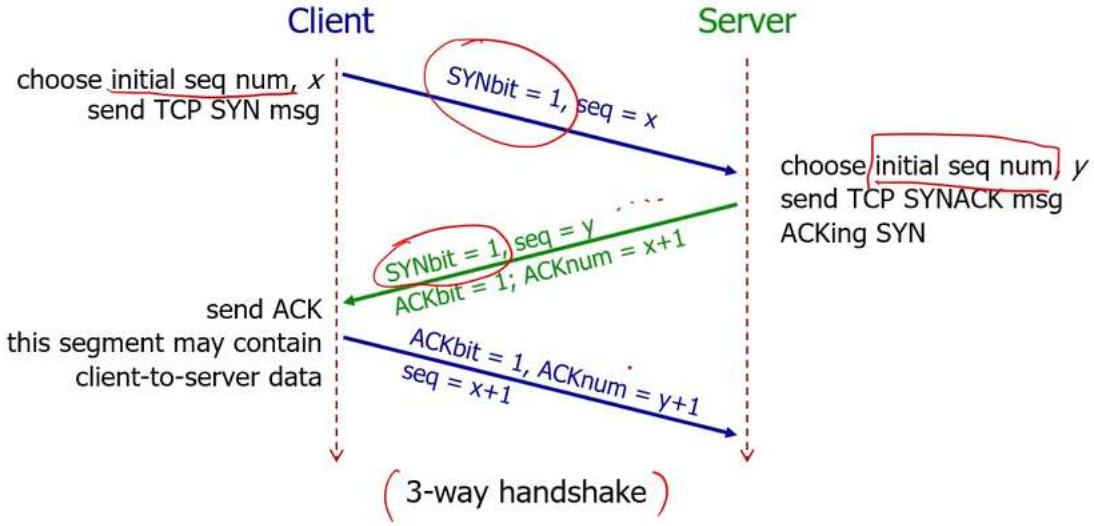
ESTABLISHING CONNECTION

- Syn packet if 1, is to establic connection



Establishing Connection

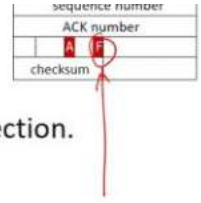
- Before exchanging app data, TCP sender and receiver "shake hands".
  - Agree on connection and exchange connection parameters.



- 3rd packet can start carrying application data

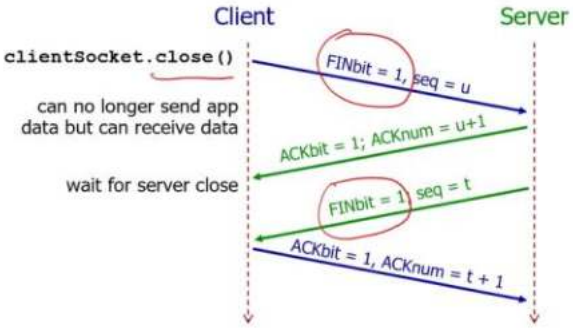
CLOSING CONNECTION

- FINbit = 1 - to close TCP
  - After sending this, client can no longer send data



Closing Connection

- Client, server each close their side of connection.
  - send TCP segment with FIN bit = 1



Timeout, duplicate ack are the 2 cases when tcp sender restransmit

Tutorial_4_qns

Friday, September 22, 2023 10:14 AM

Tutorial_4_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 4** Question paper

To students:

Please take note that our **Mid-term assessment** covers everything taught before recess week. Two practice papers will be uploaded to LumiNUS Files in recess week.

1. **[KR, Chapter 3, R6]** Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

   Note: this is exactly what you are supposed to do in Assignment 2. ☺

2. Show an example that if the communication channel between the sender and receiver can reorder messages (i.e. two messages are received in different order they are sent), then protocol **`rdt3.0`** will not work correctly.

3. **[KR, Chapter 3, P29]** It is generally a reasonable assumption, when sender and receiver are connected by a single wire, that packets cannot be reordered within the channel between the sender and receiver. However, when the "channel' connecting the two is a network, packet reordering may occur. One manifestation of packet reordering is that old copies of a packet with a sequence or acknowledgement number of *x* can appear, even though neither sender's nor receiver's window contains *x*. With packet reordering, the channel can be thought of as essentially buffering packets and spontaneously emitting these packets at any point in the future. What is the approach taken in practice to guard against such duplicate packets?

4. **[Modified from KR, Chapter 3, P37]** Host A is sending data segments to Host B using a reliable transport protocol (either GBN or SR). Assume timeout values are sufficiently large such that all data segments and their corresponding ACKs can be received (if not lost in the channel) by Host B and the Host A respectively. Suppose Host A sends 5 data segments to Host B and the 2nd data segment is lost. Further suppose retransmission is always successful. In the end, all 5 data segments have been correctly received by Host B.

Page 1 of 3

How many segments has Host A sent in total and how many ACKs has Host B sent in total if either GBN or SR protocol is used? What are their sequence numbers? Answer this question for both protocols.

A GBN - 9 B GBN - 4 ACK for 1, 4 ACK with 2,3,4,5
A SR - 6 B SR - 4 ACK with 1, 2,4,5, and 1ACK with 2

5. **[KR, Chapter 3, R15]** Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 65; the second has sequence number 92.

   a) How much data is in the first segment?

   b) Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

6. **[KR, Chapter 3, P26]** Consider transferring an enormous file of *L* bytes from Host A to Host B. Assume an MSS of 512 bytes.

   1. What is the maximum value of *L* such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field is 32 bits.
   2. For the *L* you obtain in (a), find how long it takes to transmit this file. Assume that a total of 64 bytes of transport, network, and data-link header are added to each packet before the resulting packet is sent out over a 155 Mbps link. Ignore flow control, congestion control and assume Host A can pump out all segments back to back and continuously.

7. Wireshark: TCP

   Do the following:
   1. Start up your web browser. Go the http://gaia.cs.umass.edu/wiresharklabs/alice.txt and retrieve an ASCII copy of Alice in Wonderland. Store this file somewhere on your computer.
   2. Next, go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
   3. Use the Browse button to enter the full path name on your computer containing Alice in Wonderland. Don't yet press the "Upload alice.txt file" button.
   4. Startup Wireshark and begin packet capture.
   5. Returning to your browser, press the "Upload alice.txt file" button. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
   6. Stop Wireshark packet capture.

Answer the following questions:

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

practice_paper_2

Tuesday, October 3, 2023 1:37 PM

practice_paper_2

1. Which of the following statements about HTTP is FALSE?
   A. HTTP runs on top of TCP. ✓
   B. HTTP is an application layer protocol. ✓
   C. In HTTP/1.0, the server will close the connection after every request. ✓
   D. In HTTP/1.1, the default connection type is persistent. ✓
   E. HTTP is only used to download HTML data from a Web server. ✗ post, send data update

2. UDP uses _____ to dispatch incoming packets to different processes in the same host.
   A. multiplexing
   B. de-multiplexing
   C. congestion control
   D. flow control
   E. IP address

3. Which of the following statements about DNS is FALSE?
   A. DNS provides hostname to IP address mapping.
   B. A hostname may be mapped to multiple IP addresses.
   C. The root servers have to be accessed for every DNS query.
   D. DNS servers listen to UDP port 53.
   E. Failure to contact DNS servers can cause disruption in access to Internet services.

4. A port number in TCP is _____ **bytes** long.
   A. 1
   B. 2
   C. 4
   D. 16
   E. 32

5. In a _____ network, data is first divided into manageable chunks before being sent.
   A. connection-oriented
   B. connection-less
   C. circuit-switching
   D. packet-switching
   E. telephone

6. The _____ layer of the Internet protocol stack is responsible for delivering data from sending process to receiving process.

   A. application
   B. transport
   C. network
   D. link
   E. physical

7. In HTTP, a response status code of 404 tells you

   A. Web server is unavailable
   B. Web server is currently busy
   C. your browser needs to be updated to the latest version
   D. the requested Web object is not found
   E. your HTTP request is malformed

8. It's said that a TCP Client/Server connection formation is "asymmetric" because a TCP server must exist before a TCP client can communicate with it. What can be said about UDP-based connection formation?

   A. A UDP client may send data to a non-existing UDP server without noticing that server is offline.
   B. A UDP server must exist before a client can send data to it. Otherwise client will encounter an exception.
   C. A UDP client and server must exchange control information before the client can send data to the server.
   D. Two UDP clients on one host cannot communicate with the same UDP server at the same time.
   E. None of the rest

9. Which of the following is a correct description of **`nslookup`**?

   A. It is used to check network connectivity to destination host.
   B. It is used to trace the network path between source and destination hosts.
   C. It is used to show network configuration of a host.
   D. It is used to find the DNS mapping between hostname and IP address.
   E. None of the rest

relies on timeout to handle packet loss & corruption

Sender
- recieve corrupted ACK ignore & wait for timeout to retransmit

Receiver
- receive corrupted pkt ignore, wait for timeout & sender to resend

10. In rdt 3.0, what does the sender do if it receives a corrupted ACK and what does the receiver do if it receives a corrupted data packet?

    A. Sender does nothing; receiver does nothing.
    B. Sender does nothing; receiver sends NAK.
    C. Sender resends data packet; receiver does nothing.
    D. Sender resends data packet; receiver sends ACK for the previous packet.
    E. None of the rest

11. Suppose there are multiple unacknowledged packets. Upon a timeout event, GBN sender retransmits _____ packet(s), SR sender retransmits _____ packet(s) and TCP sender retransmits _____ packet (s).

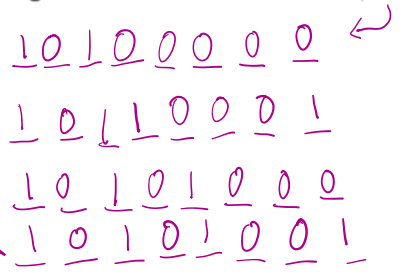
    A. One; one; one
    B. One; multiple; multiple
    C. Multiple; one; multiple
    D. Multiple; one; one
    E. None of the rest

12. In SR, ACK m means _____.

    A. Receiver has received all the packets up to packet m.
    B. Receiver has received all the packets up to packet m-1.
    C. Receiver has received packet m. But there is no implication on the receipt of other packets.
    D. The next in-order packet expected by receiver is packet m.
    E. None of the rest

13. How many of the following IP addresses belong to the subnet 192.168.160.0/20?

    i. 192.168.15.1
    ii. 192.168.177.254
    iii. 192.188.168.230
    iv. 192.168.169.31



A. 0
B. 1
C. 2
D. 3
E. 4

14. Study the following Python code snippet.

```
s = socket(AF_INET, SOCK_STREAM)
s.connect(("www.example.org", 12345))
```

means TCP client
makes new TCP connection
requires DNS query
runs over UDP
-no HTTP query

Suppose the above code snippet is executed with no error, which of the following protocols is NOT directly or indirectly invoked?

A. TCP
B. UDP
C. HTTP
D. DNS
E. None of the rest

15. A huge file is transferred over an existing TCP connection (i.e., 3-way handshake is already done). The connection is still open after transmission. The first and last TCP segments have the sequence numbers 12,345 and 2,105 respectively. MSS is 1,024 bytes and TCP sends as much data as possible in a segment.

    How many TCP segments are used to transfer the file (i.e. carries file data), assuming the communication channel is perfectly reliable?

    (Hint: TCP sequence number will wrap up and restart from 0 after reaching the biggest sequence number)

12345
$2^{32}-1$
2105

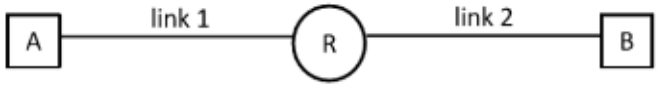
A. 10
B. 4,194,294
C. 4,194,295
D. 4,194,303
E. None of the rest

16. Consider a sender and a receiver communicating using Selective Repeat protocol. Every packet embeds a 3-bit sequence number field. Sender just sends a packet with sequence number 6. Sender window size is 3.

Which of the following CANNOT possibly be the sequence number of the next packet transmitted by sender?

A. 0
B. 2
C. 4
D. 5
E. 6

17. Two hosts **A** and **B** are connected by a router **R** as shown in the following diagram.



For link 1, link transmission rate is 1 Kbps and propagation delay is 100 milliseconds. For link 2, link transmission rate is 250 bps and propagation delay is 150 milliseconds. Suppose Host **A** sends 2000 packets to Host **B** continuously and each packet is 500 bits long. Host **A** starts sending the 1st packet at time $t = 0$.

When (in seconds) will host **B** receive the k^{th} packet ($1 \le k \le 2000$)?

A. $0.75 + 2k$
B. $2.75k$
C. $0.6 + 2k$
D. $0.6 + 2.15k$
E. None of the rest

18. Suppose two hosts are connected by a direct link of 1 Mbps. A stop-and-wait protocol is used to transfer 10 packets from the sending host to the receiving host. Each packet is 1000 bytes long. RTT is 24 milliseconds. No packet is lost or corrupted during transmission and ACK packets are of negligible size.

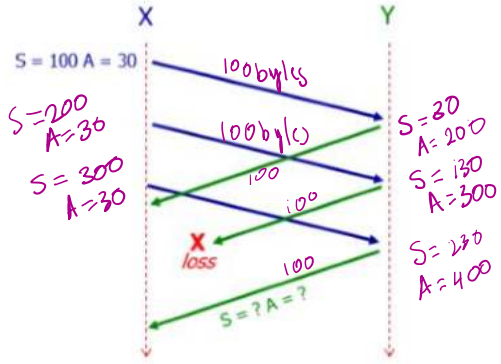
What is the throughput (in Kbps) of the transmission?

A. 40
B. 250
C. 770
D. 870
E. 1000

19. If a UDP segment contains no application data, what is the binary value of the "length" field in UDP header?

A. 0000 0000 0000 0000
B. 0000 0000 0000 1000
C. 0000 0000 0000 1111
D. 1111 1111 1111 1111
E. None of the rest

20. The following diagram shows two hosts **X** and **Y** communicating over a channel using TCP. **X** and **Y** are sending data to each other. Each TCP segment contains 100 bytes of data. None of the segments shown in the figure are retransmitted, out-of-order or corrupted packets. However, the second segment send by **Y** is lost. There are no other unacknowledged segments.



What would be the sequence number (S) and acknowledgement number (A) in the last segment sent by **Y**?

A. S = 300, A = 130
B. S = 130, A = 300
C. S = 400, A = 230
D. S = 230, A = 400
E. S = 230, A = 300

Suggested answers

1. E	8. A	15. C
2. B	9. D	16. B
3. C	10. A	17. A
4. B	11. D	18. B
5. D	12. C	19. B
6. B	13. B	20. D
7. D	14. C	

practice_paper_1

Tuesday, October 3, 2023 1:37 PM

practice_paper_1

1. Which of the following statements about client/server paradigm is TRUE?

   A. Client must always be alive.
   B. Server offers service while client requests for service from server.
   C. Only server can transmit data to client.
   D. Only client can transmit data to server.
   E. Server must run either DNS or HTTP protocol.

B

2. Which of the following is the most appropriate description of the service provided by UDP?

   A. Process-to-process communication
   B. Host-to-host communication
   C. End-to-end reliable data delivery ← TCP
   D. Guarantee on minimal throughput and timing
   E. Connection-oriented multiplexing and de-multiplexing

A

3. Which of the following statements regarding the Internet is TRUE?

   A. A call setup is always performed before data transmission starts.
   B. The Internet is structured as a network of networks.
   C. A packet passes through no more than two autonomous systems to reach destination host.
   D. The only access network technologies allowed are Ethernet and Wi-Fi.
   E. None of the rest

B

connectionless

4. A UDP server needs only one socket to communicate with *n* different clients. How many sockets would a TCP server have ever created for the same situation?

   A. 1
   B. *n*
   C. *n*+1
   D. 2*n*
   E. None of the rest

B wrong
C

listening socket + n sockets

• TCP is 1 to 1
• has a welcome socket also created

5. In a client/server connection using HTTP over TCP, if multiple objects are sent over the same TCP connection, then this connection is classified as _____.

   A. stateless
   B. stateful
   C. conditional
   D. persistent
   E. non-persistent

d

6. Consider sending a sequence of packets from a host in NUS to another host in NTU. Suppose packets may be of different length but all go through the same route to the destination. Which of the following end-to-end delay component is a constant (i.e. doesn't vary from packet to packet)? C

Same route = same propagation delay

A. Queueing delay – time pkt spends in routing queue
B. Transmission delay – time to push bits of packet into link
C. Propagation delay – time for signal to propagate through the link
D. Processing delay – time for router to process pkt header
E. None of the rest

7. Which of the following statements is TRUE when a packet containing application message is passed from router A to router B in the Internet?

A. Upon arrival, the packet may be discarded by B if B's buffer is full.
B. A and B must establish a TCP connection before the packet is transmitted.
C. A may pass the packet to B through a UDP connection. X
D. Circuit must be reserved before A can pass the packet to B.
E. None of the rest

8. Which of the following statements regarding TCP is TRUE?

A. If a TCP segment has sequence number *m*, then ACK for this segment will have acknowledgement number *m*. X
B. If a TCP segment has sequence number *m*, then ACK for this segment will have acknowledgement number *m* + 1. X
C. Host A is sending a file to host B over a TCP connection. If B has no data to send to A, B will not send ACK packets because B cannot piggyback the acknowledgments on data. X
D. TCP doesn't function correctly in a network that may re-order packets. X
E. None of the rest

9. Which of the following is a VALID subnet mask?

A. 255.250.255.0
B. 255.255.208.0
C. 255.240.0.0
D. 255.232.0.0
E. 127.0.0.0

1 1 1 0 0 0 0 0 = 240

10. In rdt3.0, what does the sender do if it receives a duplicate ACK and what does the receiver do if it receives a duplicate packet?

    A. Sender does nothing; receiver does nothing.
    B. Sender does nothing; receiver sends ACK for the previous packet.
    C. Sender resends data packet; receiver does nothing.
    D. Sender resends data packet; receiver sends ACK for the previous packet.
    E. None of the rest

11. GBN sender has _____ timer(s), SR sender has _____ timer(s) and TCP sender has _____ timer(s).

    A. One; multiple; one
    B. One; multiple; multiple
    C. One; one; one
    D. Multiple; multiple; multiple
    E. None of the rest

TCP has only 1 timer for oldest outgoing pkt

12. In GBN, ACK m means _____.

    A. Receiver has received all the packets up to packet m.
    B. Receiver has received all the packets up to packet m-1.
    C. Receiver has received packet m. But there is no implication on the receipt of other packets.
    D. The next in-order packet expected by receiver is packet m.
    E. None of the rest

13. Which of the following statements about TCP initial sequence number (ISN) is TRUE, given that sequence number field in TCP header is 32 bits?

    A. ISN is increased by 1 after sending every TCP segment.
    B. In bi-directional communication, both directions of communication must choose different ISNs.
    C. ISN determines the amount of data that can be transmitted over TCP.
    D. ISN is randomly chosen between [0, 2^{32}-1], both inclusive.
    E. None of the rest

14. **Telnet** protocol allows a user to establish a TCP connection to a remote server. Consider the following command.

```
telnet www.nus.edu.sg 80
```

Which of the following statement is TRUE?

i. The command causes a DNS lookup for the IP address of **www.nus.edu.sg**.
ii. The command causes a TCP SYN packet to be sent to **www.nus.edu.sg**.
iii. The command causes a HTTP request to be sent to **www.nus.edu.sg**.
iv. The command causes host **www.nus.edu.sg** to open port 80 and listen for incoming connections.

server should already be listening

telnet does not set HTTP requests only establishes TCP connection

A. (i) only
B. (i) and (ii) only
C. (i), (ii) and (iii) only
D. (ii), (iii) and (iv) only
E. (i), (ii) and (iv) only

15. Consider the following Python code snippet.

```
mySocket = socket(AF_INET, SOCK_STREAM)
mySocket.connect(('sunfire.comp.nus.edu.sg', 2105))
```

Suppose no runtime exception is raised, what port number is **mySocket** bound to when above statements finish execution?

in soln ans=C

A. It depends on the remote host's port that's making the connection.
B. TCP port 2105
C. Cannot say; it's operation system dependent and is usually a randomly chosen port.
D. UDP port 2105
E. None of the rest

16. A Go-Back-N sender just receives an ACK packet with ACK number 14. This ACK number falls within sender window which has the window size 6. Every data packet embeds a k-bit sequence number field (k is a constant unknown to you).

Which of the following definitely CANNOT be the sequence number of the next packet transmitted by the sender?

A. 4
B. 9
C. 15
D. 19
E. 20

17. Consider sending a packet over a path from node 0, through nodes 1, 2, ..., till node K+1. The links, from node i to node i+1, for i = 0, 1, ..., K each has the same link transmission rate C (in bits/s) and propagation delay p (in seconds). The packet has h header bits and L data bits.

    The delay D of a packet from node 0 to node K+1 is defined to be the duration from when the last bit of the packet leaves node 0 to when the last bit of the packet arrives at node K+1. Suppose the delay also includes a processing time of q seconds in each of the nodes 1, 2, ..., K. The processing time includes the waiting time in the queue.

    Which of the following formula correctly gives the delay D of a packet travelled from node 0 to node K+1?

    A. $D = p + K[(L + h)/C + p + q]$
    B. $D = (K + 1)[(L + h)/C + p + q]$
    C. $D = p + K[(L + h)/C] + (K + 1)q$
    D. $D = (K + 1)p + K[(L + h)/C] + q$
    E. None of the rest

18. A file of size 9,990 bytes is transferred over a TCP connection. The connection is still open after file transmission. Assume that MSS is 1,000 bytes, TCP sends as much data as possible in a segment.

    Assume TCP header is 20 bytes, what is the size of the last TCP segment (including TCP header and file data)?

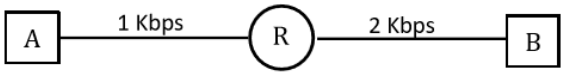
    A. 210
    B. 990
    C. 1,000
    D. 1,010
    E. None of the rest

19. What is the checksum (1's complement of the sum) of the following 3 bytes?

    ```
    11110100
    10010101
    11011101
    ```

    A. 10101100
    B. 01101000
    C. 10010111
    D. 10011001
    E. None of the rest

20. Two hosts A and B are separated by a router R in between. The bandwidth of the links between A and R and between R and B are 1 Kbps and 2 Kbps respectively. Ignore all other kind of delays. Suppose A sends $8*10^4$ bits to B as a series of consecutive packets of 1,000 bits each, when (in seconds) will B receive all the data?



A. 40
B. 80
C. 80.5
D. 161
E. None of the rest

last bit recieved at R at 80 sec

cus $\frac{8 \cdot 10^4}{1000} = 80$

second link is faster, so queue is empty for last packet to reach B, $\frac{1000}{2000} = 0.5$

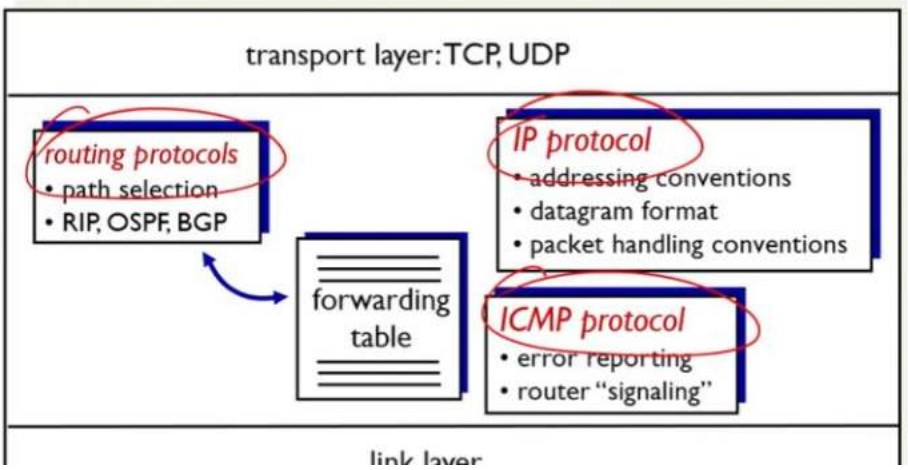
Suggested answers

1. B	8. E	15. C
2. A	9. C	16. B
3. B	10. B	17. A
4. C	11. A	18. D
5. D	12. A	19. C
6. C	13. D	20. C
7. A	14. B	

Lec 6 - network layer

Wednesday, October 4, 2023 12:26 AM

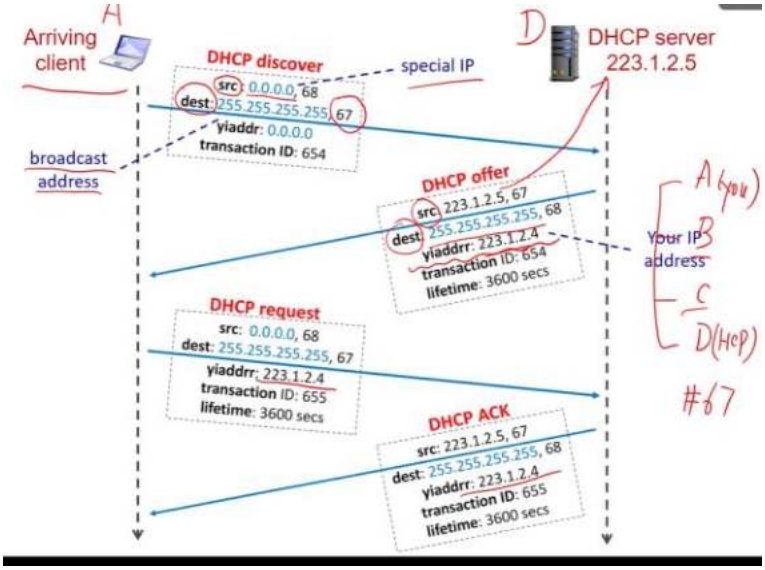
- Network layer delivers packets to receiving hosts
  - Routers examine header fields of IP datagrams passing it
  - network layer



- IP address identifies a host/router
  - Its 32-bit integer in binary or decimal
- A host gets IP address
  - Either manually configured by system admin or
  - Automatically assigned by a DHCP (dynamic host config protocol) server

DHCP

- Allows a host to dynamically obtain IP address from DHCP server when it joins network
- IP address is renewable
- Allow reuse of address (only hold address while connected). Once disconnected, reassigned to another computer
- Support mobile users
- 4 step process
  - Host broadcasts "DHCP discover" message
  - DHCP server responds with "DHCP offer" message
  - Host requests IP address "DHCP request" message
  - DHCP server sends address "DHCP ACK" message
- Arriving client's Src is special IP address (0.0.0.0) used temporarily
- Dest: is broadcase address. 255.255.255.255 - received by all hosts
  - Port 67 is on DHCP server that is listening for your discover msg



- Transaction id is like sequence number
- DHCP may also provide host additional information about network
  - IP address of first-hop router
  - Ip address of local DNS server
  - Network mask - network prefix versus host id of ip address
- DHCP runs over UDP
  - Server port = 67
  - Client port = 68

Special IP addresse

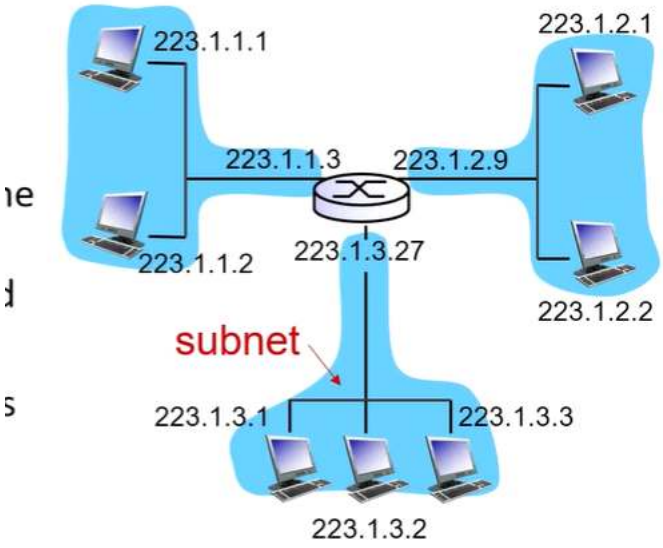
Special Addresses	Present Use
0.0.0.0/8	Non-routable meta-address for special use
127.0.0.0/8	Loopback address. A datagram sent to an address within this block loops back inside the host. This is ordinarily implemented using only 127.0.0.1/32.
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	Private addresses, can be used without any coordination with IANA or an Internet registry.
255.255.255.255/32	Broadcast address. All hosts on the same subnet receive a datagram with such a destination address.

s

-
- Private IP address is not unique

Network Interface

- An IP address is associated with a network interface
- A host usually has 1 or 2 network interfaces
- A router typically has multiple interfaces

- 

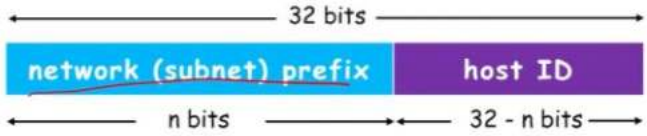
A network consisting of 3 subnets
(first 24 bits of IP addr. are network prefix)

IP Address

- Consists of 2 parts: network (subnet) prefix, and host ID

IP Address and Subnet

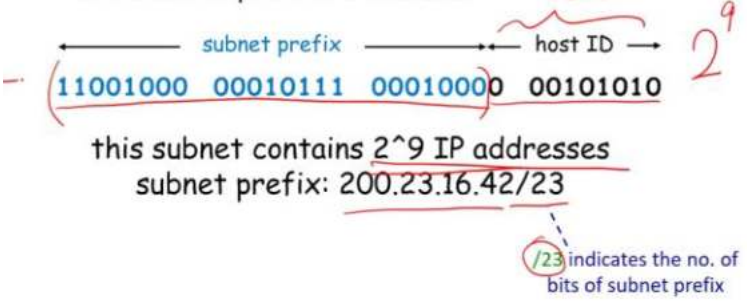
- An IP address logically comprises two parts:



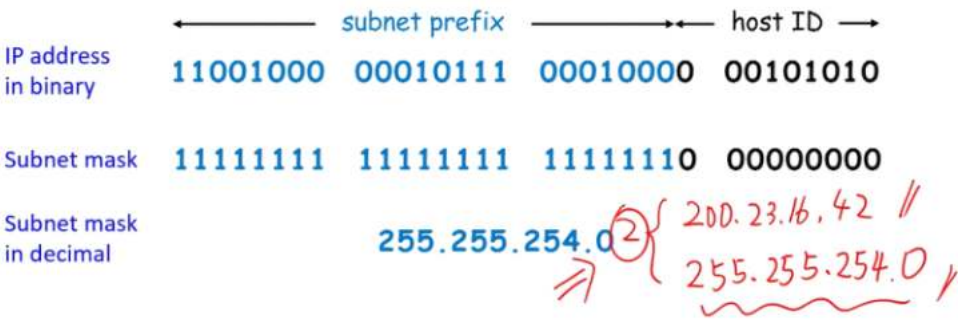
- Subnet is a network formed by a group of "directly" interconnected hosts.
  - Hosts in the same subnet have the same network prefix of IP address.
  - Hosts in the same subnet can physically reach each other without intervening router.
  - They connect to the outside world through a router.
- Intervening router is known as first hop router

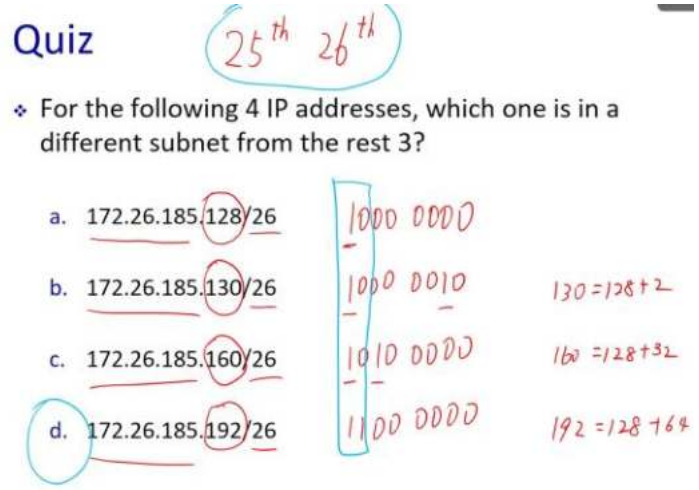
CIDR

- Internet's IP address assignment strategy is Classless Inter-domain routing
  - Subnet prefix of IP address is of arbitrary length
  - Address format: a.b.c.d/x where x is the # of bits in subnet prefix of IP address

- The Internet's IP address assignment strategy is known as Classless Inter-domain Routing (CIDR).
  - Subnet prefix of IP address is of arbitrary length.
  - Address format: **a.b.c.d/x**, where **x** is the number of bits in subnet prefix of IP address.
- 

Subnet Mask

- Used to determine which subnet an IP address belongs to
  - Made by setting all subnet prefix bits to "1"s and host ID bits to "0"s
- 

- 

  - An organization obtains block of IP addresses from registry or rent from ISP's address space

IP Address Allocation

- Q: How does an organization obtain a block of IP addresses?
- A: Buy from registry or rent from ISP's address space.

	Binary Address	Decimal Address
ISP's block	11001000 00010111 0001 000 0 00000000	200.23.16.0/20
Organization 1	11001000 00010111 0001 000 0 00000000	200.23.16.0/23
Organization 2	11001000 00010111 0001 001 0 00000000	200.23.18.0/23
Organization 3	11001000 00010111 0001 010 0 00000000	200.23.20.0/23
...	...	...
Organization 6	11001000 00010111 0001 101 0 00000000	200.23.28.0/23

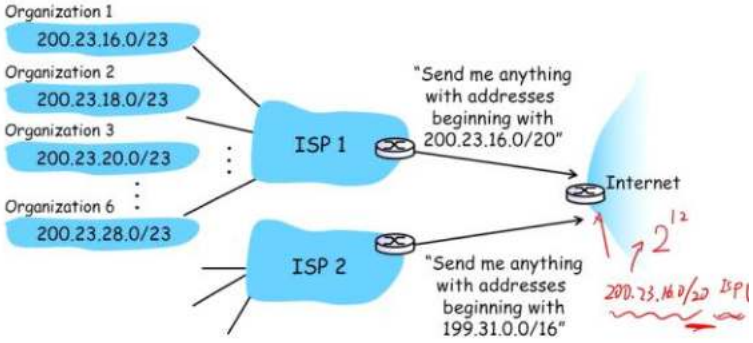
Unused < 110, 111 — use 3 more bits to differentiate 6 organizations

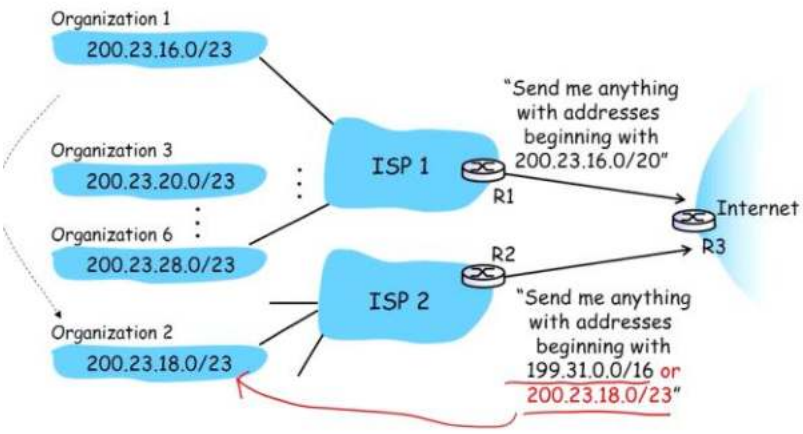
- Each organization can do further allocation itself. But cannot change the prefix

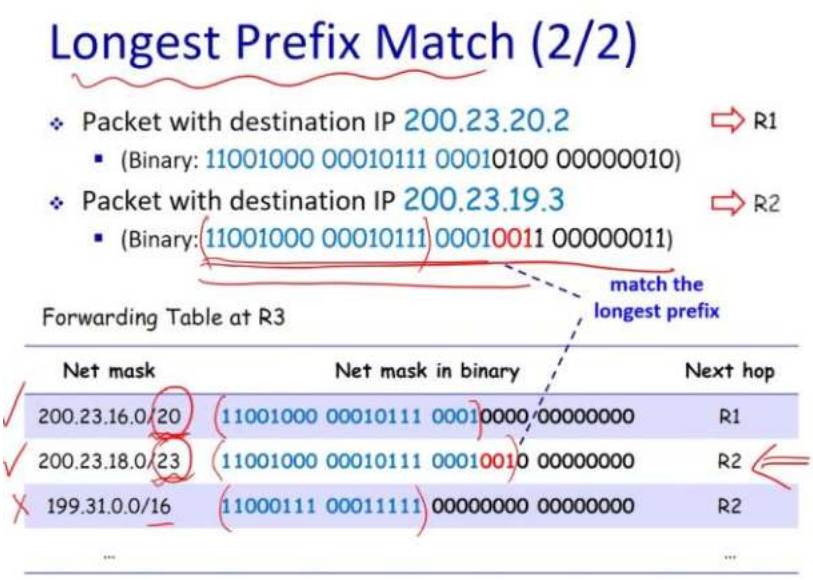
Hierarchical Addressing

Hierarchical Addressing

Hierarchical addressing allows efficient advertisement of routing information:



- 

- 

- You would have 2 matches if a organization switched ISP
  - Then new ISP would have longer match
- ISP gets block of address from ICANN
  - ICANN allocates addresses, manages DNS etc.

Tutorial_5_qns

Thursday, October 5, 2023 6:38 PM

Tutorial_5_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 5** Question paper

1. **[KR, Chapter 4, R13]** What is the 32-bit binary equivalent of the IP address 202.3.14.25?

   11001010 00000011 00001110 00011001

2. **[KR, Chapter 4, 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. Assume TCP header is 20 bytes and IP header is another 20 bytes, what percentage of each datagram will be overhead, and what percentage will be application data?

50%

3. Combine the following three blocks of IP addresses into a single block:
   a) 16.27.24.0/26
   b) 16.27.24.64/26
   c) 16.27.24.128/25

16.27.24.0/24

4. **[Modified from KR, Chapter 4, P16]**
   a) Consider a subnet with network prefix 192.168.56.128/26. Give an example IP address (of form xxx.xxx.xxx.xxx) that belongs to this network.

   128+0 10000000 192.168.56.129/26

First and last IP are reserved

   b) Suppose an ISP owns the block of addresses of the form 192.168.56.128/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the network prefixes (of form a.b.c.d/x) for the four subnets?

2^6 addresses $\frac{2^6}{2^2} = 2^4 = 32$

1

Network Prefix	Binary Expression
192.168.56.128/28	
192.168.56.129/28	
192.168.56.130/28	
192.168.56.131/28	

1000 0000 128/28
1001
1010
1011

Page 1 of 2

5. **[KR, Chapter 4, P7]** Consider a datagram network using 8-bit addresses. Suppose a router has the following forwarding table:

Prefix Match	Interface
11	0
101	1
100	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.

Prefix Match	Interface	IP Range	No. of IP
11	0	11000000 - 11111111	2^6
101	1		2^5
100	2		2^5
otherwise	3		$2^8 - 2^6 - 2(2^5)$

10100000 - 10111111

10000000 - 10011111

0 |0000000 - 01111111

6. What is private IP address? Does Canvas use private or public IP? When your laptop is connected to NUS network, does it receive a private or public IP?

private

Canvas - public
NUS - private

Lec 7

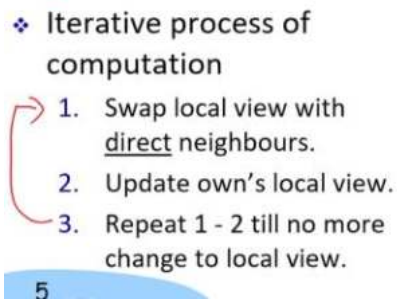
Friday, October 6, 2023 8:05 PM

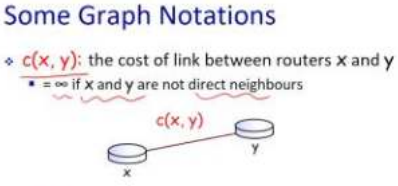
Network Layer continued

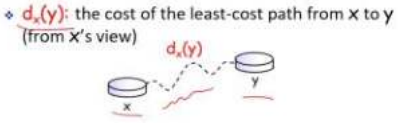
Routing Algorithms
- Internet is network of networks
- Routing on internet is done hierarchically

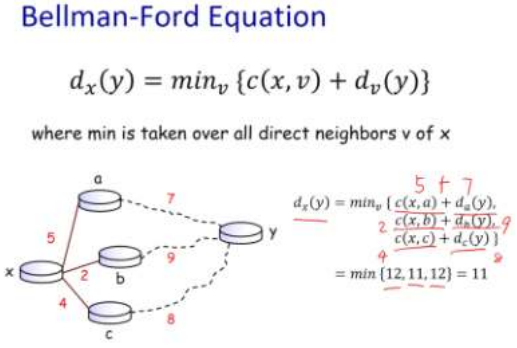
Intra-AS (autonomous) routing -
- Finds a good path between 2 routers within an AS
- Commonly used protocols are RIP, OSPF
- Single admin, so no policy decision needed
- Routing mostly focus on performance

Inter-AS routing - do not cover in module - handles interfaces between Ass. The de factor standard protocol - BGP
Decided by policy

Intra-AS Routing
- We associate a cost to each link ,and try to find shortest path
- All routers have complete knowledge of network topology and link cost
- Routers periodically broadcast link costs to each other
- Link state algorithms
- Use dijkstra algorithm to compute least cost path locally
- "distance vector" algorithm
  - Routers know physically-connected neighbors and link costs to neighbors
  - Routers exchange "local view" with direct neighbors and update own "local views" based on neighbors' view
  - 

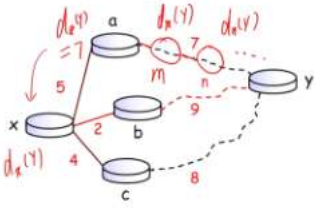
- 

- 

- Distance vector algo is based on Bellman-Ford
  - 

Bellman-Ford Equation

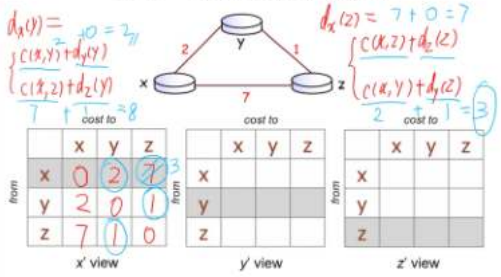
- To find the least cost path, x needs to know the cost from each of its direct neighbour to y.
- Each neighbour v sends its distance vector (y, k) to x, telling x that the cost from v to y is k.



Now x knows, to reach y, packet should be forward to b and the total cost would be 11.

Bellman-Ford Example

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



- X->Z is updated to 3 after knowing Y->Z

Distance Vector Algorithm

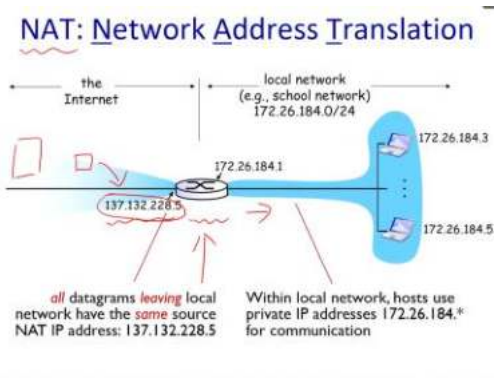
- Every router, **x**, **y**, **z**, sends its distance vectors to its directly connected neighbors.
- When **x** finds out that **y** is advertising a path to **z** that is cheaper than **x** currently knows,
  - **x** will update its distance vector to **z** accordingly.
  - In addition, **x** will note down that all packets for **z** should be sent to **y**. This info will be used to create forwarding table of **x**.
- After every router has exchanged several rounds of updates with its direct neighbors, all routers will know the least-cost paths to all the other routers.

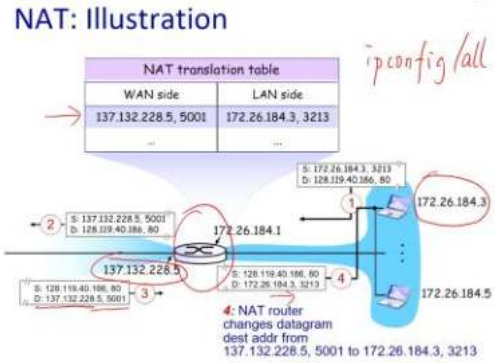
RIP - Routing Information Protocol

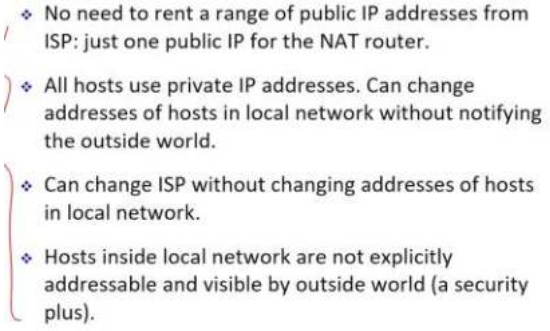
- Implements the DV algo
- Uses hop count as the cost metric ie # of links
- Insensitive to network congestion
- Exchange routing table every 30 seconds over UDP port 520
- "self-repair" if no update from neighbor router for 3 min, assume neighbor has failed

NAT - Network Address Translation

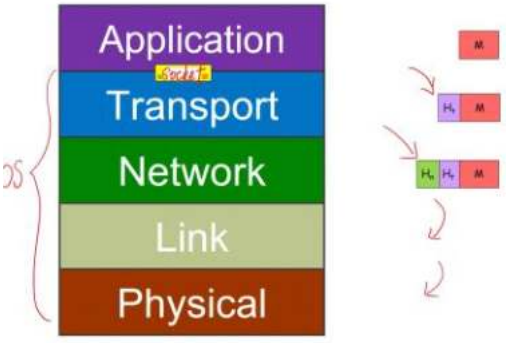
- There are public IP, and private IP. Public IP are limited and unique
- Many IP addresses are reserved
- But there are not enough public Ips
- Ex. My computer in NUS network can have same IP as someone else's computer in NTU network
- So within the local network, private IP's are used. But then they are translated through NAT to a public IP when they need to transmit through internet

- 

- NAT routers must:
  - Replace (source IP address, prot) of every outgoing data gram to (NAT IP, new port)
  - Remember mapping ^
  - Replace (NAT IP, new port #) in destination field of every incoming datagram with corresponding (source IP, port #) stored in NAT translation table
  - If port number was not changed by NAT, then computers in local network could have same port number, then the recevied package at router would be confused.
  - 

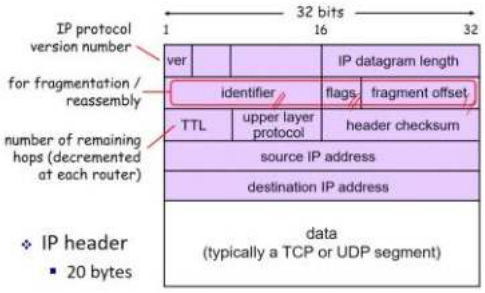


Internet Protocol



- The layers add a header

IPv4 Datagram Format

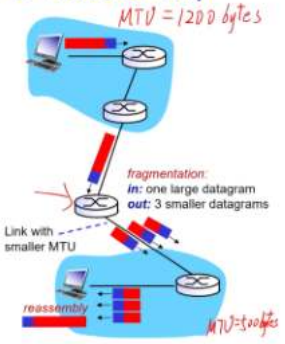


(some fields are not shown)

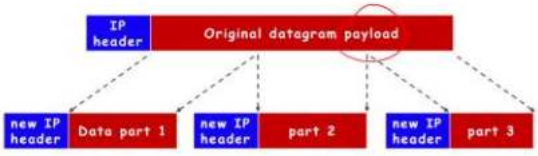
- Different links may have different MTU - max transfer unit
  - Maximum amount of data a link-level frame can carry
  -

IP Fragmentation & Reassembly

- Different links may have different MTU (Max Transfer Unit) – the maximum amount of data a link-level frame can carry.
- "Too large" IP datagrams may be fragmented by routers.



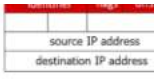
IP Fragmentation Illustration



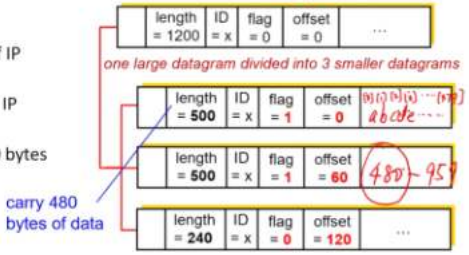
- Destination host will reassemble the packet.
- IP header fields are used to identify fragments and their relative order.

  - Byte number is always divisible by 8 for offset value

IP Fragmentation



- Flag (frag flag) is set to
  - 1 if there is next fragment from the same segment.
  - 0 if this is the last fragment.
- Offset is in expressed in unit of 8-bytes. 480/8
- Example
  - 20 bytes of IP header
  - 1,200 byte IP datagram
  - MTU = 500 bytes



1180 byte data

480 data + 20

480 data +20

220 data +20

ICMP (Internet Control Message Protocol)

- Used by hosts and routers to communicate netwrok-level inf
  - Error reporting (unreachable host/network/port
  - Echo request/reply ping
- Messages carried in IP datagrams
  - ICMP header starts after IP header

ICMP Type and Code

- ICMP header: Type + Code + Checksum + others.

Type	Code	Description
8	0	echo request (ping)
0	0	echo reply (ping)
3	1	dest host unreachable
3	3	dest port unreachable
11	0	TTL expired
12	0	bad IP header

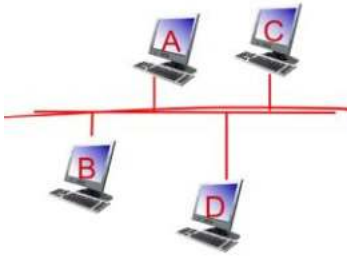
Selected ICMP Type and subtype (Code)

- Ping and traceroute cmd
- Ping checks if remote host will respond tous
- Traceroute sends a series of small packets across a netwrok and attempts to display the route that the msg would take to get to a remote host (tracert)

Lec 8/wk 9

Monday, October 16, 2023 2:09 PM

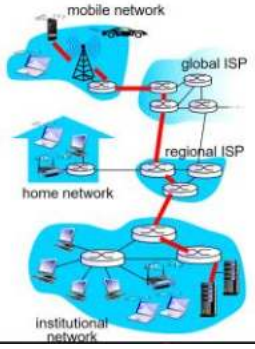
Link layer
Aim to send data between 2 nodes via cable

- Connect the 2 nodes and send data
- Send data between N nodes via cable
    - Inter-connect the N nodes and send data
    - Inter connect the N nodes via a broadcast link
        - A bus
        - 

        - Each link needs to be addressed tho
        - Only need 1 port at each node
        - Need a protocol
        - Every packet is received by everyone
        - Need to handle errors
        - Abstract shared link

Link Layer: Introduction (1/2)

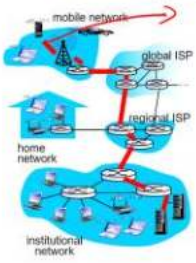
- *Network layer* provides communication service between any two hosts.
- An IP datagram may travel through multiple routers and links before it reaches destination.

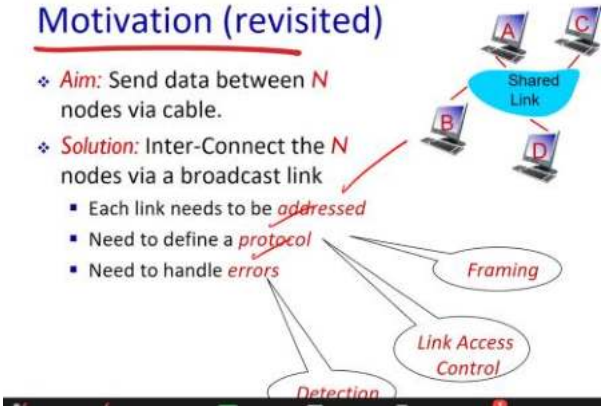


Link Layer: Introduction (2/2)

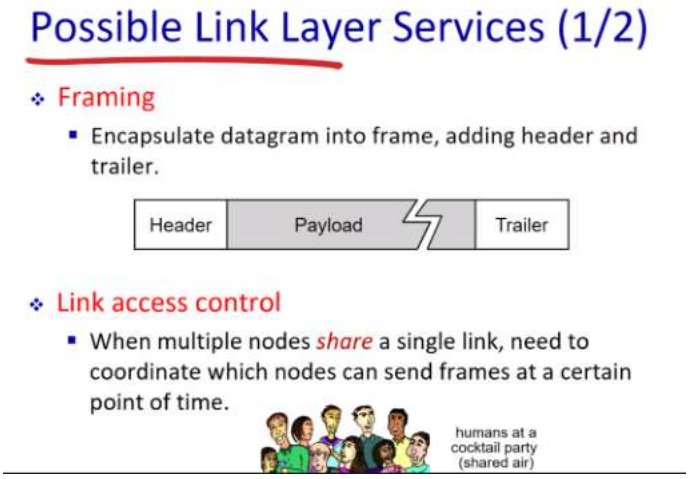
- *Link layer* sends datagram between adjacent nodes (hosts or routers) over a single link.
    - IP datagrams are encapsulated in link-layer frames for transmission.
    - Different link-layer protocols may be used on different links.

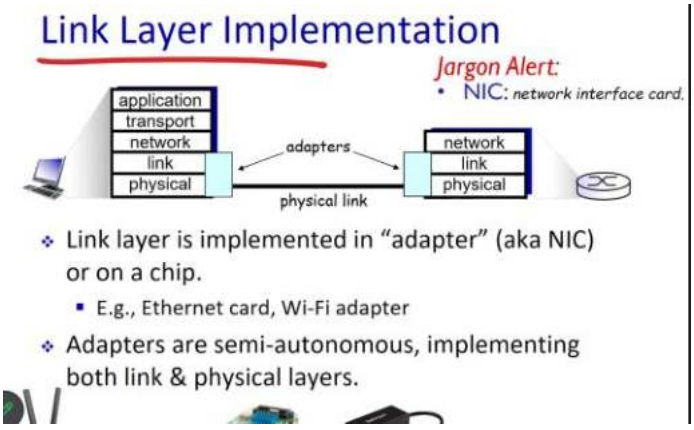
Jargon Alert:
- Adjacent: *A single hop connects the two nodes.*
- Frame: *layer-2 packet.*



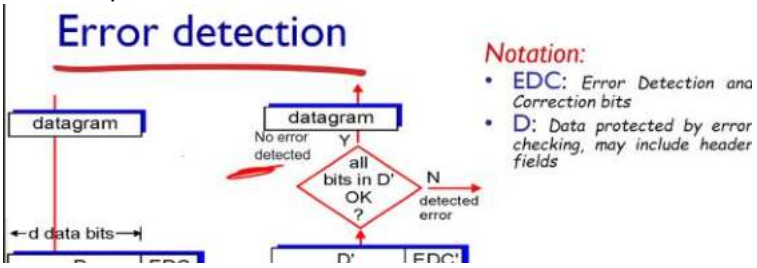


Framing

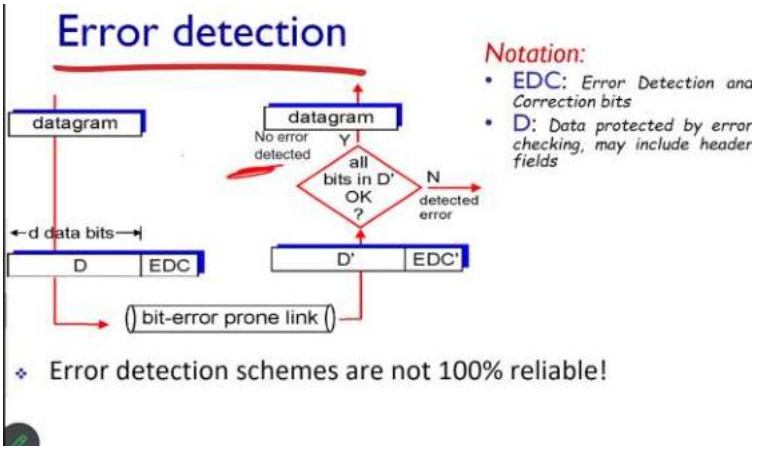




Error detection

- Bit error prone link
- 

Error detection

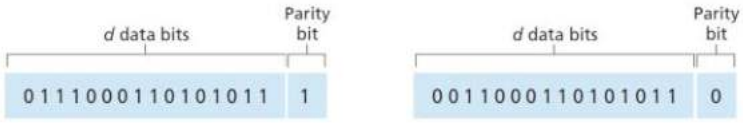


Notation:
- EDC: *Error Detection and Correction bits*
- D: *Data protected by error checking, may include header fields*

- Error detection schemes are not 100% reliable!

Parity Checking: Single bit

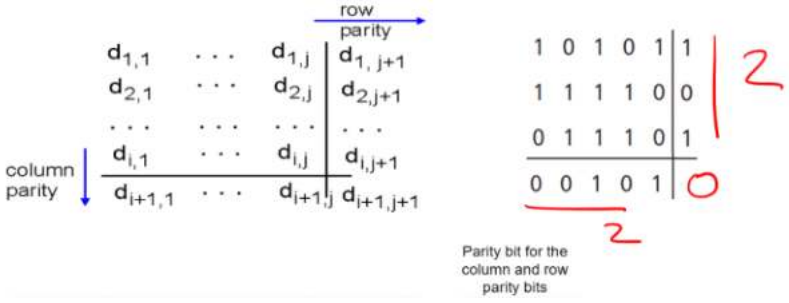
- Suppose that the information to be sent, *D*, has *d* bits.
- In an *even parity* scheme,
  - the sender simply includes one additional bit
  - chooses its value such that the total number of 1s in the *d + 1* bits is even



- Can detect single bit errors and odd # of single bit errors
- Probability of multiple bit errors is low
- However, errors are ofter clustered tgt in bursts
- Probability of undetected errors in a frame can approach 50%

Parity Checking 2-D

- The *d* bits in *D* are divided into *i* rows and *j* columns.
- A parity value is computed for each row and for each column.
  - The resulting *i + j + 1* parity bits comprise the link-layer frame's error-detection bits



- Can detect and correct single bit errors
- Can detect but not correct 2 bit errors

Parity Checking 2-D

Parity Checking 2-D

- Can *detect and correct* single bit errors in data.

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

1	0	1	0	1	1	
1	0	1	1	0	0	Parity error
0	1	1	1	0	1	
0	0	1	0	1	0	

Parity error

- Can *detect* any two-bit error in data.

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	0	0	1
0	0	1	0	1	0

Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check: Motivati

- Let $D = 21027845$, $r = 3$ and $G = 401$.
- Create a new number X by appending r $9's$ to D
  - $X = 21027845\ \mathbf{999}$
  - Mathematically, $X = D \times 10^r + (10^r - 1)$
- Find the remainder y of $\frac{X}{G}$
  - $y = X\ \%\ G$
  - $y = 281$
- The message M being transmitted is
  - $M = X - y$
  - $M = 21027845999 - 281 = 21027845718$

Cyclic Redundancy Check: Motivation

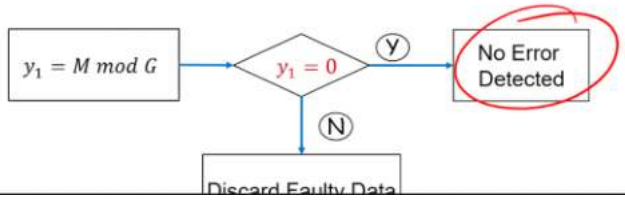
- $M = 21027845\ 718$

21027845 | 718

8 digit number | R:3 digits

- $D = 21027845$
- $r = 3$
- $G = 401.$
- $X = 21027845\ 999$
- $y = 281$

- On the Receiver end, we find the remainder



Tutorial_6_qns

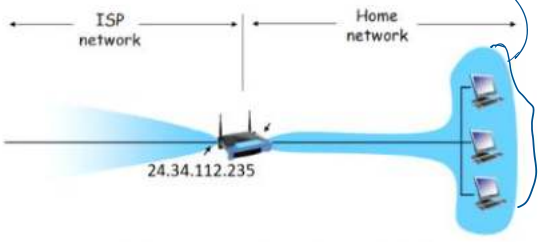
Thursday, October 19, 2023 6:25 PM

Tutorial_6_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 6** Question paper

1. **[Modified from KR, Chapter 4, P21]** Consider the network setup in the following figure. Suppose that the ISP assigns the router the address 24.34.112.235 and that the network address (i.e. network prefix) of this home network is 192.168.1/24.



a) Give an example IP address assignment to all interfaces in this home network.

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

NAT Translation Table	
WAN side	**LAN side**

Page 1 of 3

Router has 2 interfaces + 3 interfaces for hosts
1st and second IP address are unavailable

Port # should be greater than 2^10 and less than 2^16
Max outgoing connections is 2^16

Add a destination field to nat table to increase number of outgoing connections. As same port number can be used with diff destination address

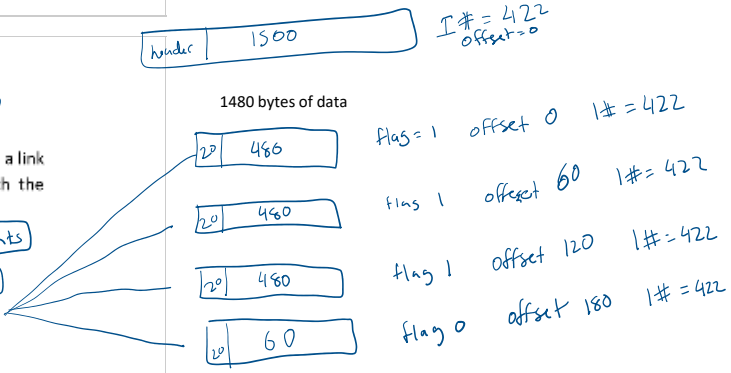
broadcast address
192.168.1.255

2. **[Modified from KR, Chapter 4, P19]** Consider sending a 1500-byte IP datagram into a link that has an MTU of 500 bytes. Suppose the original datagram is stamped with the identification number 422. Also assume that IP header is 20 bytes long.

a) How many fragments will be generated? 1500/480 = 3.124 → 4 fragments

b) What is the length of each fragment (including IP header)? 500 bytes

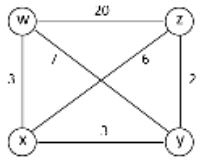
c) What are the values of *identification number*, *offset* and *flag* in each fragment?



3. **[CS2105 Final Exam, April 2006]** The following diagram shows a simple network topology with 4 nodes. The links in the diagram are labeled with the cost of each link. The nodes run distance vector routing protocol. The protocol has terminated, and each node knows the cost of the minimum cost path to every other node.

TCP segmentation avoid IP segmentation becuz otherwise IP fragmentation will separate the TCP header with packets with no

3. **[CS2105 Final Exam, April 2006]** The following diagram shows a simple network topology with 4 nodes. The links in the diagram are labeled with the cost of each link. The nodes run distance vector routing protocol. The protocol has terminated, and each node knows the cost of the minimum cost path to every other node.



The following table shows an incomplete distance vector table at node x. Fill in the missing distance vectors.

	cost to w	cost to x	cost to y	cost to z
from x	3	0	3	5
from y	6	3	0	2
from z	8	5	2	0

4. Wireshark: IP

Do the following:

1. Start up Wireshark and begin packet capture.
2. Start up the Terminal and execute the following command:
   ping www.google.com (Windows)
   OR
   Ping www.google.com -c 4 (Linux/mac)

TCP segmentation avoid IP segmentation becuz otherwise IP fragmentation will separate the TCP header with packets with no header. So the segments are made with the lowest MTU in world and headers are added

3. Stop Wireshark packet capture.
4. Type in "icmp" into the display filter specification window, then select Apply.

Answer the following questions:

1. Within the IP packet header, what is the value in the upper layer protocol field?
2. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

Tutorial_7_qns

Friday, October 27, 2023 2:30 PM

Tutorial_7_
qns

National University of Singapore
School of Computing

CS2105 **Tutorial 7** Question paper

---

1. **[KR, Chapter 6, 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?

2. **[KR, Chapter 6, P5/P6]** Consider a 4-bit generator G with value `1001`, what is the CRC checksum R if data D has the following value?

   a) `11000111010`

   b) `01101010101`

   c) `11111010101`

   d) `10001100001`

3. Consider the following two-dimensional parity matrix.

   0 1 0 1
   1 0 1 0
   0 1 0 1
   1 0 1 0

   a) Give an example of a 1-bit error that can be detected and corrected.

   b) Give an example of a 2-bits error that can be detected but cannot be corrected.

   c) Give an example of a 4-bits error that cannot be detected.

4. There are many nodes in a shared medium network and most nodes are likely to transmit frequently. Which of the following multiple access protocol(s) is (are) suitable? (1) TDMA; (2) CSMA; (3) Token passing.

5. Nodes A and B are accessing a shared medium using CSMA/CD, with propagation delay of 245 bit times between them (i.e., propagation delay equals to the amount of time to transmit 245 bits). Minimum frame size is 64 bytes. Suppose node A begins transmitting a frame at $t = 0$ bit time. Before A finishes, node B begins transmitting a frame. Assume no other nodes are active.

   Write down your answers to the following 2 questions in the unit of **bit time (time taken to transmit one bit)**.

   a) When is the latest time, by which B can begin its transmission?

   b) Suppose B begin its transmission at the time computed in a), can A detects that B has transmitted before it finishes transmission?

Link layer

Lec_8_Link_Layer

CS2105

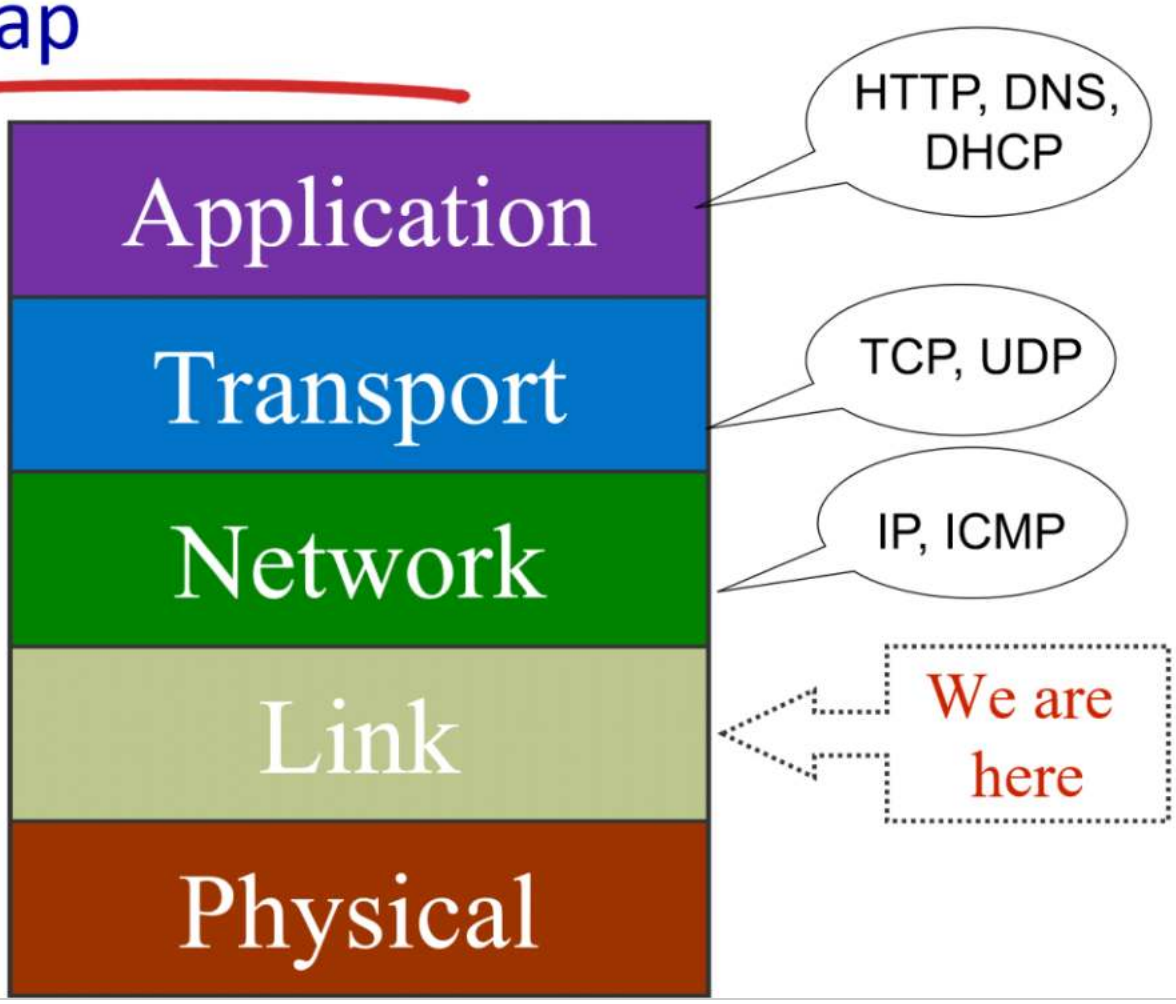
An *Awesome* Introduction to Computer Networks

The Link Layer

Department of Computer Science
School of Computing

Adopted from Prof Roger

Recap


Application
Transport
Network
Link
Physical
HTTP, DNS, DHCP
TCP, UDP
IP, ICMP
We are here

Motivation

- Let us look at networking from Bottom-up perspective
- Assuming, we have figured out the electronics of sending and receiving binary data over a *communication channel*
- *Aim:* Send data between 2 nodes via cable.
- *Solution:* Connect the 2 nodes and send data

Jargon Alert:

- Communication channel: the transmission medium of the data signals. E.g. copper wire, optical fiber, terrestrial radio, Satellite, etc.
- Node: Devices exchanging data. E.g. hosts, routers, etc.

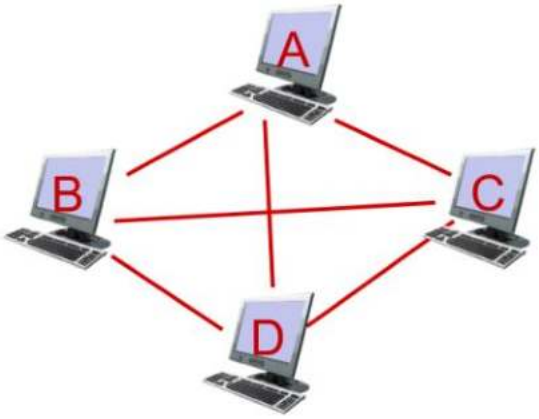
Physical

Motivation

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes and send data
  - Each link needs to be *addressed*
  - *Drawback*: Does not scale
    - N-1 links needed

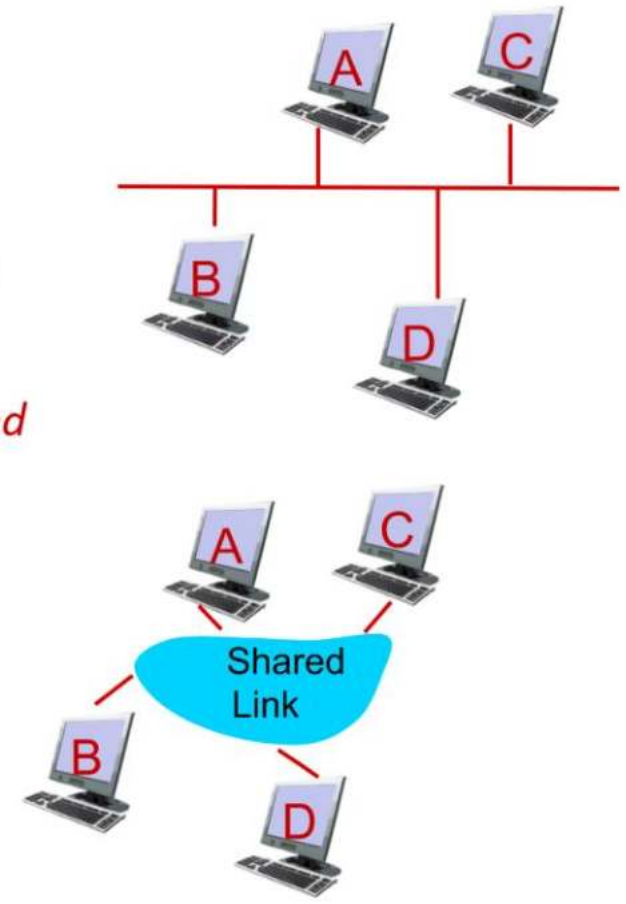
Jargon Alert:

- Link: *Communication channels that connect adjacent nodes.*



Motivation

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*



The Link Layer

After the next set of lectures, will understand:

- The role of link layer and the services it could provide.
- How parity and CRC scheme work.
- Different methods for accessing shared medium.
- How ARP allows a host to discover the MAC addresses of other nodes in the same subnet.
- The role of switches in interconnecting subnets in a LAN.

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

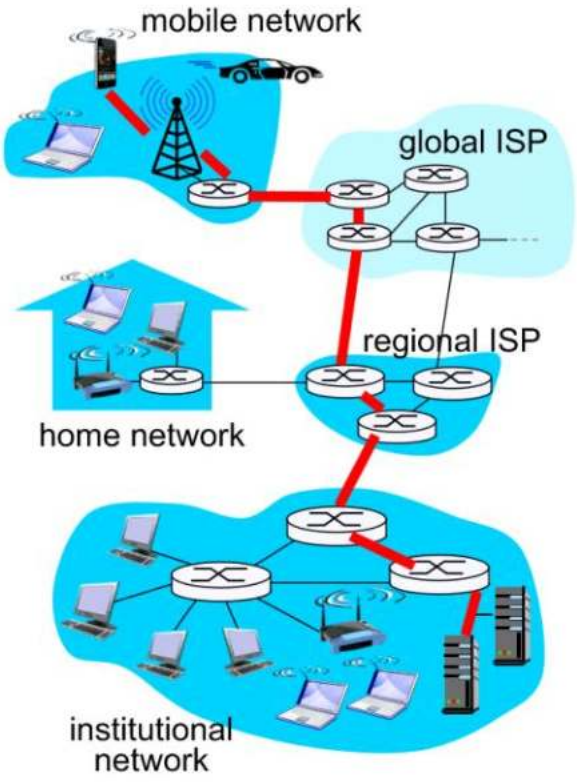
6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

Kurose Textbook, Chapter 6
(Some slides are taken from the book)

Link Layer: Introduction (1/2)

- *Network layer* provides communication service between any two hosts.
- An IP datagram may travel through multiple routers and links before it reaches destination.



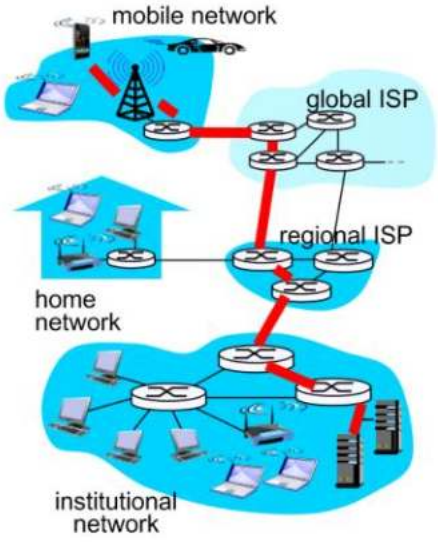
Link Layer: Introduction (2/2)

- *Link layer* sends datagram between adjacent nodes (hosts or routers) over a single link.
  - IP datagrams are encapsulated in link-layer frames for transmission.
  - Different link-layer protocols may be used on different links.
    - each protocol may provide a different set of services.

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

Jargon Alert:

- Adjacent: *A single hop connects the two nodes.*
- Frame: *layer-2 packet.*



Motivation (revisited)

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

Link Access Control

Reliability

Detection

Possible Link Layer Services (1/2)

- Framing
  - Encapsulate datagram into frame, adding header and trailer.

- Link access control
  - When multiple nodes *share* a single link, need to coordinate which nodes can send frames at a certain point of time.

humans at a cocktail party (shared air)

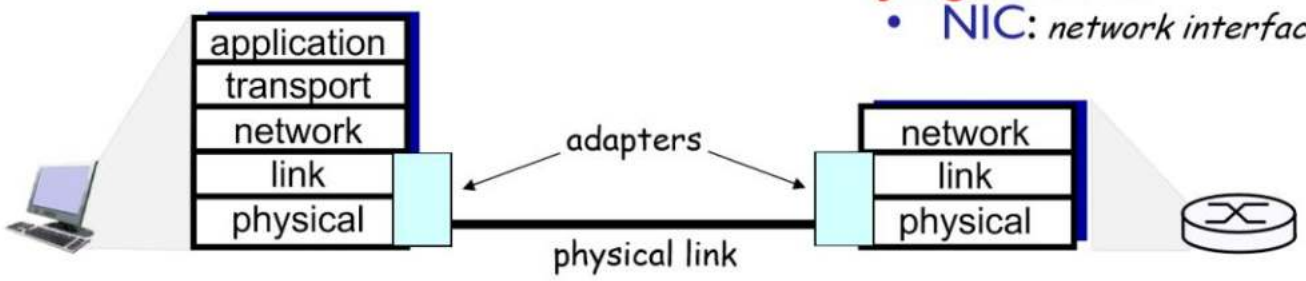
Possible Link Layer Services (2/2)

- Error detection
  - Errors are usually caused by signal attenuation or noise.
  - Receiver detects presence of errors.
    - may signal sender for retransmission or simply drops frame
- Error correction
  - Receiver identifies and corrects bit error(s) without resorting to retransmission.
- Reliable delivery
  - Seldom used on low bit-error link (e.g., fiber) but often used on error-prone links (e.g., wireless link).

Link Layer Implementation

Jargon Alert:

- NIC: *network interface card.*



- Link layer is implemented in "adapter" (aka NIC) or on a chip.
  - E.g., Ethernet card, Wi-Fi adapter
- Adapters are semi-autonomous, implementing both link & physical layers.

Roadmap

6.1 Introduction to the Link Layer

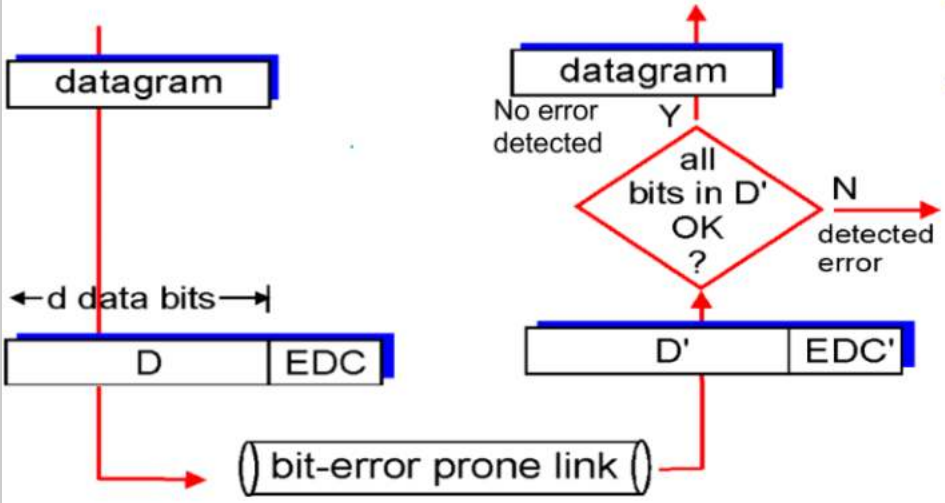
6.2 Error Detection and Correction

- 6.2.1 Parity Checks
- 6.2.3 Cyclic Redundancy Check (CRC)

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

Error detection



Notation:

- EDC: *Error Detection and Correction bits*
- D: *Data protected by error checking, may include header fields*

- Error detection schemes are not 100% reliable!
  - may miss some errors, but rarely.
  - Usually, larger EDC field yields better detection (and even correction).

Error Detection

- Popular error detection schemes:
  - Checksum (used in TCP/UDP/IP)
  - Parity Checking
  - CRC (commonly used in link layer)

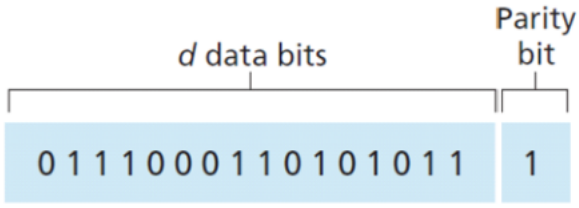
- Checksum (review)
  - treat segment contents as sequence of 16-bit integers
  - *checksum*: 1's complement of the sum of segment contents

Parity Checking: Single bit

- Suppose that the information to be sent, *D*, has *d* bits.
- In an *even parity* scheme,
  - the sender simply includes one additional bit
  - chooses its value such that the total number of 1s in the *d + 1* bits is even

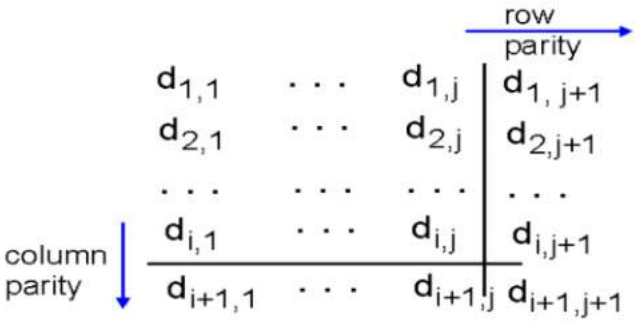
Parity Checking: Single bit

- Can *detect* single bit errors in data.
  - Actually, can detect odd number of single bit errors
  - Cannot detect even number of single bit error
- Works exceptionally well (Mathematically)
  - Probability of multiple bit errors is low (if errors are independent)
- However, errors are often clustered together in "bursts."
  - The probability of *undetected errors* in a frame can approach *50%*



Parity Checking 2-D

- The *d* bits in *D* are divided into *i* rows and *j* columns.
- A parity value is computed for each row and for each column.
  - The resulting *i + j + 1* parity bits comprise the link-layer frame's error-detection bits



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

$d_{i+1,1} \ldots d_{i+1,j} d_{1,j+1} \ldots d_{i,j+1}$

Parity bit for the column and row parity bits

$d_{i+1,j+1}$

Parity Checking 2-D

- Can *detect and correct* single bit errors in data.

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Parity error

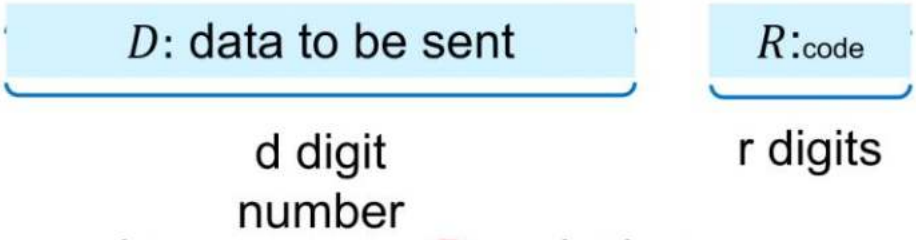
Parity error

- Can *detect* any two-bit error in data.

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	0	0	1
0	0	1	0	1	0

Cyclic Redundancy Check: Motivation

- We want to transfer a non-binary number D without error.
- R: the r digit error detection code.



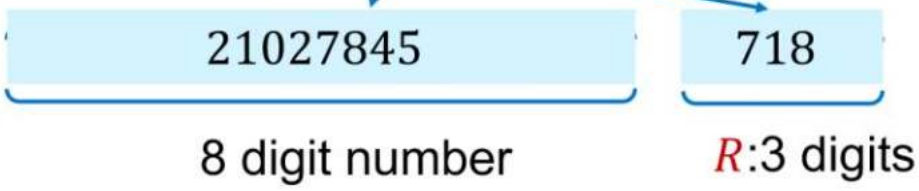
- *Aim:* We need to generate R such that
  - The sender can compute R easily.
  - The receiver can verify the integrity of D easily.
- *Solution:* Let us use the mathematical properties of "division".
  - We shall use a special r digit number G, called the "Generator".

Cyclic Redundancy Check: Motivation

- Let $D = 21027845$, $r = 3$ and $G = 401$.
- Create a new number X by appending r $9's$ to D
  - $X = 21027845\ \mathbf{999}$
  - Mathematically, $X = D \times 10^r + (10^r - 1)$
- Find the remainder y of $\frac{X}{G}$
  - $y = X\ \%\ G$
  - $y = 281$
- The message M being transmitted is
  - $M = X - y$
  - $M = 21027845999 - 281 = 21027845718$
  - M is divisible by G

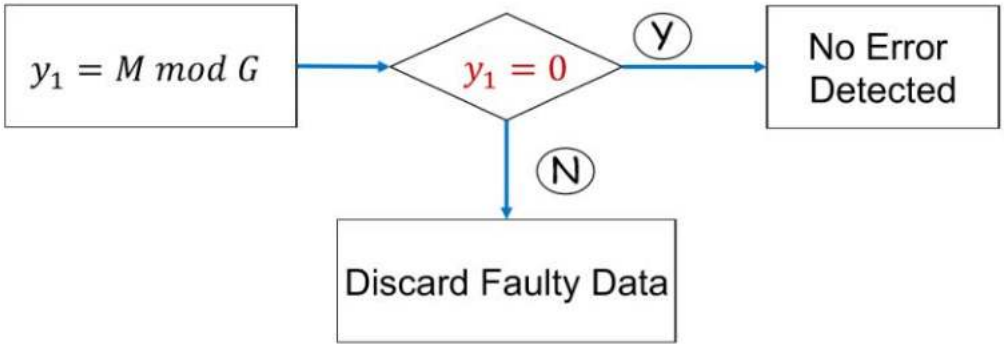
Cyclic Redundancy Check: Motivation

- $M = 21027845\ 718$



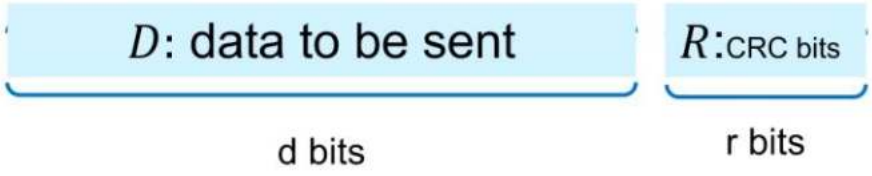
- $D = 21027845$
- $r = 3$
- $G = 401.$
- $X = 21027845\ \mathbf{999}$
- $y = 281$

- On the Receiver end, we find the remainder



Cyclic Redundancy Check (CRC)

- D: data bits, viewed as a binary number.
- G: generator of $r + 1$ bits, agreed by sender and receiver beforehand.
- R: the r bit CRC.



Cyclic Redundancy Check (CRC)

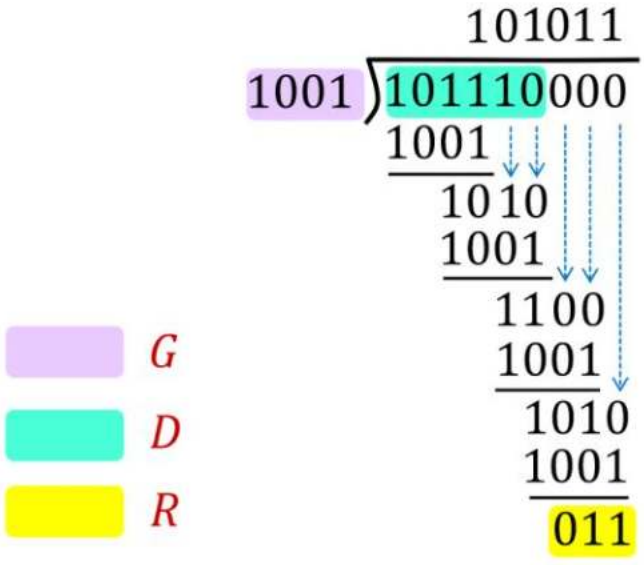
- Calculations are done modulo 2.
  - It does not have carries for addition or borrows for subtraction.
  - Both addition and subtraction are *identical* to XOR
    - $x + y = x - y = x \oplus y$
    - $0 + 1 = 0 - 1 = 0 \oplus 1 = 1$
    - $1011 \oplus 0101 = 1110$
    - $1001 \oplus 1101 = 0100$

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

- $1011 - 0101 = 1110$
- $1001 - 1101 = 0100$

Cyclic Redundancy Check (CRC)

- For performing division, we append r 0's to D.
- Because of the properties of modulo 2 arithmetic, The remainder directly gives us R



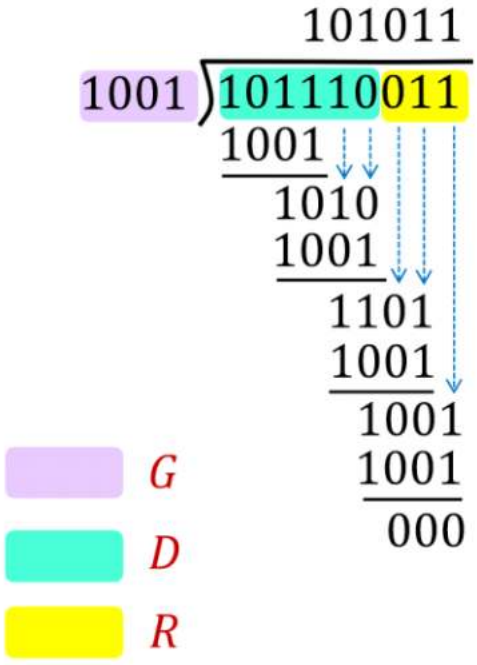
E.g.
$D = 101110$,
$r = 3$
$G = 1001$

Cyclic Redundancy Check (CRC)

- Sender sends (D, R)

  101110011

- Receiver knows G, divides (D, R) by G.
  - If non-zero remainder: error is detected!



G

D

R

Cyclic Redundancy Check (CRC)

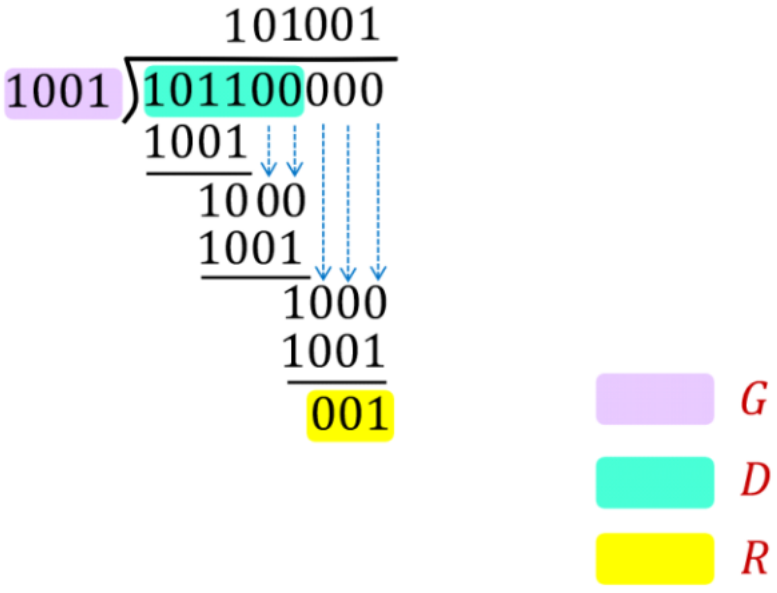
- Easy to implement on hardware
- Powerful error-detection coding that is widely used in practice (e.g., Ethernet, Wi-Fi)
  - Can detect *all odd number* of single bit errors
  - CRC of r bits can detect
    - all burst errors of less than $r + 1$ bits
    - all burst errors of greater than r bits with probability $1 - 0.5^r$
- CRC is also known as Polynomial code
  - A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from x^{k-1}

    E.g. 110001

    $\Rightarrow 1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0 = x^5 + x^4 + 1$

Cyclic Redundancy Check (CRC)

- E.g. $D = 101100$, $r = 3$ and $G = 1001$



Summary

- Checksum (used in TCP/UDP/IP)
- Parity Checking
  - Single bit
  - 2-Dimensional
    - Capable of error correction
- Cyclic redundancy Check (CRC)
  - Commonly used in link layer
  - Efficient
  - effective

Motivation (revisited)

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Link Access Control

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

- 6.3.1 Channel Partitioning Protocols
- 6.3.2 Random Access Protocols
- 6.3.3 Taking-Turns Protocols

6.4 Switched Local Area Networks

Two Types of Network Links

- **Type 1: point-to-point link**
  - A sender and a receiver connected by a dedicated link

A host connects to router
through a dedicated link

  - Example protocols: Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP)
    - No need for multiple access control

Two Types of Network Links

- **Type 2: broadcast link** (shared medium)
  - Multiple nodes connected to a shared broadcast channel.
  - When a node transmits a frame, the channel broadcasts the frame and every other node receives a copy.

802.11 Wi-Fi

Satellite

Ethernet with bus topology

Multiple Access Protocols

- In a broadcast channel, if two or more nodes transmit simultaneously
  - *collision* if node receives two or more signals at the same time.

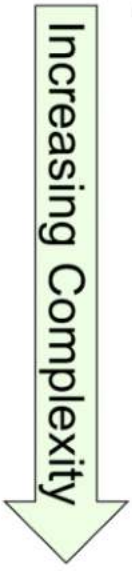
humans at a cocktail party (shared air)

Multiple Access Protocols: Motivation

- The central questions in a conversation carried in a group are *who, when* and how *long* one gets to talk.
- Desired Conversational Characteristics: *etiquettes*
  - Give everyone a chance to speak.
  - Don't speak until you are spoken to.
  - Don't monopolize the conversation.
  - Raise your hand if you have a question.
  - Don't interrupt when someone is speaking.
  - Don't fall asleep when someone is talking.

Multiple Access Protocols: Motivation

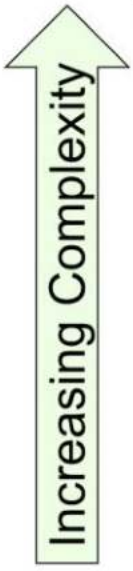
- Human Conversation Protocols can be categorized into three broad classes:



  - **Random Access**
    - No coordination, collisions are possible.
    - "recover" from collisions.
    - E.g. Most of our conversations
  - **"Taking turns"**
    - Each person take turns to talk.
    - E.g. Question answer sessions in seminars
  - **Channel partitioning**
    - divide channel into fixed, smaller "pieces" (e.g., time slots, subject).
    - allocate piece to a person for exclusive use.
    - E.g. U.S. Presidential debates (Mostly)

Multiple Access Protocols: Motivation

- Multiple access protocols can be categorized into three broad classes :



  - **Random Access**
    - channel is not divided, collisions are possible.
    - "recover" from collisions.
  - **"Taking turns"**
    - Each node take turns to transmit.
  - **Channel partitioning**
    - divide channel into fixed, smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.

An ideal multiple access protocol

Given: Broadcast channel of rate R bps

Desired Properties:

1. Collision Free
2. Efficient: when only one node wants to transmit, it can send at rate R.
3. Fairness: when M nodes want to transmit, each can send at average rate R/M
4. fully decentralized:
   - no special node to coordinate transmissions

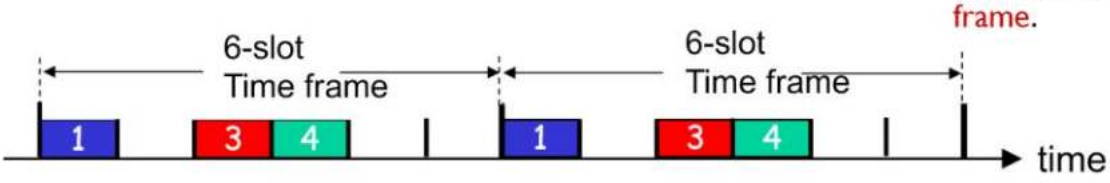
Mandatory Requirement: coordination about channel sharing must use channel itself!: no out-of-band channel signaling

Channel Partitioning Protocols: TDMA

- TDMA (time division multiple access)
  - Access to channel in "rounds".
    - Similar to US presidential debates
  - Each node gets fixed length time slots in each round.
    - Length of time slot = data frame transmission time
  - Example: 6 nodes sharing a link
    - Nodes 1, 3, 4 have data to send
    - slots 2, 5, 6 are idle.

Jargon Alert:

- Frame: Unfortunately, in TDMA, the collection of N time slots is called a Frame. We will disambiguate this by calling a frame as either data frame or time frame.



Channel Partitioning Protocols: TDMA

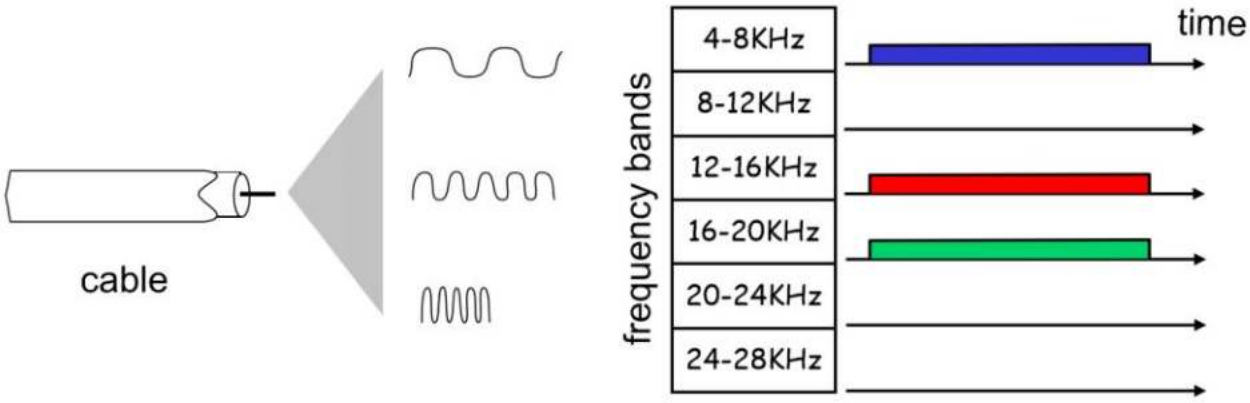
- Collision Free: Yes
- Efficiency
  - Inefficient
  - Unused slots go idle.
  - The maximum throughput for a node is R/N
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Channel Partitioning Protocols: FDMA

- FDMA (frequency division multiple access)
  - Channel spectrum is divided into frequency bands.
  - Each node is assigned a fixed frequency band.
  - Unused transmission time in frequency bands go idle.
  - Example: 6 nodes, 1, 3, 4 have frames, frequency bands 2, 5, 6 are idle.



Channel Partitioning Protocols: FDMA

- Collision Free: Yes
- Efficiency
  - Inefficient
  - Unused slots go idle.
  - The maximum throughput for a node is R/N
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

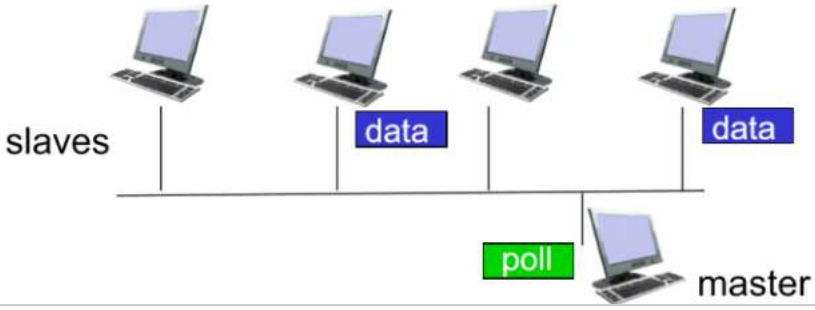
1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Multiple Access Protocols

- Multiple access protocols can be categorized into three broad classes:
  - Channel partitioning
    - divide channel into fixed, smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.
  - **"Taking turns"**
    - nodes take turns to transmit.
  - Random Access
    - channel is not divided, collisions are possible.
    - "recover" from collisions.

"Taking Turns" Protocols: Polling

- The polling protocol requires one of the nodes to be designated as a *master* node.
- The master node *polls* each of the nodes in a *round-robin* fashion.
  - master informs node 1, it can transmit up to some maximum number of frames.
  - After node 1 transmits some frames, the master node tells node 2 it (node 2) can transmit up to the maximum number of frames.
  - The procedure continues in this manner



"Taking Turns" Protocols: Polling

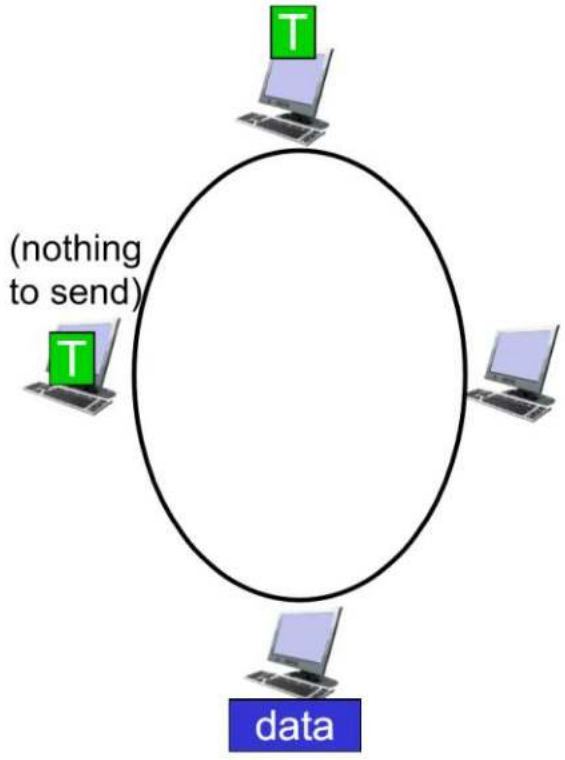
- Collision Free: Yes
- Efficiency
  - Higher efficiency.
  - Overhead of polling.
- Fairness: Perfectly Fair
- Decentralized:
  - No
  - Master node is a single point of Failure

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

"Taking Turns" Protocols: Token Passing

- Special frame, *token*, is passed from one node to next, sequentially.
- When a node receives a token
  - hold onto the token only if some frames to transmit
    - it sends up to a maximum number of frames and then forwards the token to the next node.
  - otherwise, forward the token to the next node.



"Taking Turns" Protocols: Token Passing

- Collision Free: Yes
- Efficiency
  - Higher efficiency.
  - Overhead of token passing
- Fairness: Perfectly Fair
- Decentralized: Yes
- *Downside*
  - Token loss can be disruptive
    - data frame loss
    - System bugs
  - Node failure can break the ring

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Multiple Access Protocols

- Multiple access protocols can be categorized into three broad classes:
  - Channel partitioning
    - divide channel into smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.
  - "Taking turns"
    - nodes take turns to transmit.
  - **Random Access**
    - channel is not divided, collisions are possible.
    - "recover" from collisions.

Random Access Protocols

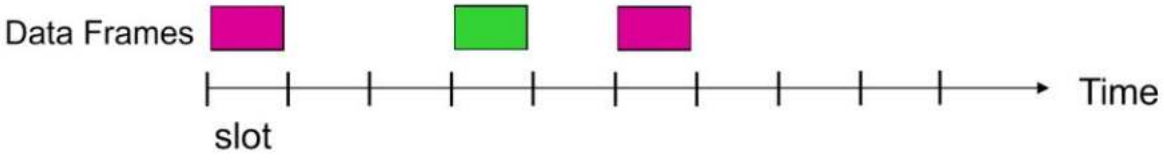
- When node has data to send
  - transmit at full channel data rate *R*.
  - no *a priori* coordination among nodes
- Two or more transmitting nodes ➜ "collision"
- Random access protocols specify:
  - how to *detect* collisions
  - how to *recover* from collisions
- We will explore various protocols
  - Slotted ALOHA, ALOHA
  - CSMA, CSMA/CD

Slotted ALOHA

- When node has data to send
  - transmit at full channel data rate *R*.
  - no *a priori* coordination among nodes

Design:

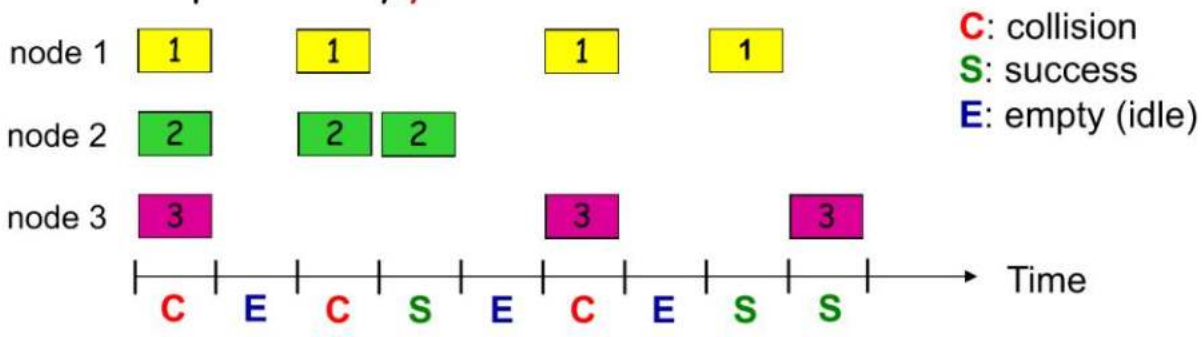
- All frames are of equal size, *L* bits.
- Time is divided into slots of equal length
  - length = time to transmit 1 frame = *L/R*
- Nodes start to transmit only at the beginning of a slot.
  - Time is synchronized at each node.



Slotted ALOHA

Operation:

- When the node has a fresh frame to send
  - wait until the beginning of the next slot and transmits the entire frame in the slot.
  - If *no collision*: data transmission is a success.
  - If *collision*: data transmission is a failure.
    - retransmit the frame in each subsequent slot with probability *p* until success.



Slotted ALOHA

- Collision Free: No
- Efficiency
  - Yes, when only one node is active, it gets a throughput of R
  - No, when there are many active nodes the maximum efficiency is only *37%*
    - Slots are wasted due to both *collision* and because of being *empty*
    - 100 Mbps system will give only 37 Mbps
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

A Little Side Note

- Q: Why is it called ALOHA?
- A: The **ALOHAnet**, also known as the ALOHA System, or simply ALOHA, was a pioneering computer networking system developed – maybe you can guess it – at the University of Hawaii.
- Norman Abramson was the leader of the team.
- The idea was to use a radio network to connect Oahu and the other Hawaiian islands together. ALOHA made use of one, shared, inbound channel, and thus requiring a novel ***multiple access protocol***.

Pure (unslotted) ALOHA

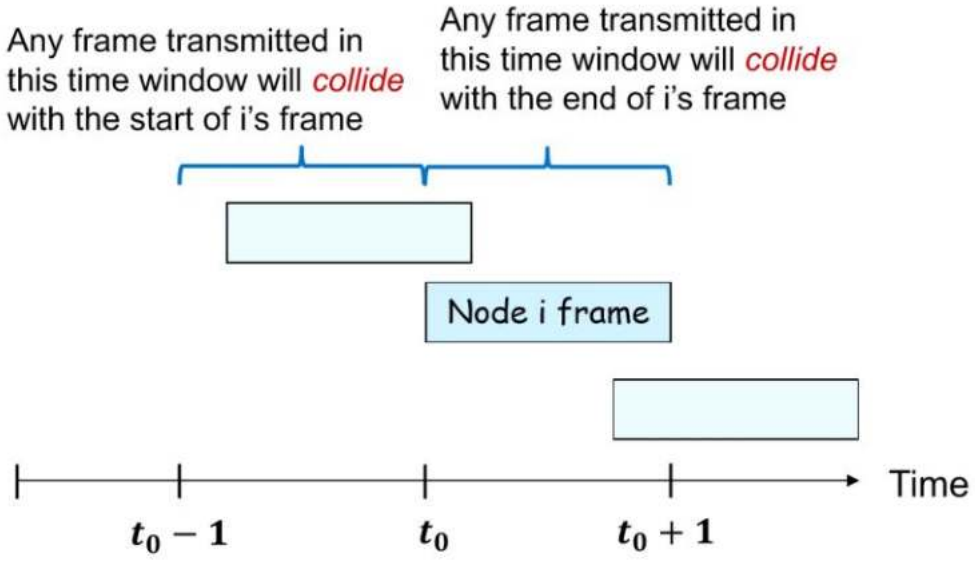
- Even simpler than Slotted ALOHA
  - *No* time slots
  - *No* Synchronization

Operation:

- When the node has a fresh frame to send
  - Transmits the entire frame *immediately*.
  - If *no collision*: data transmission is a success.
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.

Pure (unslotted) ALOHA

- Chance of collision increases:
  - frame sent at t_0 collides with other frames sent in $(t_0 - 1, t_0 + 1)$



Pure (unslotted) ALOHA

- Collision Free: No
- Efficiency
  - Yes, when only one node is active, it gets a throughput of R
  - No, when there are many active nodes the maximum efficiency is only *18%*
    - Slots are wasted due to both *collision* and because of being *empty*
    - 100 Mbps system will give only 18 Mbps
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Carrier Sense Multiple Access

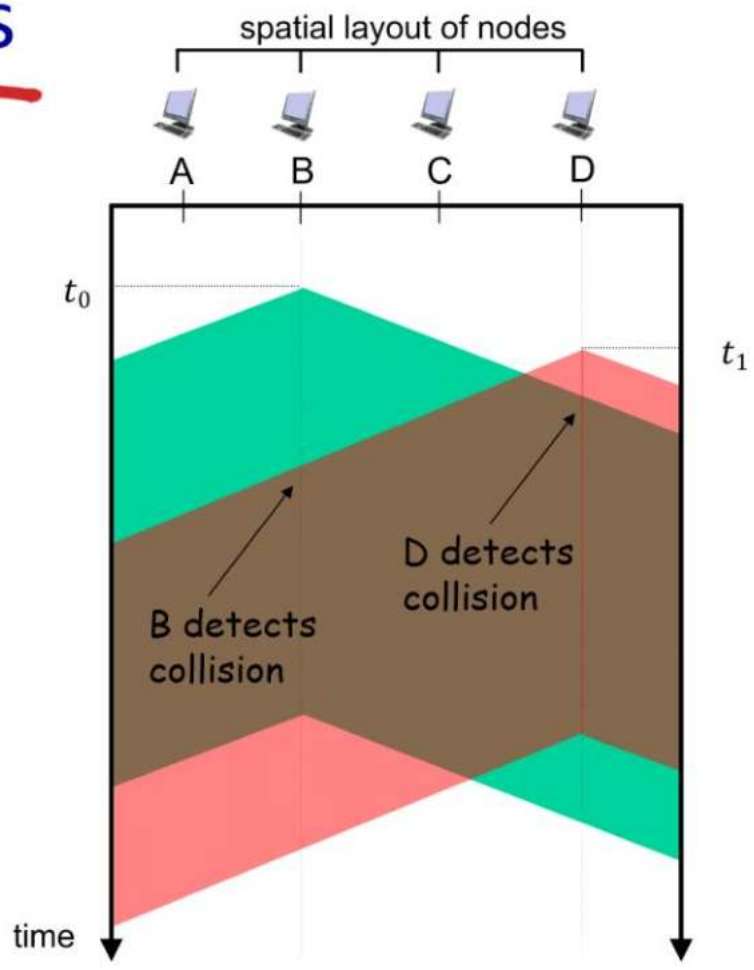
- One major *design flaw* in ALOHA
  - a node's decision to transmit is made *independently* of the activity of the other nodes attached to the broadcast channel.
  - a node *pays no attention* to whether another node happens to be transmitting when it begins to transmit
- Human analogy
  - Listen before you speak

CSMA: listen before transmit

- if channel *sensed idle*: transmit entire frame
- if channel *sensed busy*: defer transmission

CSMA Collisions

- Collisions can still occur:
  - *propagation delay* means two nodes may not hear each other's transmission immediately.



CSMA/CD (Collision Detection)

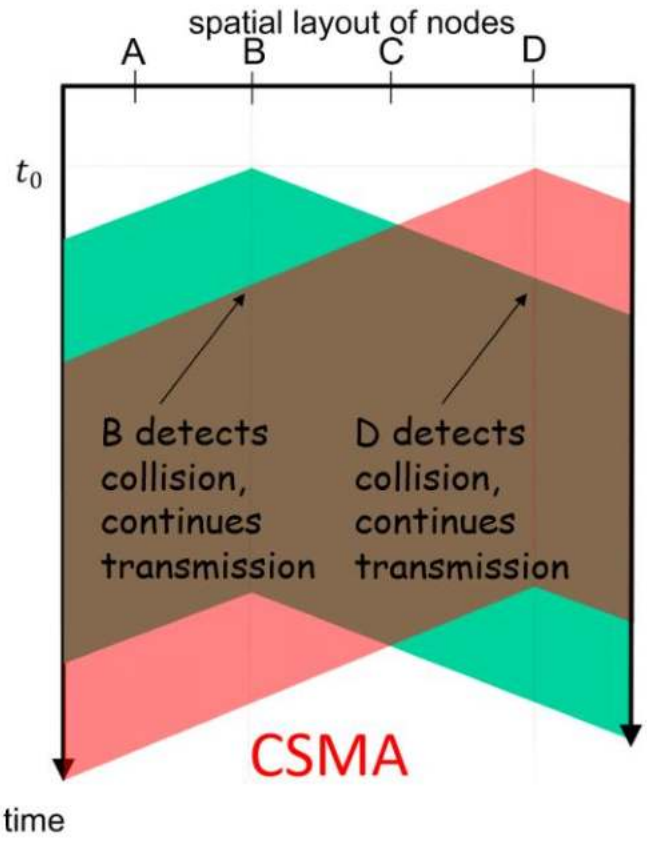
- One major *design flaw* in ALOHA and CSMA
  - a node *does not stop* transmitting even when collision is detected
- Human analogy
  - If someone else begins talking at the same time, stop talking

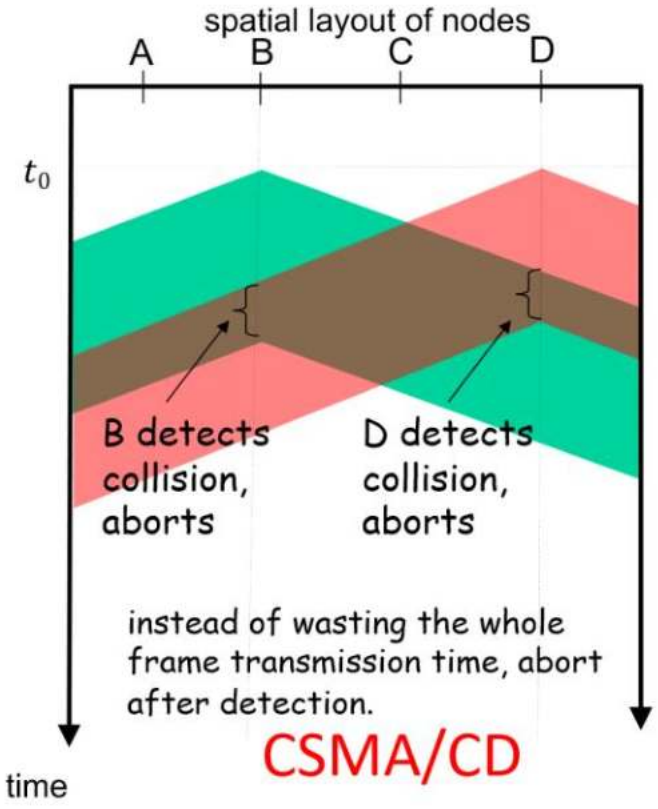
Bob Metcalfe

CSMA/CD:

- if channel *sensed idle*: transmit entire frame
- if channel *sensed busy*: defer transmission
- If *collision detected*: Abort transmission
  - Retransmit after a random delay

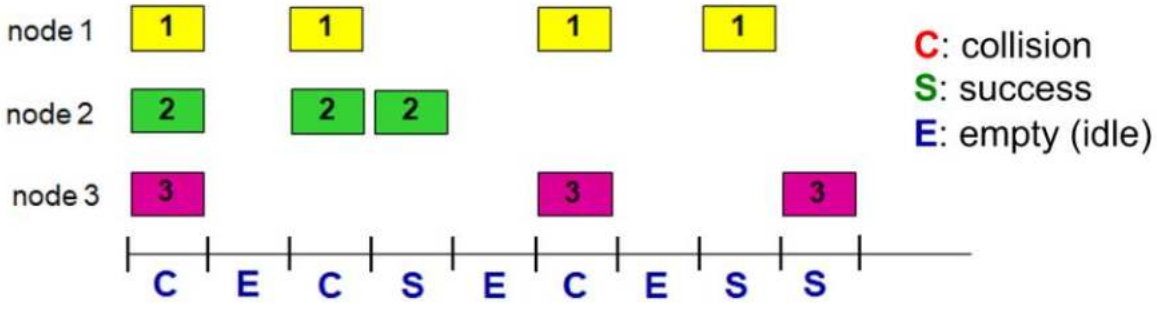
CSMA/CD (Collision Detection)





CSMA/CD Backoff Algorithm

- If *collision detected*: Abort transmission
  - Retransmit after a random delay
- *Motivation:* ALOHA
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.



CSMA/CD Backoff Algorithm

- If *collision detected*: Abort transmission
  - Retransmit after a random delay
- *Motivation:* ALOHA
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.
  - Major Drawback:
    - The probability of collision in all subsequent time slots remain the same
      - It can even increase if a new node starts transmitting
- *Goal*: adapt retransmission attempts to estimated current load
  - More collisions implies heavier load.
  - longer back-off interval with more collisions.

CSMA/CD Backoff Algorithm

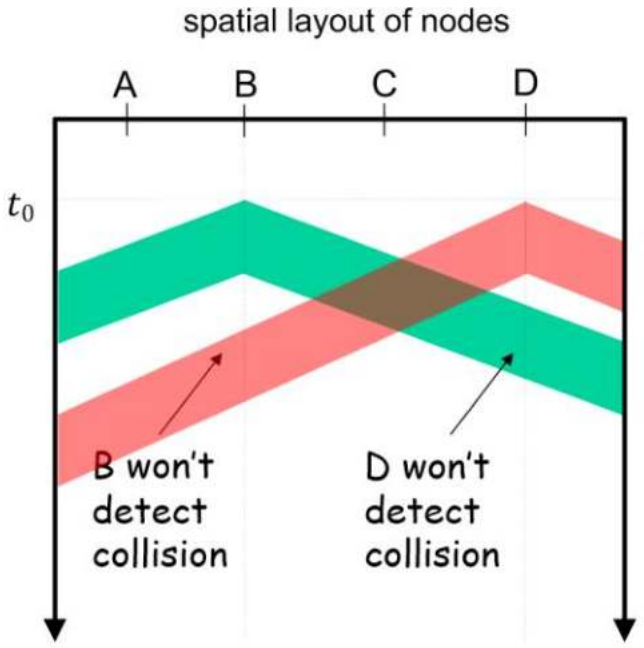
For Ethernet 1 time unit is set as 512 bit transmission times

Binary Exponential backoff:

- After 1st collision: $p = 1/2$
  - choose K at random from {0, 1};
  - wait K time units before retransmission.
- After 2nd collision: $p = 1/4$
  - choose K from {0, 1,2, 2^2-1}.
  - wait K time units before retransmission.
- After m^{th} collision $p = 1/2^m$
  - choose K at random from {0, 1, …, 2^m-1}
- *Property*: retransmission attempts to estimates current load
  - More collisions implies heavier load.
  - longer back-off interval with more collisions.

Minimum Frame Size

- What if the frame size is too small?
  - Collision happens but may not be detected by sending nodes.
    - No retransmission!
- For example, Ethernet requires a minimum frame size of 64 bytes.



CSMA & CSMA/CD

- Collision Free: NO
- Efficiency: Yes
- Fairness: Yes
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Summary

- Channel partitioning
  - Divide channel by time, used in GSM
  - Divide channel by frequency, commonly used in radio, satellite systems
- Taking turns
  - polling from central site, used in Bluetooth
  - token passing, used in FDDI and token ring
- Random access
  - ALOHA wireless packet switched network.
  - CSMA/CD used in Ethernet

Link layer 2

Friday, November 3, 2023 3:09 PM

Lec_9_Link
_Layer_II

CS2105

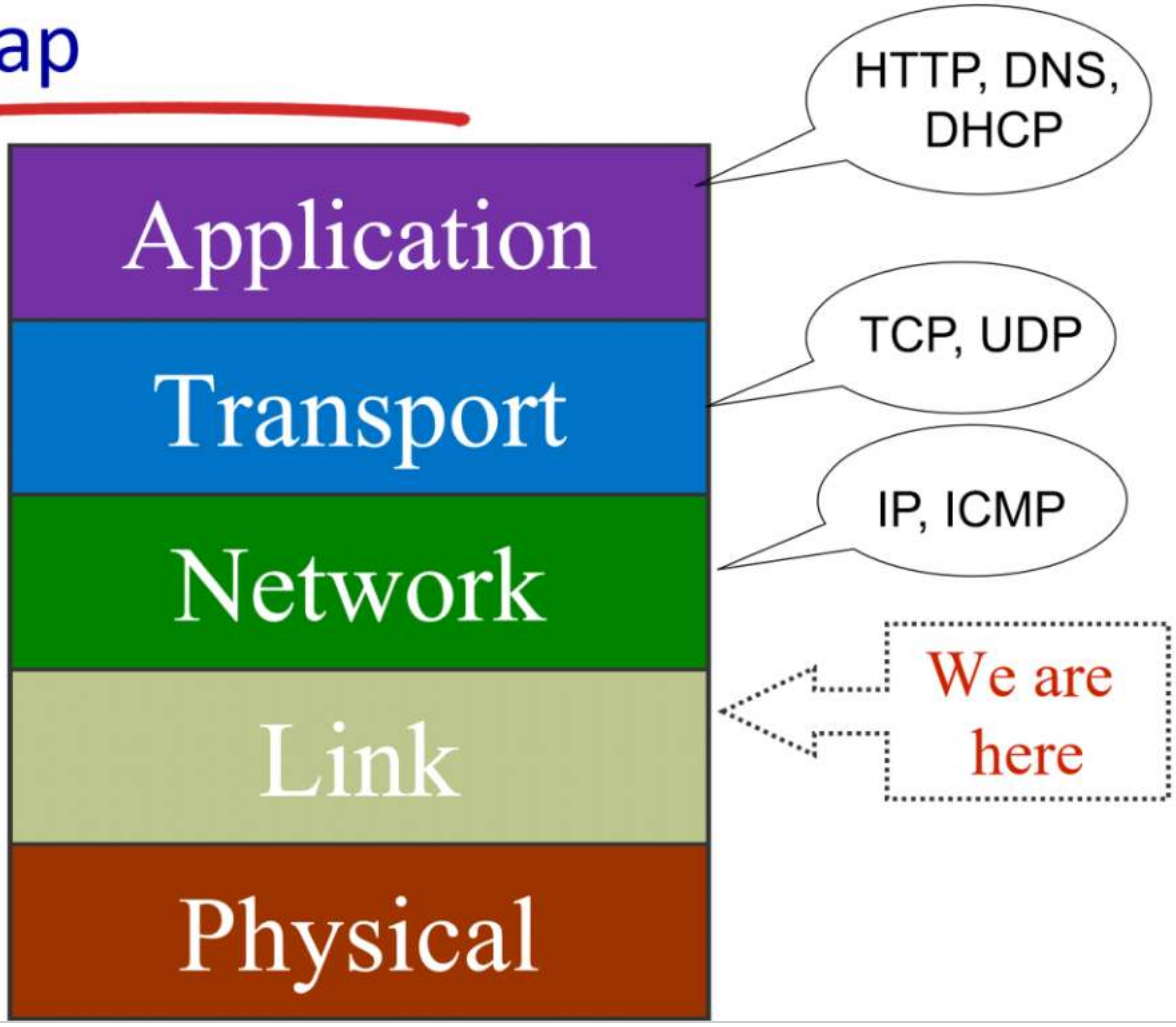
An *Awesome* Introduction to Computer Networks

The Link Layer, LAN

Department of Computer Science
School of Computing

Adapted from Slides by Prof Roger
And

Recap


HTTP, DNS, DHCP
Application
TCP, UDP
Transport
IP, ICMP
Network
We are here
Link
Physical

The Link Layer

After this set of lectures, we will understand:

- the role of link layer and the services it could provide.
- how parity and CRC scheme work.
- different methods for accessing shared medium.
- the role of switches in interconnecting subnets in a LAN.
- how ARP allows a host to discover the MAC addresses of other nodes in the same subnet.

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

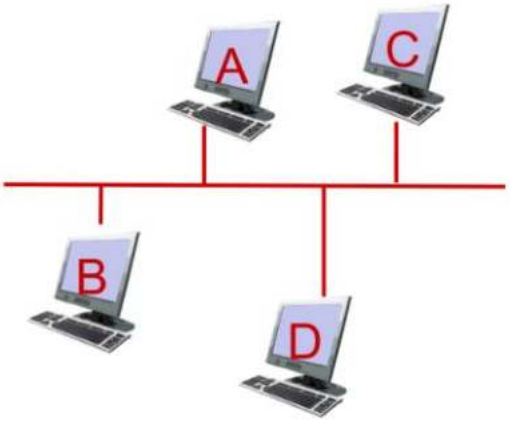
6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

Motivation

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*



Detection

Link Access Control

Framing

MAC Address

Jargon Alert:
- MAC: *Media Access Control*

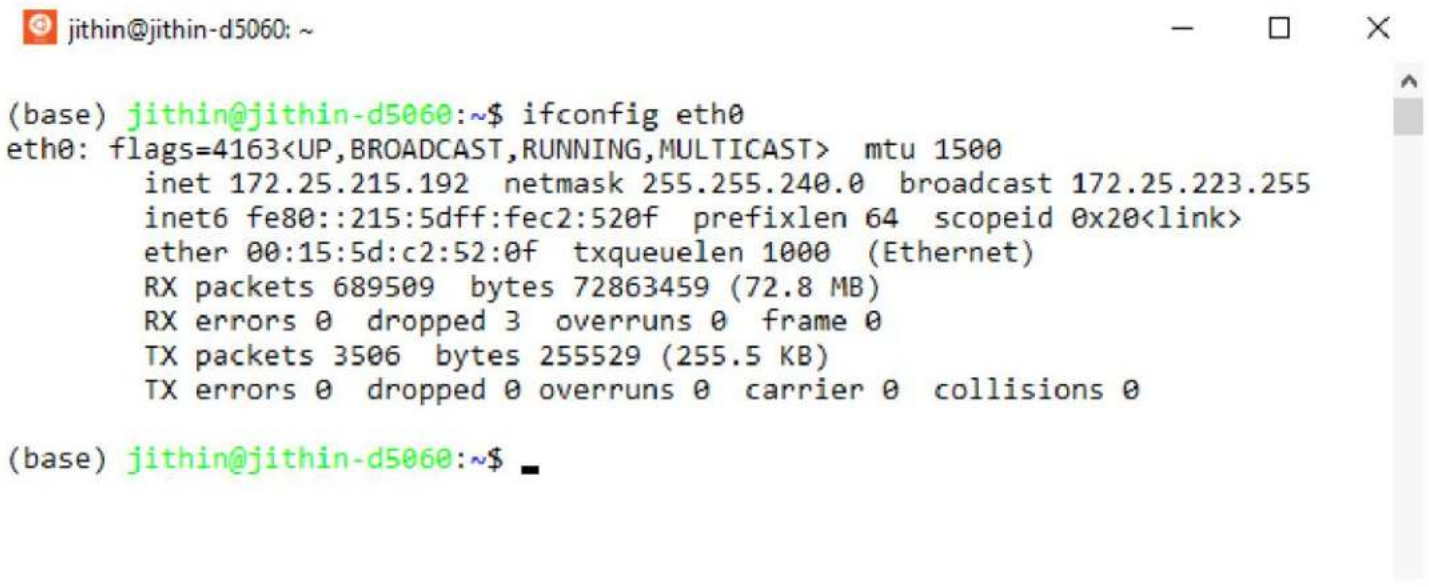
- Every adapter (NIC) has a MAC address (aka physical or LAN address).
  - Used to send and receive link layer frames.
  - When an adapter receives a frame, it checks if the destination MAC address of the frame matches its own MAC address.
    - If yes, adapter extracts the enclosed datagram and passes it to the protocol stack.
    - If no, adapter simply discards the frame without interrupting the host.

MAC Address

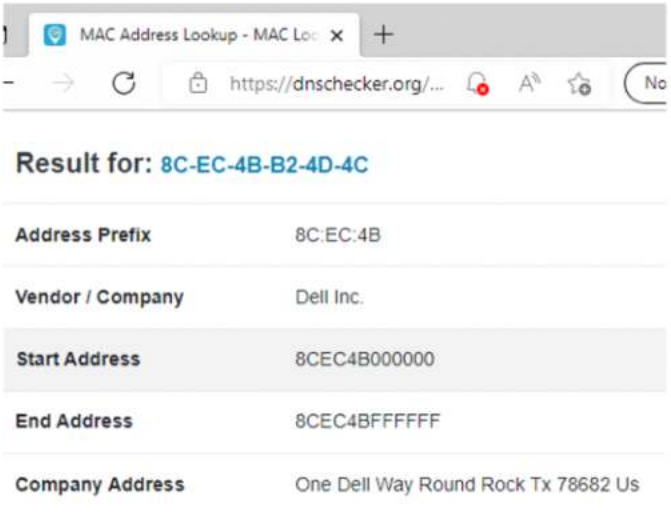
Jargon Alert:
- NIC: *Network Interface Card*
- ROM: *Read-only Memory*

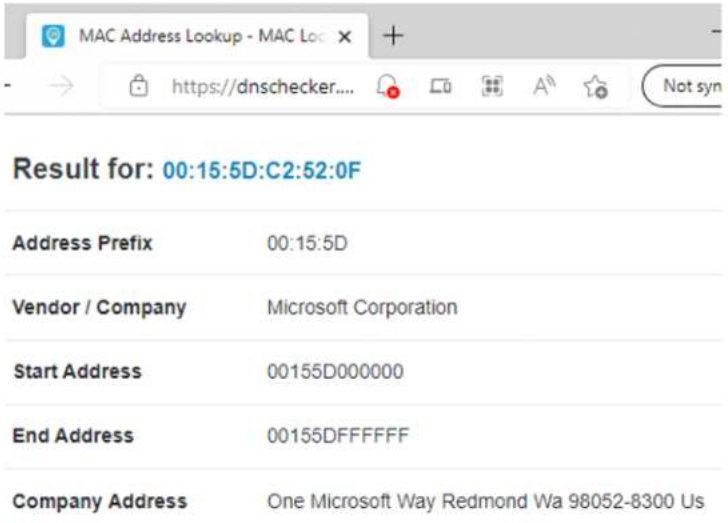
- MAC address is typically 48 bits, burned in NIC ROM (sometimes software settable).
  - Example: `5C-F9-DD-E8-E3-D2` — hexadecimal (base 16) notation
    - `0101 1100 1111 1001 1101 1101 1110 1000 1110 0011 1101 0010`
  - MAC address allocation is administered by IEEE.
    - The first three bytes identifies the vendor of an adapter.
  - Broadcast Address: `FF-FF-FF-FF-FF-FF`

MAC Address


jithin@jithin-d5060: ~
(base) jithin@jithin-d5060:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.25.215.192 netmask 255.255.240.0 broadcast 172.25.223.255
inet6 fe80::215:5dff:fec2:520f prefixlen 64 scopeid 0x20<link>
ether 00:15:5d:c2:52:0f txqueuelen 1000 (Ethernet)
RX packets 689509 bytes 72863459 (72.8 MB)
RX errors 0 dropped 3 overruns 0 frame 0
TX packets 3506 bytes 255529 (255.5 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(base) jithin@jithin-d5060:~$

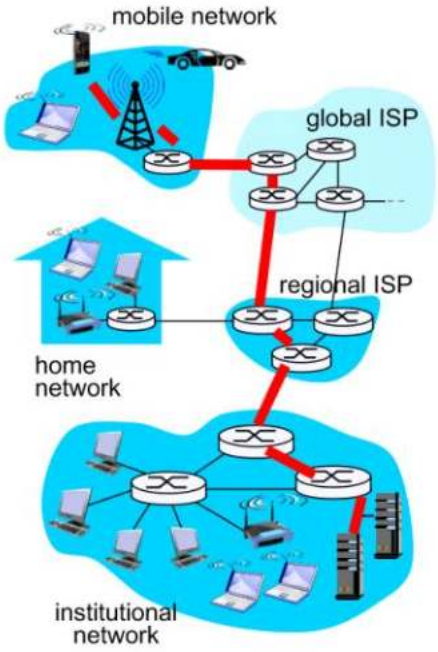
MAC Address


MAC Address Lookup - MAC Lo
https://dnschecker.org/...
No
Result for: 8C-EC-4B-B2-4D-4C
Address Prefix 8C:EC:4B
Vendor / Company Dell Inc.
Start Address 8CEC4B000000
End Address 8CEC4BFFFFFF
Company Address One Dell Way Round Rock Tx 78682 Us


MAC Address Lookup - MAC Lo
https://dnschecker....
Not syn
Result for: 00:15:5D:C2:52:0F
Address Prefix 00:15:5D
Vendor / Company Microsoft Corporation
Start Address 00155D000000
End Address 00155DFFFFFF
Company Address One Microsoft Way Redmond Wa 98052-8300 Us

Link Layer: Introduction

- *Link layer* sends datagram between adjacent nodes (hosts or routers) over a single link.
  - IP datagrams are encapsulated in link-layer frames for transmission.
  - Different link-layer protocols may be used on different links.
    - each protocol may provide a different set of services.



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

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

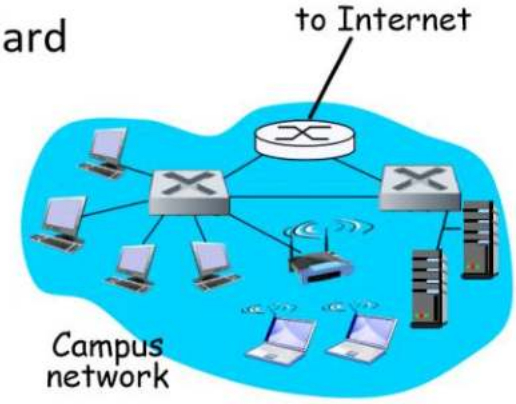
6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

Local Area Network (LAN)

- LAN is a computer network that interconnects computers within a *geographical area* such as office building or university campus.

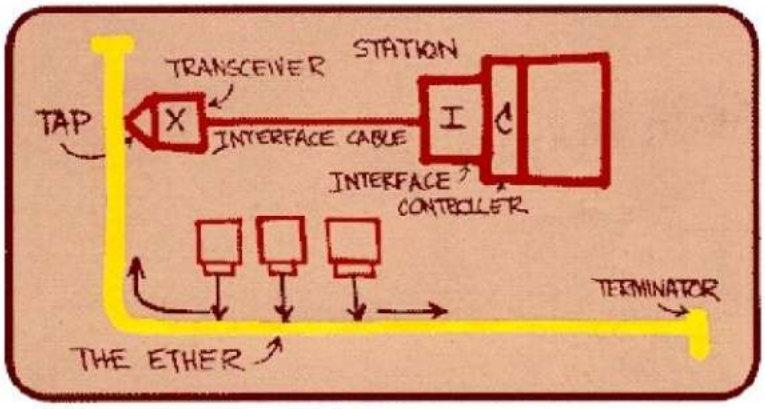
- LAN technologies:
  - IBM Token Ring: IEEE 802.5 standard
  - Ethernet: IEEE 802.3 standard
  - Wi-Fi: IEEE 802.11 standard
  - Others



Ethernet

- "*Dominant*" wired LAN technology:
  - Developed in mid 1970s
  - Standardized by Xerox, DEC, and Intel in 1978
  - Simpler and cheaper than token ring and ATM

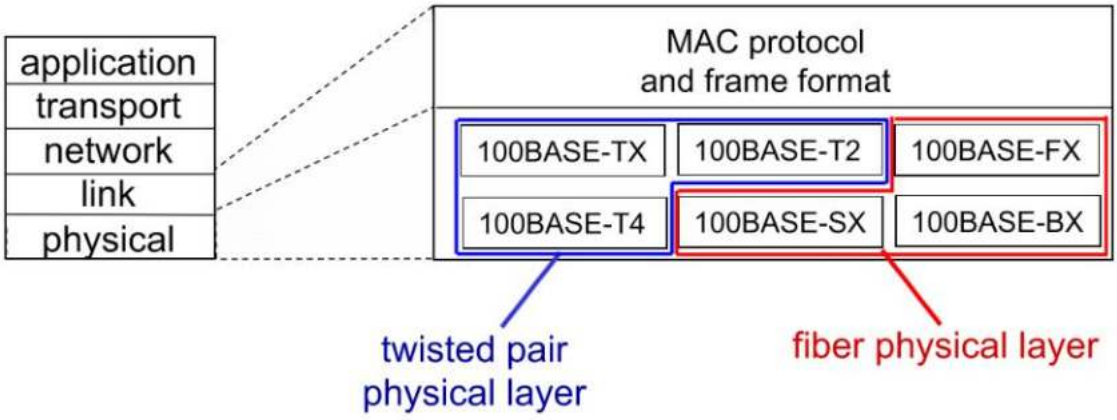
Ethernet connection
(Source: Wikipedia)



Metcalfe's Ethernet sketch

802.3 Ethernet Standards

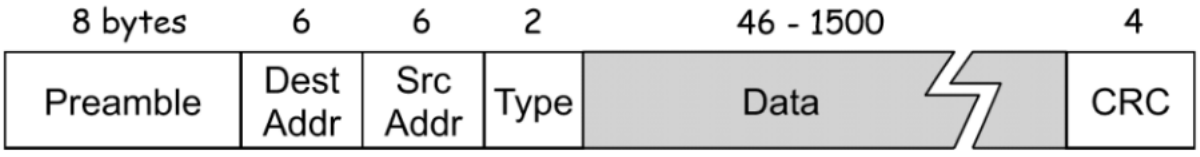
- A series of Ethernet standards have been developed over the years.
  - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps
  - Different physical layer media: cable, fiber optics
  - MAC protocol and frame format remain unchanged



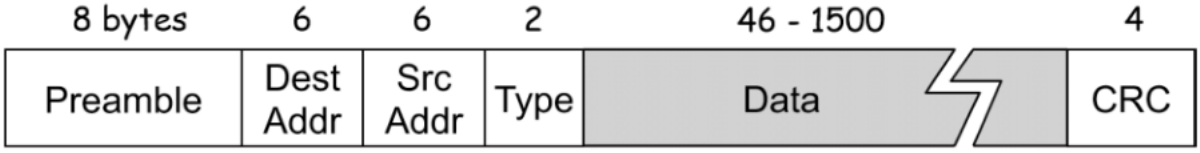
Ethernet Frame Structure

- Let us consider the case of sending an IP datagram from one host to another, on the same Ethernet LAN

- Sending NIC (adapter) encapsulates IP datagram in Ethernet frame.



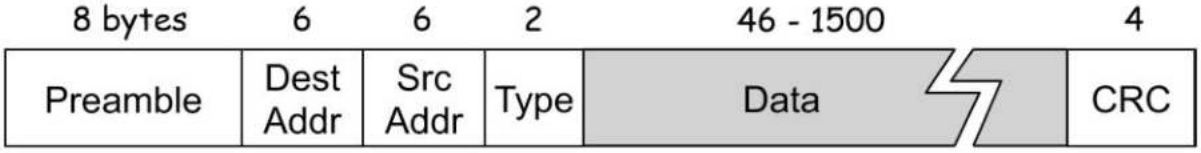
Ethernet Frame Structure



- *Source and dest MAC address:*
  - If NIC receives a frame with matching destination address, or with broadcast address
    - it passes data in the frame to network layer protocol.
  - Otherwise
    - NIC discards frame.

Ethernet Frame Structure

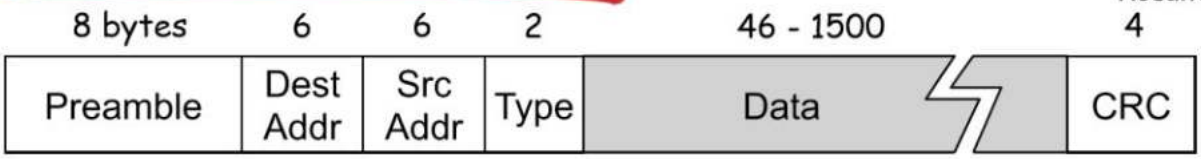
Jargon Alert:
- MTU: *Maximum Transmission Unit*



- *Data:*
  - The maximum size is 1500 bytes.
    - This maximum size is the link MTU which we mentioned when we discussed IP fragmentation.
  - The minimum size is 46 bytes
    - The minimum size is to ensure that a collision will always be detected.

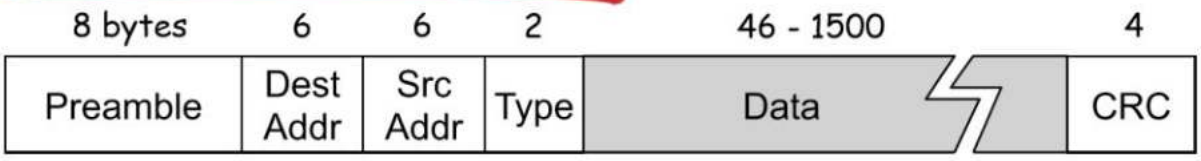
Ethernet Frame Structure

Jargon Alert:
- CRC: *Cyclic Redundancy Check*



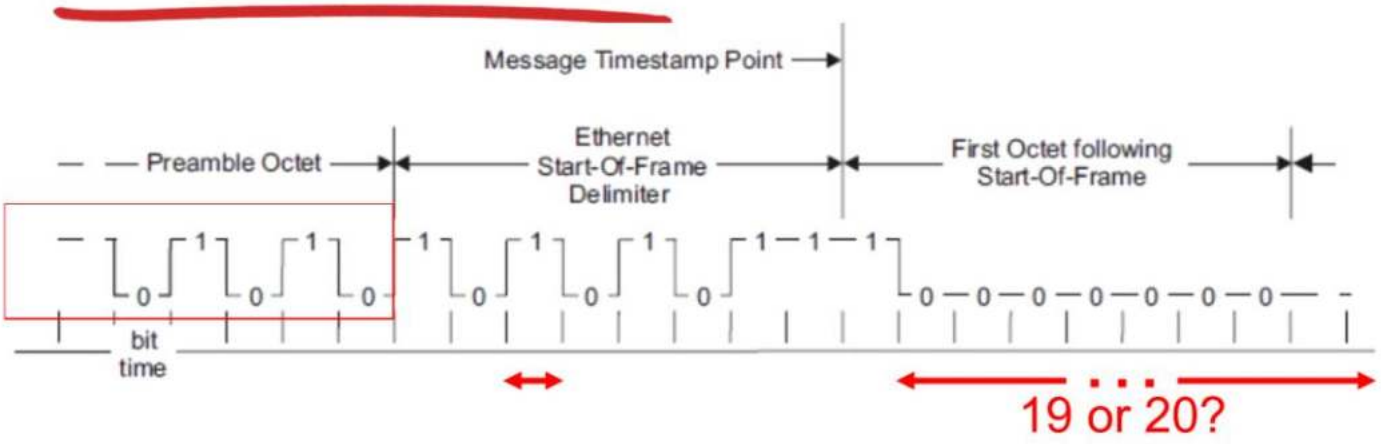
- *CRC:* corrupted frame will be dropped.
- *Type:* Indicates higher layer protocol
  - To understand this, we need to keep in mind that hosts can use other network-layer protocols besides IP.
    - E.g. Novell IPX, AppleTalk, ARP, etc.
  - The type field permits Ethernet to multiplex network-layer protocols.
  - Type field is analogous to
    - the protocol field in the network-layer datagram and
    - the port-number fields in the transport-layer segment

Ethernet Frame Structure



- *Preamble:*
  - 7 bytes with pattern **10101010** (AA_{Hex})
  - Followed by 1 byte with pattern **10101011** (AB_{Hex}).
    - Also called "*start of frame*"
  - used to synchronize receiver and sender clock rates.

Ethernet Frame Structure



- The preamble provides a "square wave" pattern that tells the receiver the sender's clock rate
  - it tells the receiver the width of a bit
  - which is important if there is a long string of bits of the same value, e.g., 19 or 20 zeros.

Motivation (revisited)

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

Link Access Control

Reliability

Detection

Ethernet Data Delivery Service

- *Unreliable*: receiving NIC doesn't send ACK or NAK to sending NIC.
  - data in dropped frames will be recovered only if initial sender uses higher layer rdt (e.g. TCP); otherwise dropped data is lost.
- Ethernet's multiple access protocol:
  - *CSMA/CD* with binary (exponential) backoff.

Motivation (revisited)

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

Link Access Control

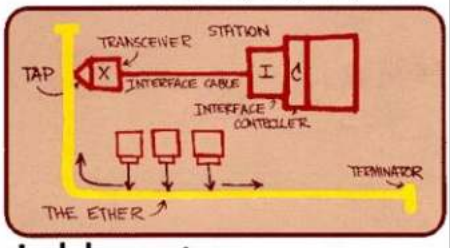
Reliability

Detection

Ethernet: Physical Topology

- How do we interconnect the nodes to create this shared link?

Ethernet: Bus Topology



Bus topology: popular till mid 90s

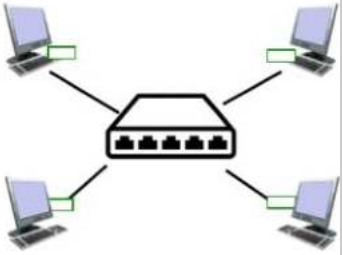
- The *original* Ethernet LAN used a coaxial bus to interconnect the nodes.
- Is a *broadcast* LAN
  - All transmitted frames received by all adapters connected to the bus.
    - all nodes can collide with each other
- Drawbacks
  - Back bone cable
    - If damaged, the entire network will fail
  - Difficult to troubleshoot problems
  - Very slow and not ideal for larger networks
    - Due to collisions

Ethernet with bus topology

Ethernet: Star Topology

Star topology: prevalent today

- Hub
  - Popular in late 1990's
  - nodes are directly connected to a hub
  - A hub is a *physical-layer* device that acts on individual bits rather than frames.
    - When a bit arrives from one interface,
      - the hub simply re-creates the bit
      - boosts its energy strength, and
      - transmits the bit onto all the other interfaces.

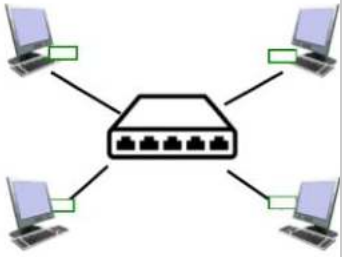


Ethernet with star topology

Ethernet: Star Topology

Star topology: prevalent today

- *Hub*
  - Advantages
    - Cheap
    - Easy Maintenance
      - Modular design of the network
  - Drawbacks
    - Very slow and not ideal for larger networks
      - Due to collisions

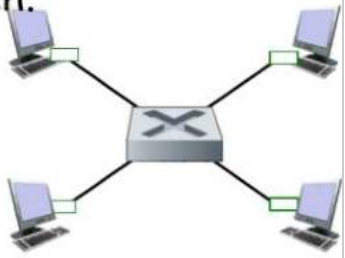


Ethernet with star topology

Ethernet: Star Topology

Star topology: prevalent today

- *Switch*
  - Popular since early 2000's
  - nodes are directly connected to a switch
  - A switch is a *layer-2* device
    - Works acts on *frames* rather than individual bits.
    - *No collisions*
    - A bona-fide *store-and-forward* packet switch.



Ethernet with star topology

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

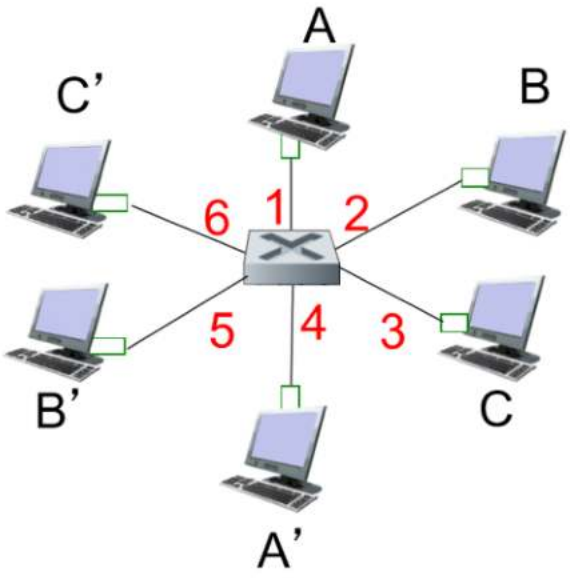
Ethernet Switch

a 50-port Ethernet switch
(Source: Wikipedia)

- A *link-layer device* used in LAN
  - Examine incoming frame's MAC address
    - *selectively forward* frame to one-or-more outgoing links.
  - *Store and forward* Ethernet frames
  - uses *CSMA/CD* to access link
- *Transparent*
  - hosts are unaware of presence of switches
- *Plug-and-play (self-learning)*
  - switches do not need to be configured

Switch: *multiple* simultaneous transmissions

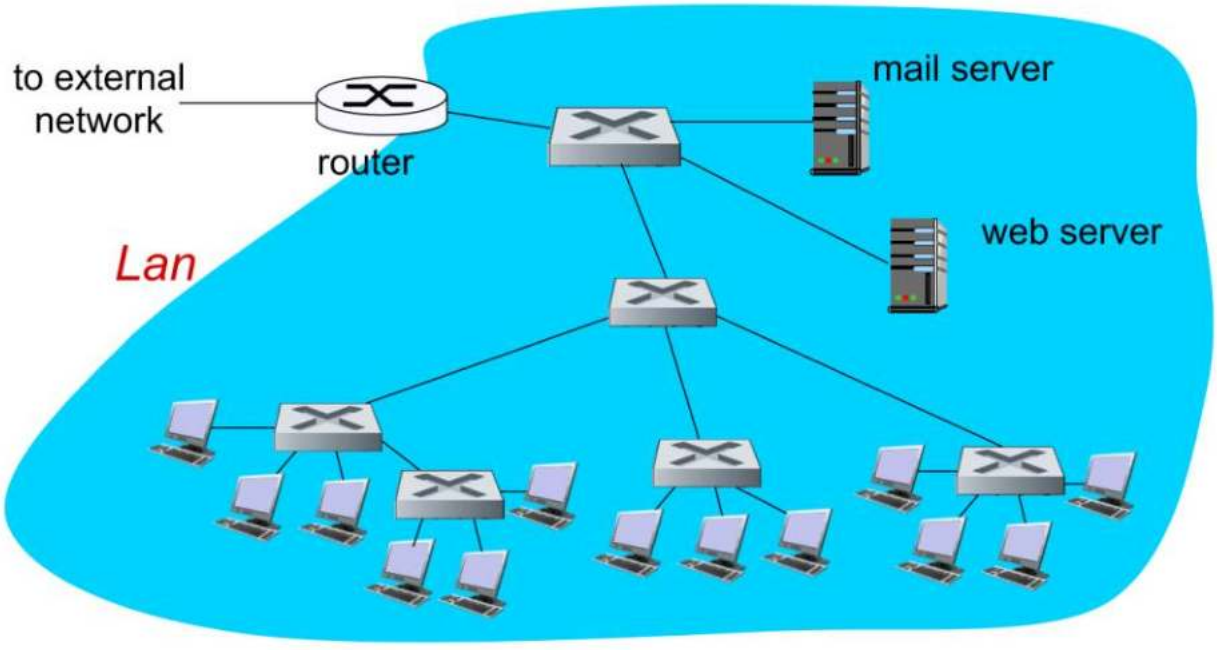
- Nodes have *dedicated*, *direct* connection to switch
- switches *buffer* packets
- Ethernet protocol used on *each* incoming link
  - but *no* collisions!
- *Switching:*
  - A-to-A' and B-to-B' can transmit simultaneously, without collisions



switch with six interfaces (1,2,3,4,5,6)

Interconnecting Switches

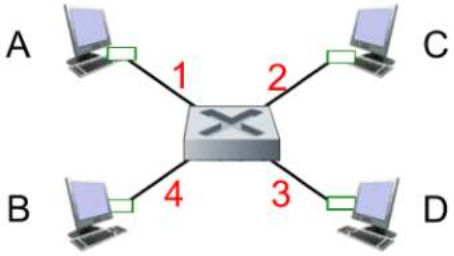
- Switches can be connected in hierarchy.



Interconnecting Switches

Switch Forwarding Table

- Selective Forwarding?
- Q: how does switch know A is reachable via interface 1?
  - A: each switch has a switch table.
    - Format of entry:



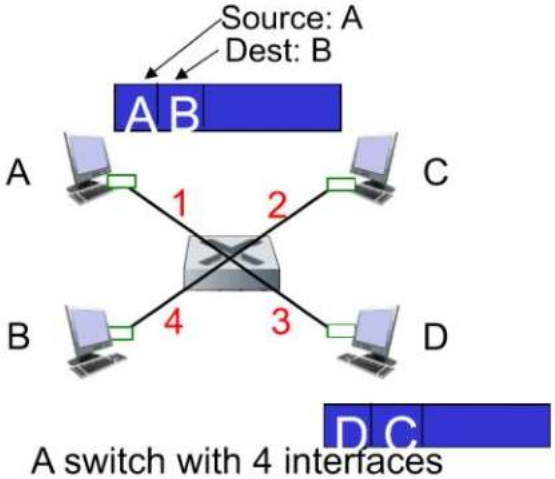
A switch with 4 interfaces (1, 2, 3, 4)

< MAC address of host, interface to reach host, TTL >

    - Looks like a routing table !
- Q: how are entries created and maintained in a switch table?
  - Something like a routing protocol?

Switch: Self-learning

- Switch *learns* which hosts can be reached through which interfaces.
  - when frame received, switch "*learns*" location of sender
  - *records* sender/location pair in switch table
    - When receiving a frame from A, note down the location of A in switch table.



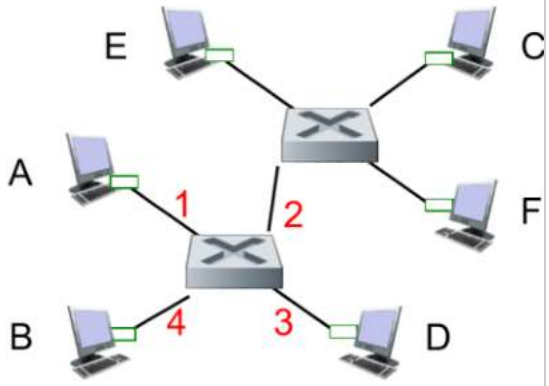
A switch with 4 interfaces (1, 2, 3, 4)

MAC addr	Interface	TTL
A	*1*	*60*
D	*3*	*60*

Switch table (initially empty)

Switch: frame filtering/forwarding

Let us try to *design* the forwarding algorithm



- Received a frame to *A* on interface *4*
  - Forward to interface *1*
- Received a frame to *D* on interface *1*
  - Forward to interface *2,3,4 (all except 1)*
- Received a frame to *F* on interface *2*
  - *Filter* the frame *(drop the frame)*

MAC addr	Interface	TTL
A	1	20
B	4	56
C	2	10
F	2	60

Switch table

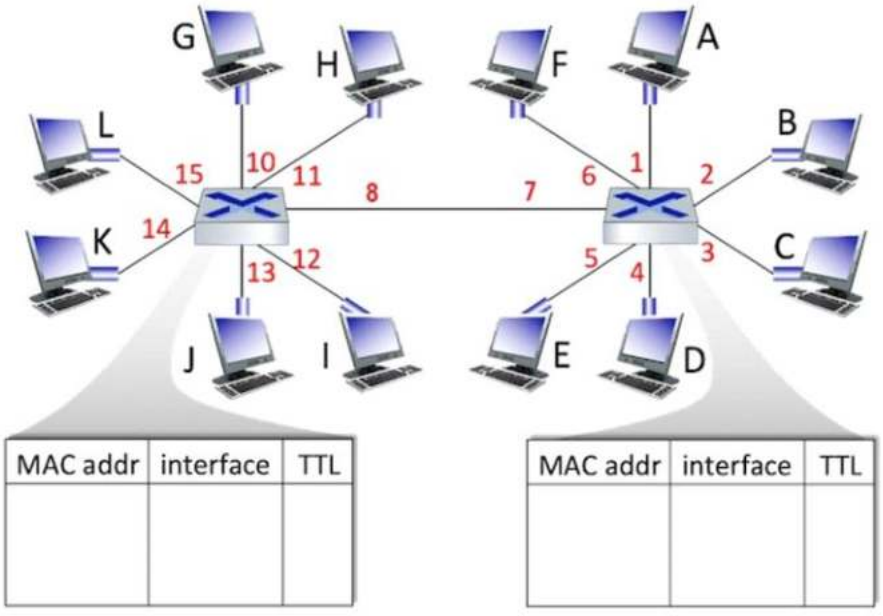
Switch: frame filtering/forwarding

When frame received at switch:

1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. *if* entry found for destination
    1. if destination on segment from which frame arrived
        1. drop frame
    2. else forward frame on interface indicated by entry
4. *else* flood
    1. forward on all interfaces except arriving interface

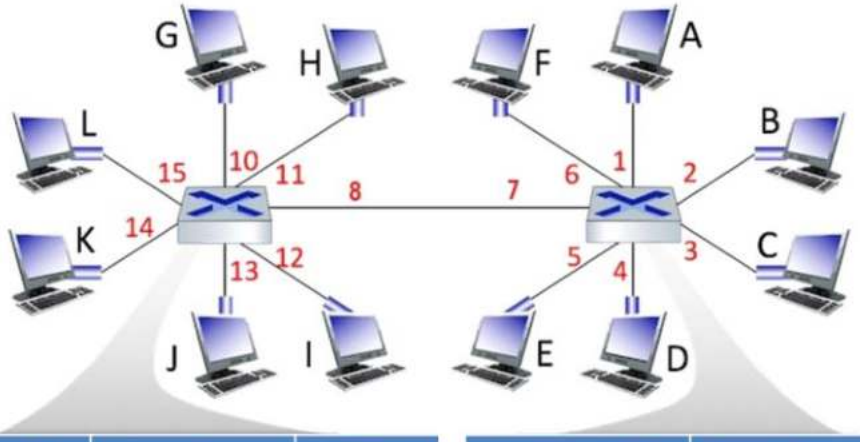
Self-learning multi-switch example

- At t=0 the switch table entries for both switches are empty.
- When a source node sends to a destination node, and the destination replies immediately (well before the next time step).



Self-learning multi-switch example

- At t=1: B ↔ L

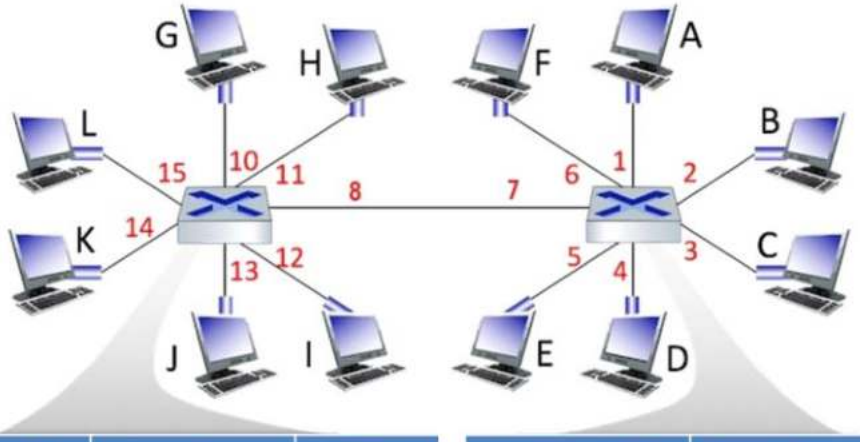


MAC addr	Interface	TTL
B	8	60
L	15	60

MAC addr	Interface	TTL
B	2	60
L	7	60

Self-learning multi-switch example

- At t=2: E ↔ D

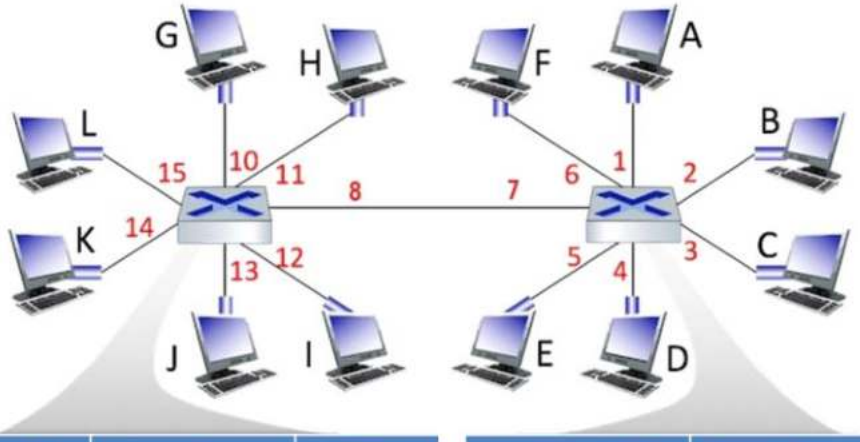


MAC addr	Interface	TTL
B	8	59
L	15	59
E	8	60

MAC addr	Interface	TTL
B	2	59
L	7	59
E	5	60
D	4	60

Self-learning multi-switch example

- At t=3: B ↔ L

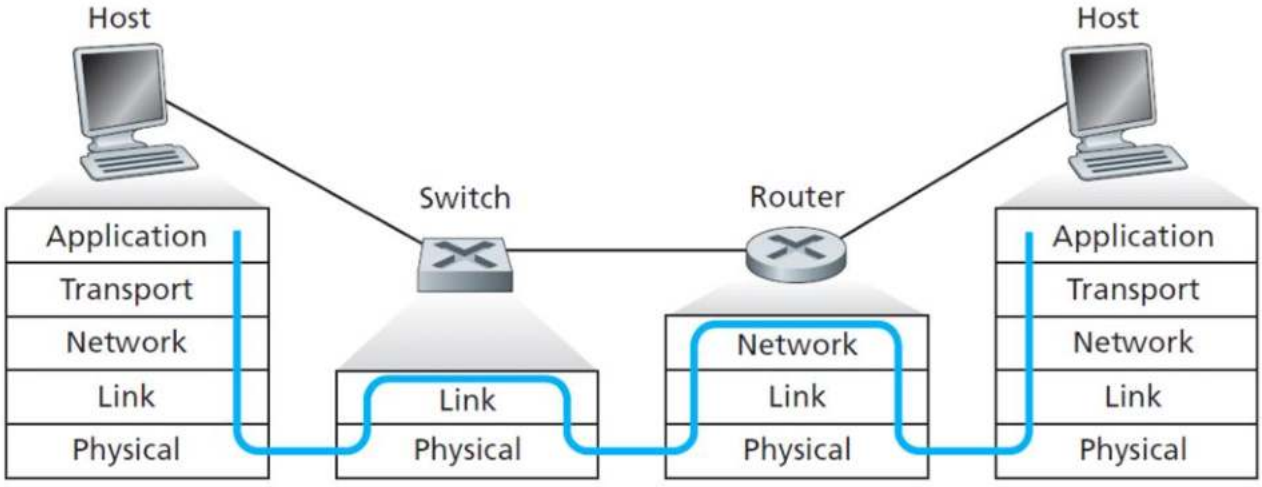


MAC addr	Interface	TTL
B	8	~~58~~ 60
L	15	~~58~~ 60
E	8	59

MAC addr	Interface	TTL
B	2	~~58~~ 60
L	7	~~58~~ 60
E	5	59
D	4	59

Switches vs. Routers

- Routers
  - Check IP address
  - Store-and-forward
  - Compute routes to destination
- Switches
  - Check MAC address
  - Store-and-forward
  - Forward frame to outgoing link or broadcast



Roadmap

6.1 Introduction to the Link Layer

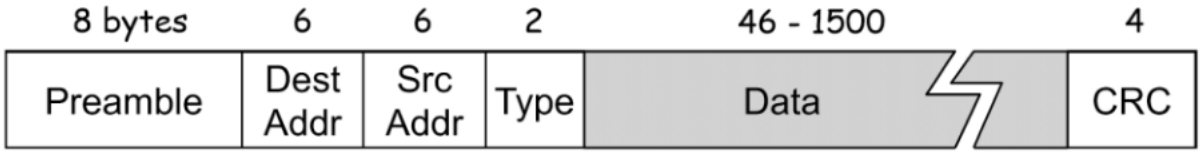
6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

Mac Address?



- Question: How to know the MAC address of a receiving host, knowing its IP address?

ARP: Address Resolution Protocol

- Question: How to know the MAC address of a receiving host, knowing its IP address?
  - Use ARP [RFC 826]
    - Provides a *query* mechanism to learn the MAC address
- Each IP node has an ARP table.
  - Stores the mappings of IP address and MAC address of other nodes in the same subnet.

< IP address; MAC address; TTL >

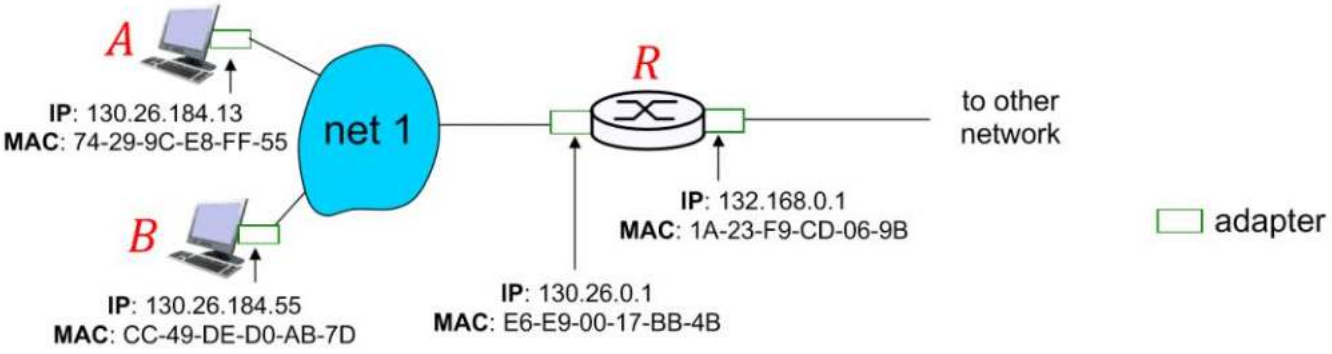
time after which address mapping will be forgotten (typically a few minutes)

ARP Demo: Linux Device

```
xilinx@pynq: ~
xilinx@pynq:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.95.4             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.5             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.6             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.7             ether   34:73:2d:1b:07:df   C                     eth0
_gateway                 ether   00:00:0c:07:ac:00   C                     eth0
192.168.95.248           ether   00:24:9b:77:29:66   C                     eth0
192.168.95.228           ether   00:e0:4c:36:00:2b   C                     eth0
xilinx@pynq:~$ ping 192.168.95.233
PING 192.168.95.233 (192.168.95.233) 56(84) bytes of data.
64 bytes from 192.168.95.233: icmp_seq=1 ttl=64 time=2.13 ms
64 bytes from 192.168.95.233: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.95.233: icmp_seq=3 ttl=64 time=1.16 ms
^C
--- 192.168.95.233 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.027/1.439/2.127/0.489 ms
xilinx@pynq:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.95.4             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.5             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.6             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.7             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.233           ether   00:e0:4c:36:10:ce   C                     eth0
_gateway                 ether   00:00:0c:07:ac:00   C                     eth0
192.168.95.248           ether   00:24:9b:77:29:66   C                     eth0
192.168.95.228           ether   00:e0:4c:36:00:2b   C                     eth0
xilinx@pynq:~$
```

Sending Frame in the Same Subnet

- Suppose A wants to send data to B. They are in the same subnet.
  1. If A knows B's MAC address from its ARP table
     - create a frame with B's MAC addresses and send it.
     - Only B will process this frame.
     - Other nodes may receive but will ignore this frame.
  2. What if A is not aware of B's MAC address?

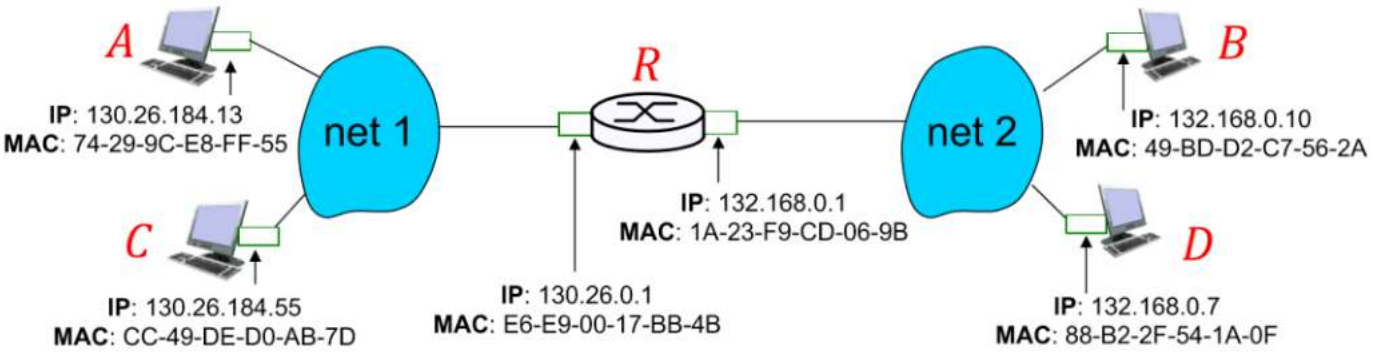


Sending Frame in the Same Subnet

- What if B's MAC address is not in A's ARP table?
  1. A broadcasts an ARP query packet, containing B's IP address.
     - Dest MAC address set to FF-FF-FF-FF-FF-FF
     - All the other nodes in the same subnet will receive this ARP query packet, but only B will reply to it.
  2. B replies to A with its MAC address.
     - Reply frame is sent to A's MAC address.
  3. A caches B's IP-to-MAC address mapping in its ARP table (until TTL expires).
- ARP is *"plug-and-play"*:
  - nodes create their ARP tables *without intervention from network administrator*

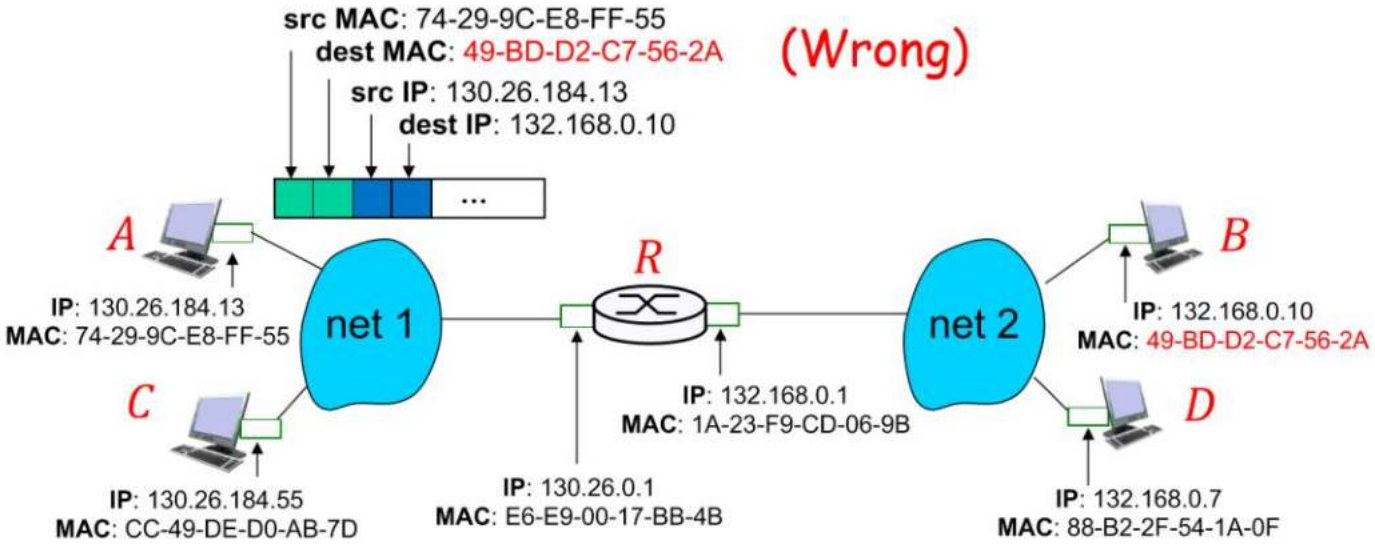
Sending Frame to Another Subnet

- Question: What if we send data to a host in another subnet?
  - For example, A sends datagram to B in another subnet.



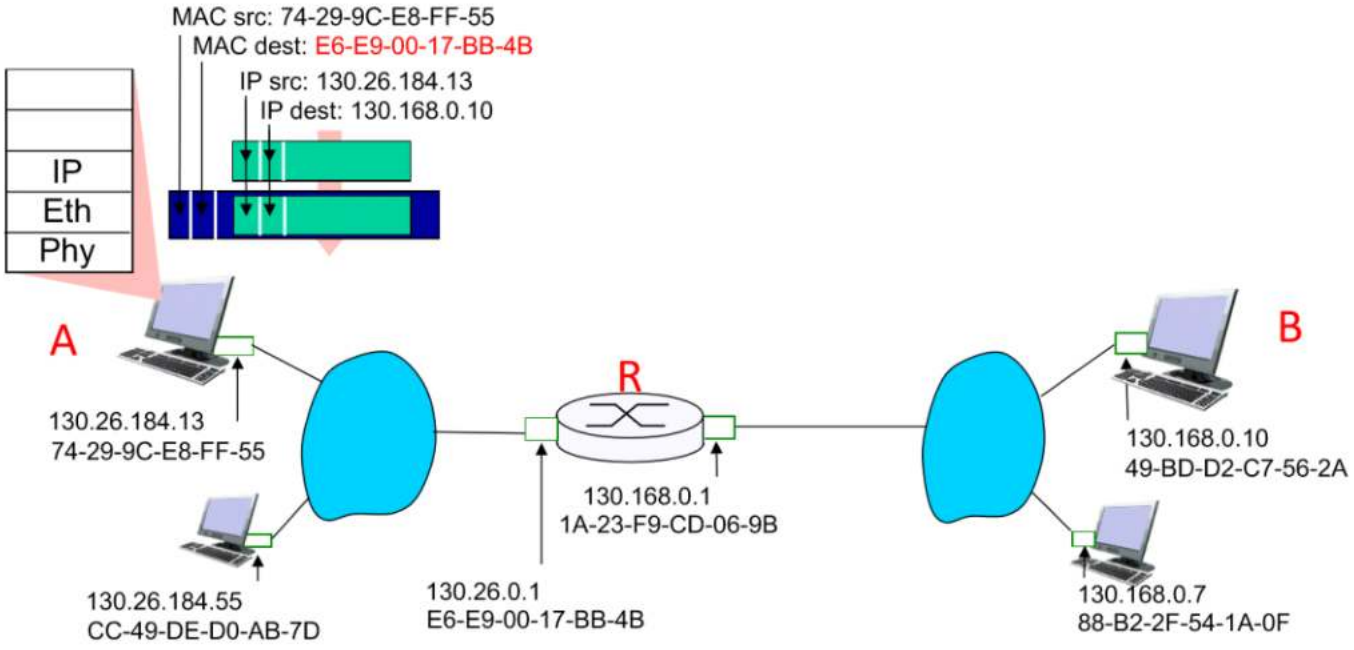
Sending Frame to Another Subnet

- *A* creates IP datagram with IP source *A*, destination *B*
- *A* creates a frame as follows
  - FALSE. all adapters in net 1 will ignore this frame because of the mismatch of destination MAC address.



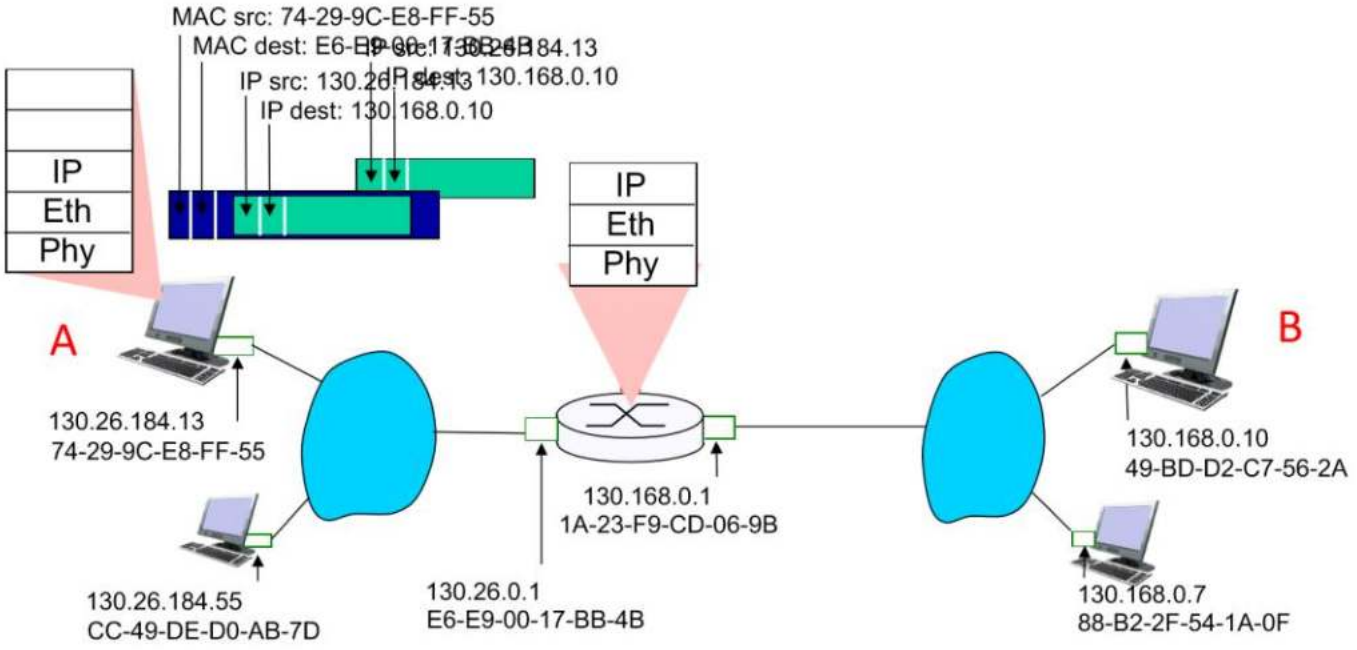
Sending Frame to Another Subnet

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram



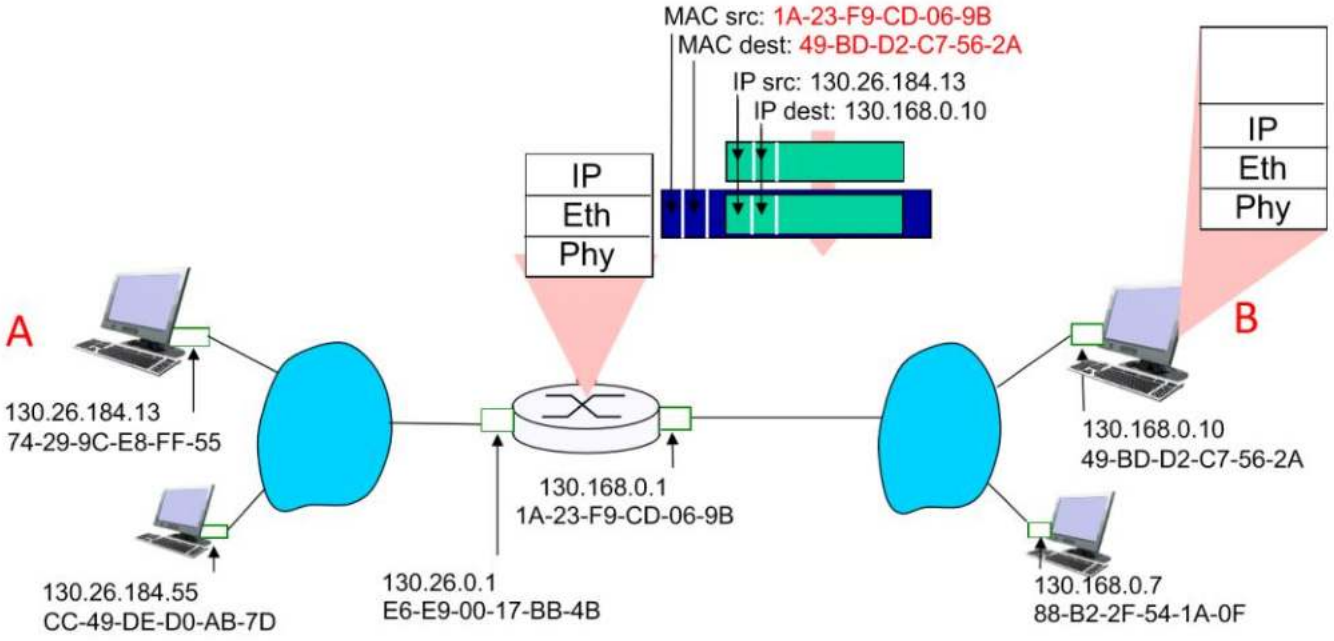
Sending Frame to Another Subnet

- frame sent from *A* to *R*
- frame received at *R*, datagram removed, passed up to IP



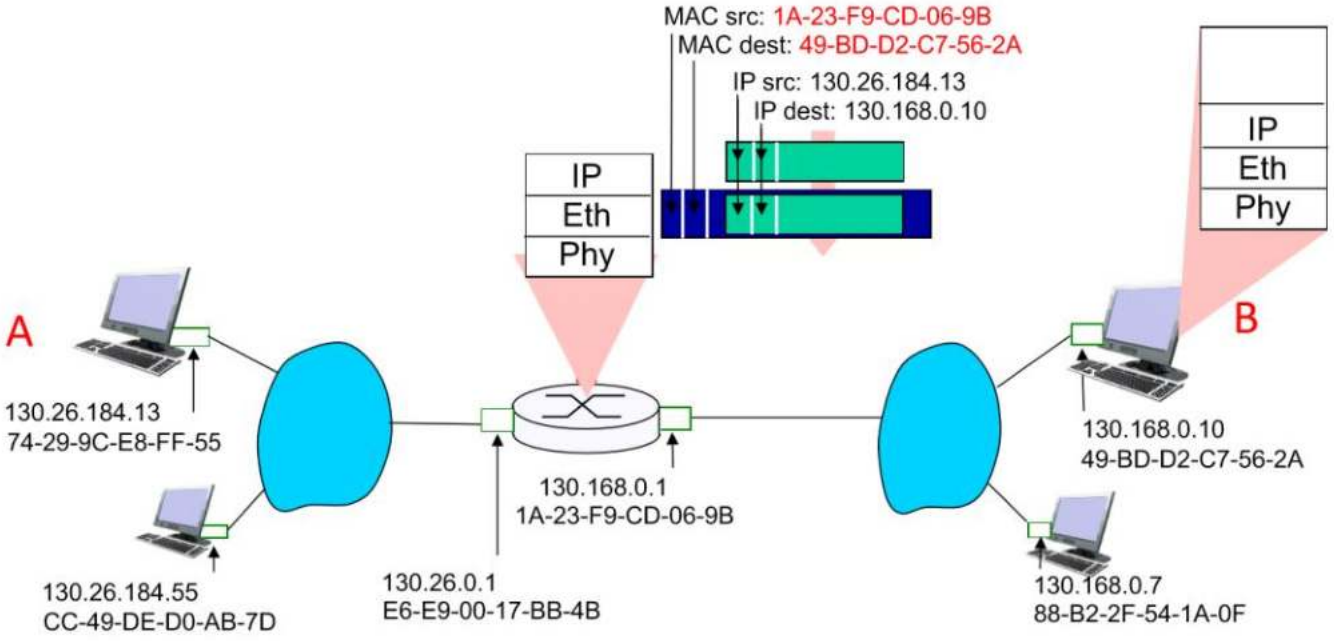
Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



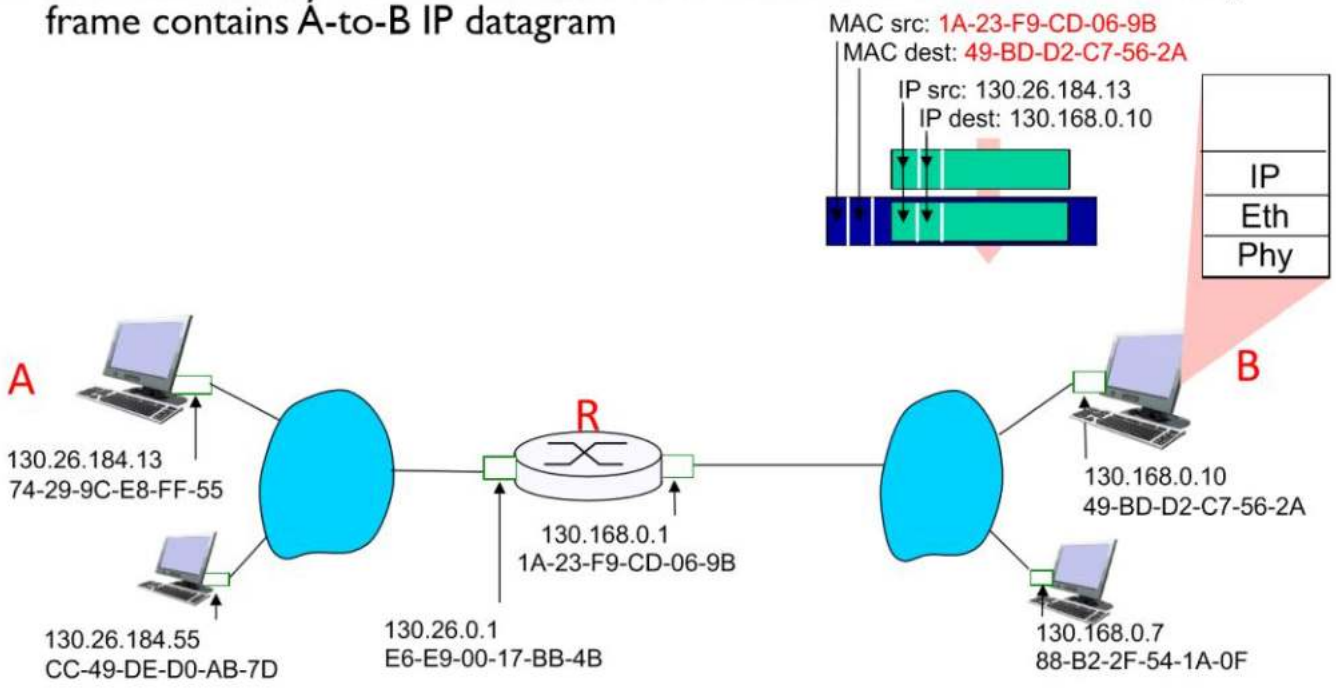
Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



Sending Frame to Another Subnet

- *R* forwards datagram with IP source *A*, destination *B*
- *R* creates link-layer frame with *B* 's MAC address as destination address, frame contains A-to-B IP datagram



IP Address vs. MAC Address

- IP address
  - 32 bits in length
  - network-layer address used to move datagrams from source to dest.
  - Dynamically assigned; hierarchical (to facilitate routing)
  - Analogy: postal address
- MAC address
  - 48 bits in length
  - link-layer address used to move frames over every single link.
  - Permanent, to identify the hardware (adapter)
  - Analogy: NRIC number

Summary

- ARP [RFC 826] resolves the mapping from network layer (IP) address to link layer (MAC) address.
- Instantiation and implementation of link layer technologies.
  - Ethernet
  - Ethernet switches and switch tables

Chapter 6: let's take a deep breath

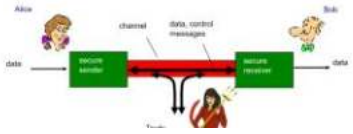
- journey down protocol stack *complete* (except PHY)
- *solid* understanding of networking principles, practice
- ….. could stop here …. but *lots* of interesting topics!
  - wireless
  - multimedia
  - security

Wk11 security

Thursday, November 2, 2023 2:10 PM

Alice, Bob, Trudy

- Bob, Alice (lovers!) want to communicate
- Trudy (intruder) may
  - Eavesdrop
  - Delete
  - Add messages



Alice and bob may repudiate

What is network security?

- Confidentiality: -
  - only sender, intended receiver should "understand" message contents
- Authentication:
  - sender, receiver want to confirm identity of each other
- Message integrity:
  - sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

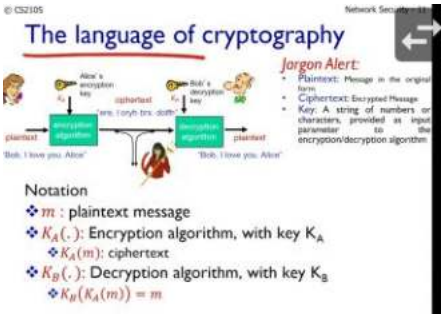
Motivation: Countering Trudy

- Bob, Alice (lovers!) want to communicate "securely"
- What would you, as Alice, do?
  - Make the Message Physically secure.
  - Use a language only know to you
    - Code Words
    - Code Language
      - Language of a gang of friends
      - Navajo Code Talkers in WWII
      - book cipher
  - Is it good?
    - Confidentiality: If the code language is secret, Trudy cannot understand.
    - Authentication: If the code language is secret, only Alice and Bob can write it.
    - Message Integrity: If the code language has an inviolable property.

Cryptographic techniques are inextricably woven into authentication, message integrity, and confidentiality

Aim:
- Confidentiality
- Authentication
- Message Integrity

The language of cryptography



Jargon Alert:
- Plaintext: Message in the original form
- Ciphertext: Encrypted Message
- Key: A string of numbers or characters, provided as input parameter to the encryption/decryption algorithm

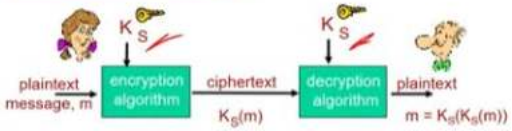
Notation
- m : plaintext message
- $K_A(.)$: Encryption algorithm, with key K_A
  - $K_A(m)$: ciphertext
- $K_B(.)$: Decryption algorithm, with key K_B
  - $K_B(K_A(m)) = m$

Types of Cryptography

Based on the values of the keys, there are *two* types of encryption

- Symmetric Key Cryptography
  - Sender and receiver use the *same key*
  - $K_A = K_B$ (We are talking about the key not the algorithms)
- Asymmetric Key Cryptography AKA Public key Cryptography
  - Sender and receiver use *different key*
  - $K_A \neq K_B$ (We are talking about the key not the algorithms)

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K_S

Q: how do Bob and Alice agree on key value?

Ans: *Will need to decide on the common key prior to communication via some other secure means, like face to face meeting*

Caesar's cipher

- This method is named after Julius Caesar, who used it in his private correspondence
- Is a form of *Substitution cipher*: substituting one thing for another
- Fixed shift of alphabet
  - e.g., right shift by 3:

```
abcdefghijklmnopqrstuvwxyz
defghijklmnopqrstuvwxyzabc
```

e.g.:
```
plaintext: the quick brown fox
ciphertext: wkh txlfn eurzq ira
```

Encryption key: only need shift number, 25 possible values

Monoalphabetic cipher

25

26!

- Substitute one letter for another
  - Is a *Substitution cipher*

```
abcdefghijklmnopqrstuvwxyz
mnbvcxzasdfghjklpoiuytrewq
```

e.g.:
```
Plaintext: bob, i love you. alice
ciphertext: nkn, s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters, 26! Mappings possible

tchow ihmou

vemry smart

Breaking an encryption scheme

- Ciphertext only attack: Trudy has ciphertext she can analyze
- Known-plaintext attack: Trudy has plaintext corresponding to ciphertext
  - e.g., in monoalphabetic cipher, Trudy determines pairings for *a,l,i,c,e,b,o,*
- Chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext
  - e.g., in monoalphabetic cipher, Trudy determines gets Alice to send *"The quick brown fox jumps over the lazy dog."*

Polyalphabetic encryption

- What is the fundamental weakness of Monoalphabetic Cipher?
  - Each letter has only one mapping
- Solution?
  - Use multiple mappings
- E.g.
  - Use n substitution ciphers, $C_1, C_2, \ldots, C_n$
  - Define a cycling pattern:
    - e.g., n=4: C_1, C_3, C_4, C_3, C_2
  - for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
    - dog: d from C_1, o from C_3, g from C_4

KEY

Polyalphabetic encryption

- Example:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
C_1	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
C_2	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
C_3	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i

Cycling Pattern: $C_1C_3C_4C_2$

e.g.:
```
plaintext: bob, i love you. alice
ciphertext: exk, o oxek bxd. durih
```

Block Ciphers

- The message to be encrypted is processed in blocks of K bits.
- For example,
  - if $K = 64$, the message is broken into 64-bit blocks
  - each block is encrypted independently.
- To encode a block, the cipher uses a one-to-one mapping

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

- E.g.: $K = 3$
  - Input: 010110001111
  - Encrypted output: 101000111001
- Number of keys: $2^K!$
  - $2^{64}!$ is an astronomical value.

Block Cipher

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit block
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in *less than a day*
- Making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

AES: Advanced Encryption Standard

- Symmetric-key NIST standard, replaced DES (Nov 2001)
- 128 bit blocks; 128, 192, or 256 bit keys
- How secure is AES?
  - Machine capable of Brute force decryption DES in 1 sec on DES, takes *149 trillion years* for 128-AES

Public Key Cryptography

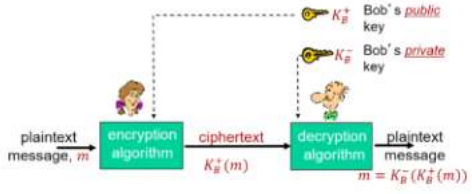
symmetric key crypto

- *Drawback*: requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key crypto

- sender, receiver *do not share* secret key
- Sender uses a *public* encryption key known to *all*
- receiver uses a *private* decryption key known *only to receiver*
- radically different approach [Diffie-Hellman'76, RSA'78]

Public key cryptography



Public key encryption algorithms

Requirements:

1. need $K_B^+(.)$ and $K_B^-(.)$ such that

$$m = K_B^-(K_B^+(m))$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

Prerequisite: modular arithmetic

- $x \bmod n$ = remainder of x when divide by n
- facts:

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$$
$$[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$$
$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

- Thus

$$(a \bmod n)^d \bmod n = a^d \bmod n$$

- example:
  - $a = 14, n = 10, d = 2$
  - $a^d \bmod n = 14^2 \bmod 10 = 6$
  - $(a \bmod n)^d \bmod n = (14 \bmod 10)^2 \bmod 10 = 16 \bmod 10 = 6$

RSA: getting ready

- *message*: just a bit pattern
  - bit pattern can be uniquely represented by an integer number
  - thus, encrypting a message is equivalent to encrypting a number

example:

- m = 10010001.
- This message is uniquely represented by the decimal number 145.
- to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

In RSA only working with integers

RSA: Creating public/private key pair

1. choose two large prime numbers p, q.
2. compute $n = pq, z = (p-1)(q-1)$
3. choose e (with $e < n$) that has no common factors with z (i.e., e and z are "relatively prime").
4. choose d such that $ed - 1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).
5. *public* key is (n, e). *private* key is (n, d).

K_B^+ K_B^-

RSA: encryption, decryption

0. given (n, e) *and* (n, d) as computed above
1. to encrypt message m (*Note:* $m < n$), compute

$$c = m^e \bmod n$$

2. to decrypt received bit pattern, c, compute

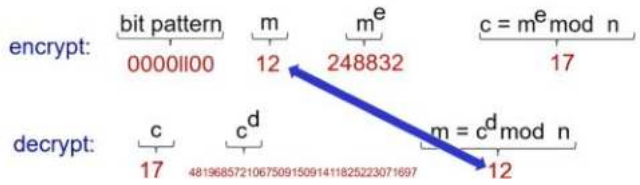
$$c^d \bmod n$$

magic happens! $(\underbrace{m^e \bmod n}_{c})^d \bmod n = m$

RSA example: Bob Chooses

1. $p = 5, q = 7$.
   1. $n = pq = 35$,
   2. $z = (p-1)(q-1) = 24$
2. $e = 5$ (with $e < n$ & e and z are "relatively prime").
3. $d = 29$ such that $ed \bmod z = 1$.

encrypting 8-bit messages.



RSA: another important property

The following property will be *very* useful later:

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

use public key first, followed by private key

use private key first, followed by public key

result is the same!

RSA in practice: session keys

- Exponentiation in RSA is computationally intensive
- DES is at least 100 times faster than RSA, but needs prior knowledge of Key K_S
- Can we Combine them?
  - Select a Key K_S
  - Use RSA to transfer K_S
  - Use K_S as the symmetric key in DES for encrypting data for this session
- The symmetric key K_S, is called the *session key*.

Message Integrity

- *Message integrity:*
  - sender, receiver want to ensure message *not altered* (in transit, or afterwards) *without detection*
- Have we *seen* this requirement earlier?
  - Does "*error detection*" fit the bill?
  - Yes
    - Checksum
    - Parity
    - CRC

Internet checksum

Internet checksum:

- produces fixed length digest (16-bit sum) of message
- is many-to-one

Consider the given message: "IOU100.99BOB"

message	ASCII format	message	ASCII format
I O U 1	49 4F 55 31	I O U 9	49 4F 55 39
0 0 . 9	30 30 2E 39	0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42	9 B O B	39 42 D2 42
	B2 C1 D2 AC	different messages but identical checksums!	B2 C1 D2 AC

- It is easy to find another message with same checksum value

CRC

Better than Checksum

- Yet poor
- Output is biased to the input
  - Minor changes in input produce minor changes in output
- We can find messages of same CRC
- E.g.
  - "Steven has fifteen white tables." and "Marla has nine red beds."
    - Both have CRC32 checksum = *2482109933*
  - "Joe has fourteen magenta things." and "Lars has thirteen black balls."
    - Both have CRC32 checksum = *93832682*

https://crc32.online/

Cryptographic Hash Function

Hash function:

- If a function $H(.)$ that takes an input m and produces fixed-size msg digest (*fingerprint*)
  - many-to-1

Cryptographic Hash function:

- Is a hash function such that it is computationally infeasible to find any two different messages x and y such that $H(x) = H(y)$
- Informally, this property means that it is computationally infeasible for an intruder to substitute one message for another message

Cryptographic Hash Function

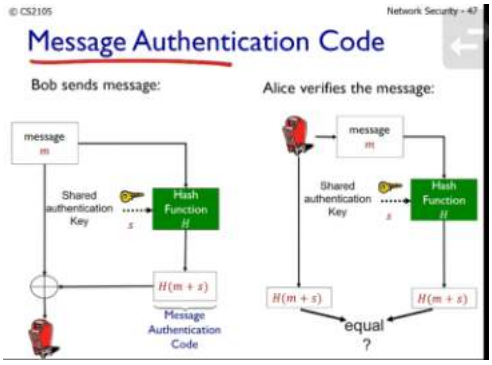
- MD5 hash function widely used (RFC 1321)
  - computes 128-bit message digest.
- SHA-1 is also used
  - US standard [NIST]
  - 160-bit message digest
- Both SHA-1 and MD5 are cryptographically broken
  - NIST formally deprecated use of SHA-1 in 2011
  - Replaced by SHA-2, SHA-3

Message Integrity

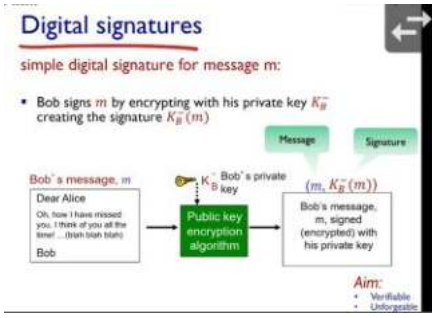
- *To ensure Message integrity:*
  - *Send* $(m, H(m))$
- *Does this work?*
  - *No!!!*
- *Recall: Message integrity:*
  - sender, receiver want to ensure message *not altered* (in transit, or afterwards) *without detection*
- What happens if the attacker replaces $(m, H(m))$ with $(m', H(m'))$?

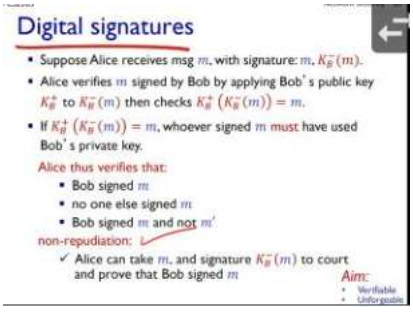
Message Authentication Code

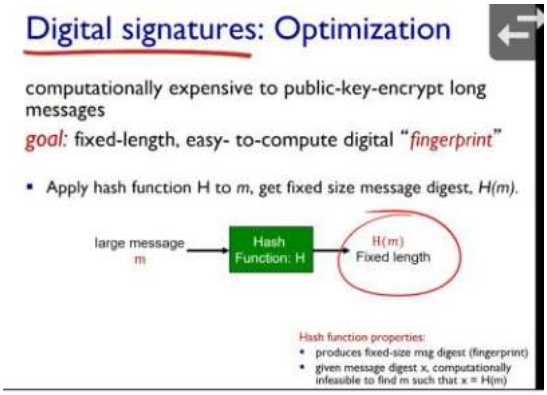
- *The sender and receiver share a "Authentication key"* s
- *To ensure Message integrity:*
  - *Send* $(m, H(m + s))$ (Message Authentication Code)
- *Does This work?*
  - *Yes!!!*
  - s *is a secret key known to the receiver and no one else*
  - *Receiver can generate the authentication code directly from* m *and compare with the received code*



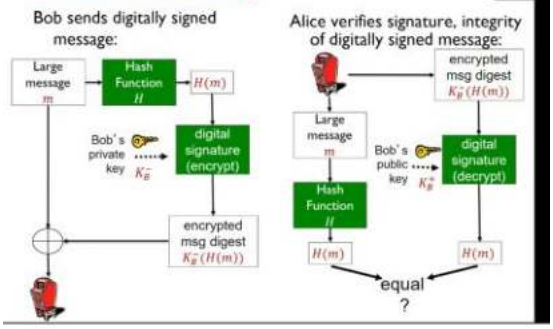
Authentication







Digital signature = signed message diges



Certification authorities

- *Problem:* *The reason we failed was, we did not know Bob's public key*
- *Solution:*
  - We create a Certification authority(CA) who maintain a public database of everyone's public key
  - Anyone who receives a message from "Bob" will access this database for K_B^+
- *Problem:* What if Trudy intercepts the communication with the CA and alters it?
- *Solution:*
  - CA signs it's messages.
- *Problem:* *We do not know CA's public key*
- *Solution:*
  - Let us make this a universal knowledge!!!!!
  - We maintain a list of CAs trusted a priori.

Tutorial_8_qns

Friday, November 3, 2023 2:02 PM

Tutorial_8_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 8** Question paper

1. **[KR, Chapter 6, 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 microseconds on a 10 Mbps Ethernet?

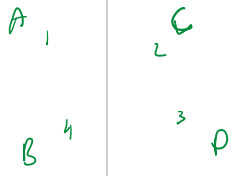
0,1,2, … 2^5 -1 = 31
1/32
1 time unit = 512 bit transmission times
4(512)(10/10^6)(1000) = 204.8 ms

2. **[Modified from KR, Chapter 6, P26]** Let's consider the operation of a learning switch in the context of a network in which 4 nodes, labeled A through D, are star connected into an Ethernet switch (refer to the diagram on Lecture 9 notes page 37/38).

   Suppose that the following events happened in sequence,

   i. B sends a frame to D
   ii. D replies with a frame to B
   iii. D sends a frame to A

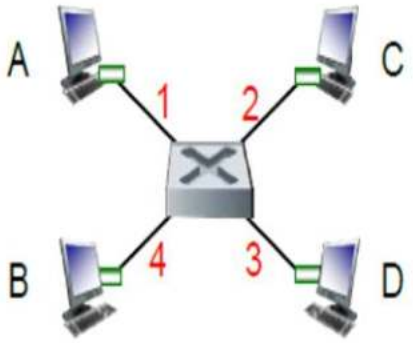
   The switch table is initially empty. Show the state of the switch table after each of the above events (ignore TTL field). For each event, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.



Event	Switch table after event	Link(s) a frame is forwarded to
B sends a frame to D	B,4	1,2,3
D replies with a frame to B	B,4 D 3	4
D sends a frame to A	B 4 D 3	1,2,4

doesn't get sent back cuz loop

3. Suppose nodes A, B and R are star connected into a switch S. A, B and R are aware of the IP addresses of each other.



a) Consider sending an IP datagram from Host A to Host B. Suppose all of the ARP tables and switch table are up to date. Enumerate all the steps the host and switch take to move the packet from A to B.

b) Repeat the problem in a), assuming that ARP table in the sending host is empty, but all other tables are up to date.

c) Repeat the problem in a), assuming that all tables in all nodes are empty.

d) Suppose A sends an IP datagram to a host in another subnet. All of the ARP tables and switch table are up to date. Enumerate all the steps the host, switch and router take to move the packet to another subnet.

4. Wireshark: Ethernet

Do the following:

1. Make sure your browser's cache is empty. To do this, Clear Recent History.
2. Start up the Wireshark packet sniffer.
3. Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html.
4. Stop Wireshark packet capture.

Answer the following questions:

1. Based on the contents of the Ethernet frame containing the HTTP GET message:
    a. What is the 48-bit Ethernet address of your computer?

b. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? What device has this as its Ethernet address?

2. Based on the contents of the Ethernet frame containing the first byte of the HTTP response message:

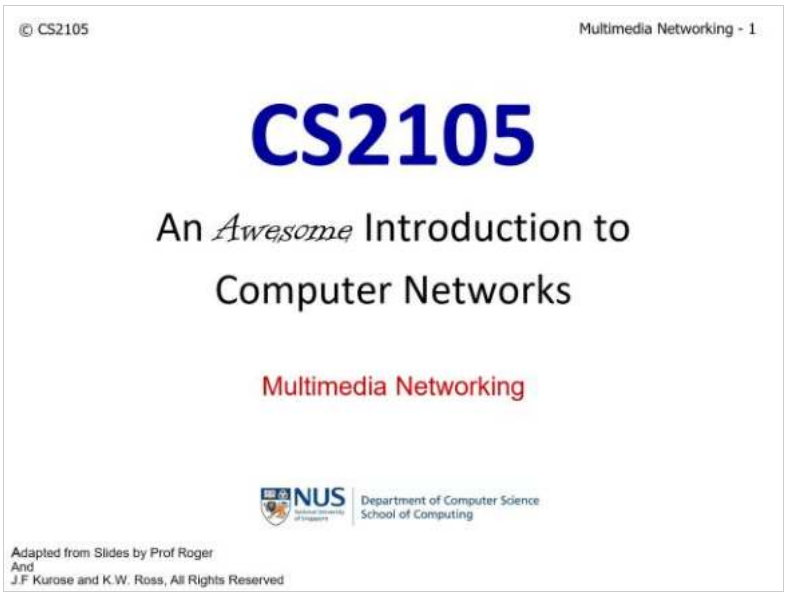
   a. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu. What device has this as its Ethernet address?

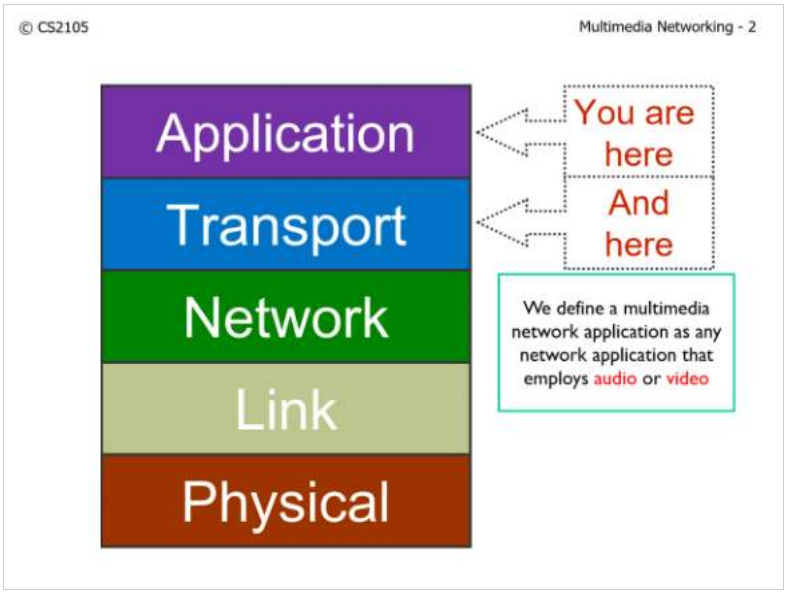
   b. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

Lec_11_Multimedia_Networking_pre_lecture

Monday, November 6, 2023 2:03 PM

Lec_11_Multimedia_N...





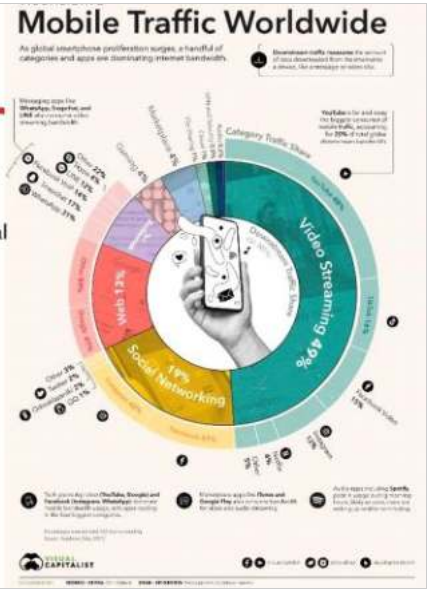
Multimedia networking: outline

9.1 multimedia networking applications
9.2 streaming *stored* video
9.3 voice-over-IP
2.6 dynamic adaptive streaming over HTTP (DASH)

Motivation

- Sandvine, THE GLOBAL INTERNET PHENOMENA REPORT, JAN 2022:
  - In 2021, **53.7%** of the global Internet traffic was video
- Top users of internet:
  - YouTube (14.6%)
  - Netflix (9.39%)
- All these are delivered as OTT

Jargon Alert:
- OTT: over-the-top



Multimedia networking: 3 application types

- *Streaming stored* audio, video
  - *Streaming:* can begin playout before downloading entire file
  - *Stored (at server / CDNs):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *Conversational ("two-way live")* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
    - Delay more than 400 milli seconds, intolerable
  - e.g., Skype, Zoom, WhatsApp
- *Streaming live ("one-way live")* audio, video
  - Typically done with CDNs
  - e.g., live sporting event (soccer, football)

Jargon Alert:
- CDN: Content Distribution Network

You have 10 s ish to send for streaming live

Multimedia: video

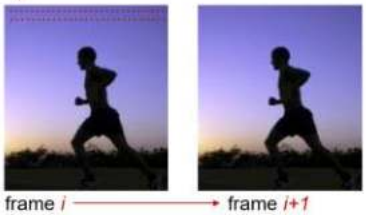
- Video: sequence of images displayed at constant rate
  - e.g., 30 images/sec
- Digital image: array of pixels
  - each pixel represented by bits

- The most salient characteristic of video is its *high bit rate*
- To reduce data usage, we compress the video:
  - use redundancy *within* and *between* images to decrease # bits used to encode image

Multimedia: video

- Use redundancy *within* and *between* images to decrease # bits used to encode image
  - Spatial Coding (within image)
  - Temporal Coding (from one image to next)

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)



temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

Video: original frame *i*

Difference between 2 frames

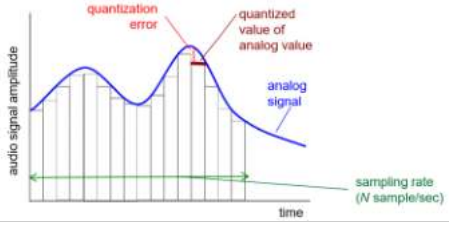
Multimedia: video

- CBR: (constant bit rate): video encoding rate fixed
  - Not responsive to the complexity of the video.
  - Need to set your bitrate relatively high to handle more complex segments of video.
  - The consistency of CBR makes it well-suited for real-time encoding.
    - For *real-time live streaming*
- VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
  - VBR best suited for *on-demand video* due to longer time to process the data.
- examples:
  - MPEG 1 (CD-ROM) 1.5 Mbps
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4/H.264 (often used in Internet, < 2 Mbps)
  - H.265, 4K video > 10 Mbps

The more movement you have, more bit rate you need

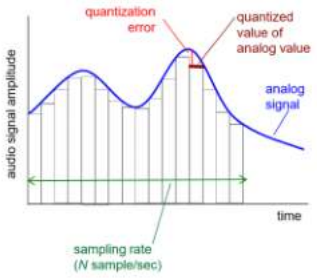
Multimedia: audio

- Analog audio signal
  - sampled at constant rate
    - Telephone: 8,000 samples/sec
    - CD music: 44,100 samples/sec
- each sample quantized, i.e., rounded
  - each quantized value represented by bits, e.g., 8 bits for 256 (2^8) values



Multimedia: audio

- example: 8,000 samples/sec, 256 quantized values (8 bits): 64,000 bps
- receiver converts bits back to analog signal (DAC):
  - some quality reduction



example rates

- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up

Jargon Alert:
- ADC: analog-to-digital converter
- DAC: digital-to-analog converter

Mp3 was compressed by removing redundany by exploiting human hearring

Multimedia networking: outline

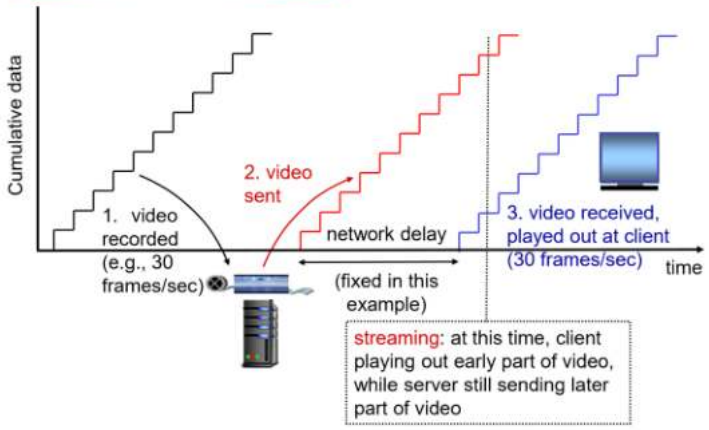
9.1 multimedia networking applications
9.2 streaming *stored* video
9.3 voice-over-IP
2.6 dynamic adaptive streaming over HTTP (DASH)

Streaming stored video:

- *Streaming stored* video
  - *Streaming:* Can begin playout *before* downloading entire file
  - *Stored (at server / CDNs):* can transmit faster than audio/video will be rendered (implies *storing/buffering* at client)

Jargon Alert:
- CDN: content distribution Network

Streaming stored video:

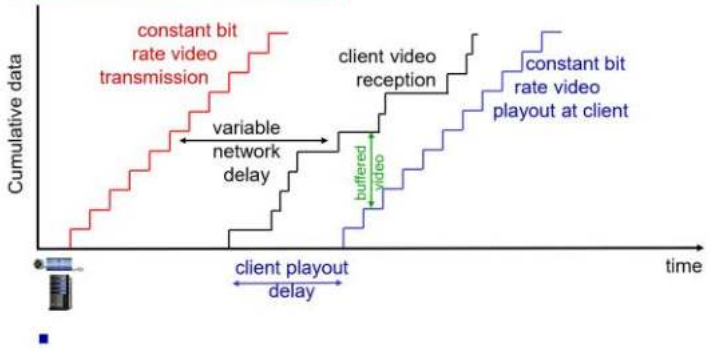


Streaming stored video: challenges

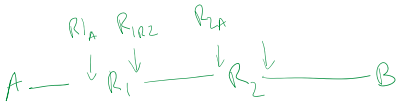
- Continuous playout constraint:
  - once client playout begins, playback *must match* original timing
  - … but network delays are variable (jitter),
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted

No freezing should occur
Even tho network delay is not constant

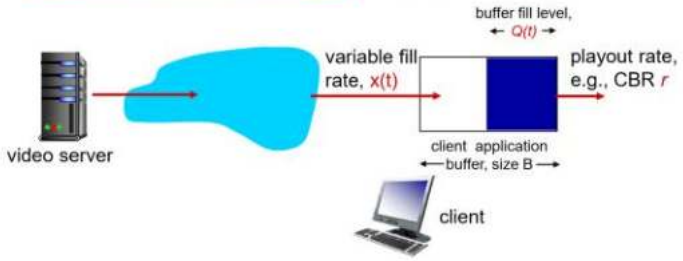
Streaming stored video: revisited



Want to ensure playout does not intersect client reception
"buffering" to compensate for network added delay, to delay jitter

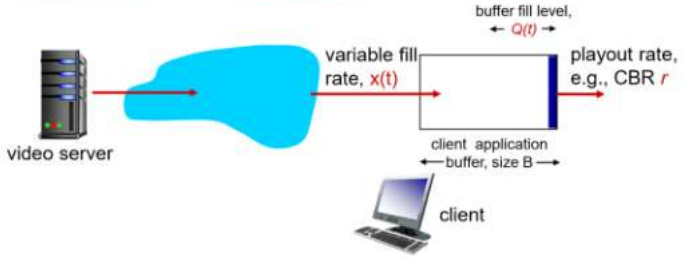


Client-side buffering, playout



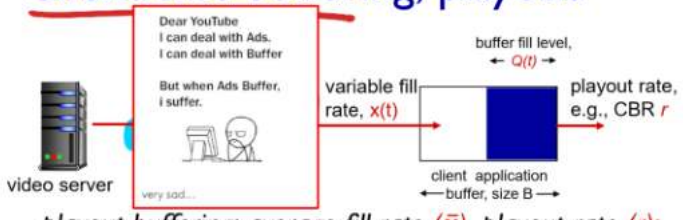
x(T) rate at which buffer growing
Q(t) current size of buffer
Ensure buffer is never below r

Client-side buffering, playout



1. Initial fill of buffer until playout begins at t_p
2. playout begins at t_p,
3. buffer fill level varies over time as fill rate x(t) varies and playout rate r is constant

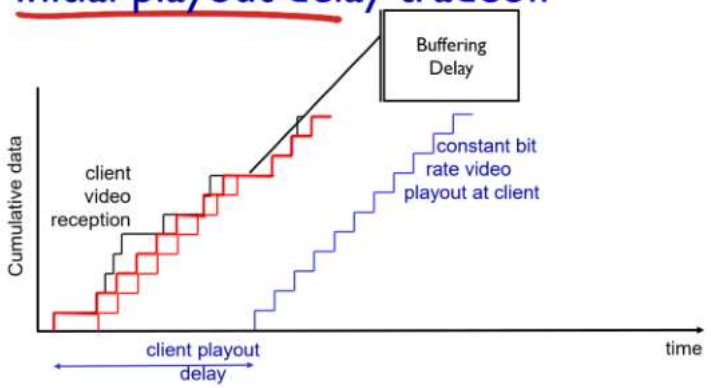
Client-side buffering, playout



playout buffering: average fill rate ($\bar{x}$), playout rate (r):

- $\bar{x}$ < r: buffer eventually empties (causing freezing of video playout until buffer again fills)
- $\bar{x}$ > r: buffer will not empty

Initial playout delay tradeoff



Buffering

- You are using a media player which has a playout buffer size of $B_{Playout}$ = 8 MB.
- The buffer is initially empty. The time when you press "play" for a video is t_0. It takes 4 seconds to fill the buffer $B_{Playout}$ to its mid-point, i.e., 4 MB of data.
- At that point (t_0 + 4 secs) the media player starts to play the video.
- The video has a size of 14 MB.
- The data continues to arrive after (t_0 + 4 secs) from the server at a constant rate of 8 Mb/s and the player plays (decodes) the media at a rate of 4 Mb/s until the video ends.
- Which of the following statements are **TRUE**? (Times are measured relative to t_0).

8 Mb = 1 MB

T = 4	4mb	
T=5	12 mb	8 mb in buffer
T = 6		4mb
T = 7	16 mb	8 mb in buffer

3 MB = 64 Mb

14 MB = 112 Mb

	buffer	played
t_0+4	32 Mb	0
t_0+5	36 Mb	4
⋮		
t_0+12	64 Mb	32
t_0+13	60 Mb	36
14	56	40
⋮		
$t+12+4$		
$t+16$		112 ← t_0+32

96 Mb remains pause

t_0+28+4

Which of the following statements are TRUE? (Times are measured relative to t_0)

- The video will play normally for the whole duration of the video and will end at t_0 + 28 seconds. ✓
- The $B_{Playout}$ buffer will overflow at t_0 + 12 seconds. (And the video may stall/stop.) ✓
- The $B_{Playout}$ buffer will overflow at t_0 + 8 seconds. (And the video may stall/stop.)
- The $B_{Playout}$ buffer will underflow at t_0 + 12 seconds. (And the video may stall/stop.)
- The server will have delivered all the data of the video to the client at t_0 + 12 seconds.

A ,B,E

Poll locked. Responses not accepted.

Which of the following statements are TRUE? (Times are measured relative to t_0)

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Poll locked. Responses not accepted.

Which of the following statements are TRUE? (Times are measured relative to t_0)

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Buffering

- $Size(video) = 14MB = 112Mb \quad = 28\ chunks$
- $B_{Playout} = 8MB = 64Mb \quad = 16\ chunks$
- $Q(t_0 + 4) = 4MB = 32Mb \quad = 8\ chunks$
- $t_p = t_0 + 4$
- fill rate, $\bar{x} = 8Mbps \quad = 2\ chunks\ per\ sec$
- playout rate, $r = 4Mbps \quad = 1\ chunk\ per\ sec$
- Rate of buffer growth $= \bar{x} - r = 4Mbps \quad = 1\ chunk\ per\ sec$

- Time taken by the buffer to fill $= (t_0 + 4) + \frac{16-8}{1}$
  $= t_0 + 12$
- Video played till $(t_0 + 12) = (t_0 + 12) - (t_0 + 4) = 8\ chunks$
- Data delivered till $(t_0 + 12) = (8 + 16) \quad = 24\ chunks$
- fill rate after $(t_0 + 12)$, $\bar{x} = 1\ chunk\ per\ sec$
- Time to deliver the video in total $= (t_0 + 12) + (size(video) - data\ delivered)/\bar{x}$
  $= (t_0 + 12) + \frac{28 - 24}{1}$
  $= t_0 + 16$
- End of play time = $(t_0 + 4) + 28 = t_0 + 32$

Streaming multimedia: UDP

- server sends at rate appropriate for client
  - often: send rate = encoding rate = constant rate,
    - ☞ push-based streaming (*server push*)
  - UDP has no congestion control
    - Hence transmission without rate control restrictions
- short playout delay (2-5 seconds) to remove network jitter
- Error recovery: application-level, time permitting

Streaming multimedia: UDP

- Video chucks encapsulated using RTP

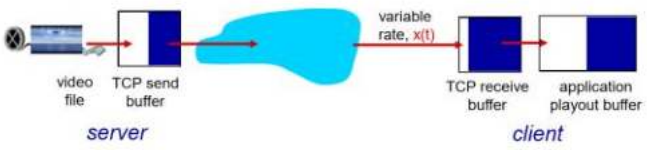
Seq num,
Time stamp,
Video encoding

- Control Connection is maintained *separately* using RTSP
  - Is used for establishing and controlling media sessions between endpoints.
  - Clients issue commands such as *play*, *record* and *pause*
- Drawbacks
  - Need for a separate media control server like RTSP, increases cost and complexity
  - UDP may *not* go through firewalls

Jargon Alert:
- RTP: Real time Transport protocol
- RTSP: Real time Streaming protocol

Streaming multimedia: HTTP

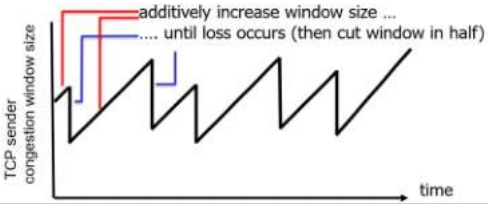
- multimedia file retrieved via HTTP GET,
  ☞ pull-based streaming (*client pull*)
- send at maximum possible rate under TCP



Streaming multimedia: HTTP

- Advantages
  - HTTP/TCP passes more easily through *firewalls*
  - Network *infrastructure* (like CDNs and Routers) fine tuned for HTTP/TCP
- Drawbacks
  - fill rate *fluctuates* due to TCP congestion control, retransmissions (in-order delivery)
  - *larger* playout delay: smooth TCP delivery rate

Additive Increase Multiplicative Decrease *saw tooth* behavior: *probing for bandwidth*



Multimedia networking: outline

9.1 multimedia networking applications
9.2 streaming *stored* video
9.3 voice-over-IP
2.6 dynamic adaptive streaming over HTTP (DASH)

Conversational Multimedia: VoIP

- *VoIP end-end-delay requirement*: needed to maintain "conversational" aspect
  - Higher delays noticeable, impair interactivity
    - < 150 msec: good
    - > 400 msec: bad
    - includes application-level (packetization, playout), network delays
  - Data loss over 10% makes conversation unintelligible.
- *Challenge:*
  - Internet (IP layer) is a *best-effort* service
    - No upper bound on *delay*
    - No upper bound on percentage of *packet loss*

Jargon Alert:
- VoIP: Voice over IP

VoIP characteristics

- Speaker's audio:
  - alternating talk spurts, silent periods.
  - pkts generated only during *talk spurts*
  - 20 msec *chunks* at 8 Kbytes/sec: 160 bytes of data
- application-layer header added to each chunk
- *chunk+header* encapsulated into UDP or TCP segment
  - application sends segment into socket every 20 msec during talk spurt
- *Challenge:*
  - No upper bound on *delay*
  - No upper bound on percentage of *packet loss*

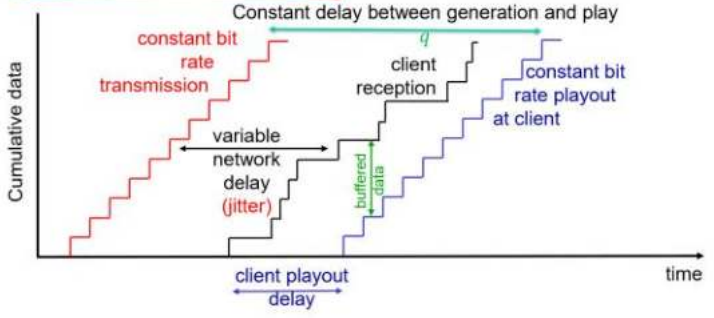
When to play back a chuck?

What to do with missing chuck?

VoIP: packet loss, delay

- *Network loss:* IP datagram lost due to network congestion (router buffer overflow, etc.)
- *Delay loss:* IP datagram arrives too late for playout at receiver
  - *delays*: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
  - VoIP Applications typically use UDP to avoid Congestion control.
- *loss tolerance:* depending on voice encoding, loss concealment, packet loss rates between 1% and 10% can be tolerated

Delay jitter

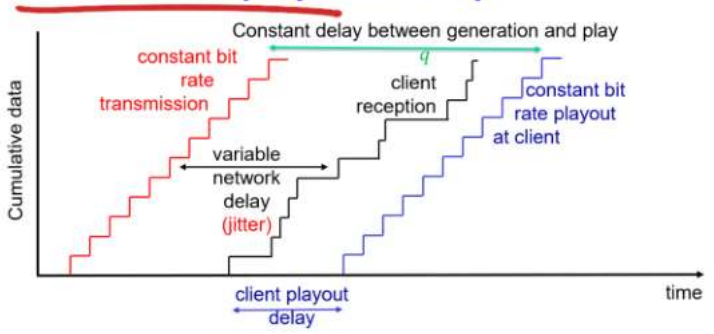


VoIP: fixed playout delay

- receiver attempts to playout each chunk exactly q msecs after chunk was generated.
  - chunk has time stamp t: play out chunk at $t + q$
  - chunk arrives after $t + q$: data arrives too late for playout: data "lost"
- tradeoff in choosing q :
  - *large* q : less packet loss
  - *small* q : better interactive experience

Every Chunk will have
- Sequence Number
- Timestamp

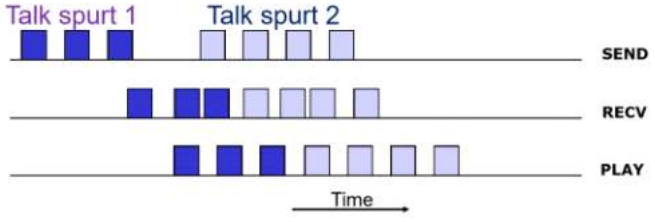
VoIP: fixed playout delay



- No value of q can guarantee an optimal performance
  - We will eventually have a packet loss, or
  - Waste a lot of playout time

Adaptive playout delay

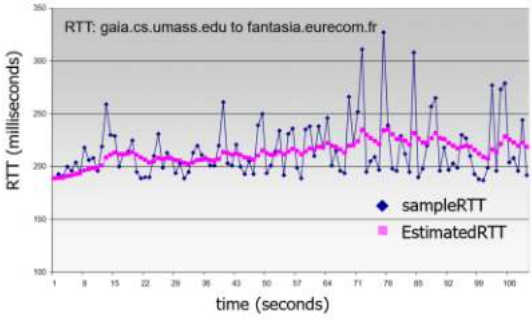
- *goal:* low playout delay, low late loss rate
- *approach:* adaptive playout delay adjustment
  - *estimate* network delay, *adjust* playout delay at beginning of each talk spurt
  - silent periods *compressed* and *elongated*
    - chunks still played out every 20 msec during talk spurt



Recall: TCP round trip time, timeout

`EstimatedRTT = (1- α)*EstimatedRTT + α*SampleRTT`

- exponential weighted moving average
- influence of past sample decreases exponentially fast
- typical value: $\alpha = 0.125$



Recall: TCP round trip time, timeout

- timeout interval: `EstimatedRTT` plus "safety margin"
  - large variation in `EstimatedRTT` -> larger safety margin
- estimate SampleRTT deviation from EstimatedRTT:

```
DevRTT = (1-β)*DevRTT +
         β*|SampleRTT-EstimatedRTT|
         (typically, β = 0.25)
```

```
TimeoutInterval = EstimatedRTT + 4*DevRTT
```

estimated RTT — "safety margin"

Adaptive playout delay

Jargon Alert:
- EWMA: exponentially weighted moving average

- Adaptively estimate packet delay (EWMA):

$$d_i = (1-\alpha)d_{i-1} + \alpha\,(r_i - t_i)$$

delay estimate after ith packet — *small constant, e.g. 0.1* — *time received - time sent (timestamp)* = *measured delay of ith packet*

estimate of average deviation of delay after ith packet

$$v_i = (1-\beta)v_{i-1} + \beta\,|r_i - t_i - d_i|$$

- Estimates, d_i *and* v_i calculated for every received packet, but used only at start of talk spurt
  - for first packet in talk spurt, playout time is:

$$playout\text{-}time_i = t_i + d_i + 4v_i$$

  - remaining packets in talk spurt are played out periodically

VoIP: recovery from packet loss

Challenge: recover from packet loss given small tolerable delay between original transmission and playout

- Use ACK/NAK
  - Each ACK/NAK takes ~ one RTT
  - Too slow
- Alternative: *Forward Error Correction (FEC)*
  - send enough bits to allow recovery without retransmission (recall two-dimensional parity)

VoIP: recovery from packet loss

Simple FEC

- for every group of n chunks
  - create redundant chunk by XOR-ing n original chunks
  - send $n+1$ chunks
- can reconstruct original n chunks if at most one lost chunk from $n+1$ chunks, with playout delay
- Drawback
  - Increasing bandwidth by factor $1/n$
  - Playout delay is increased during packet loss
    - Receiver waits for $n+1$ chunks before playout

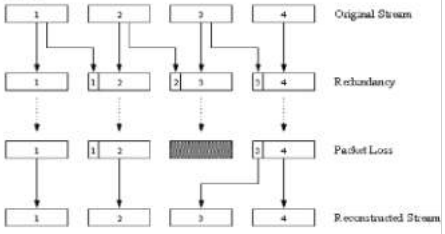
```
1 0 1 0 1 1
1 1 0 1 0 1
1 0 1 0 0 0
1 1 0 1 1 0
```

XOR (exclusive OR)
- Commutative
- Associative

VoIP: recovery from packet loss

Another cool FEC scheme:

- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
  - e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps



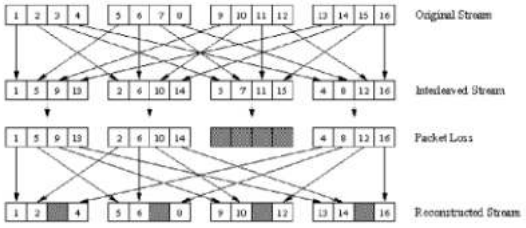
- *non-consecutive* loss: receiver can *conceal* loss
- generalization: can also append (n-1)st and (n-2)nd low-bit rate chunk

The retransmitted low quality version helps regenerate the original chunk

VoIP: recovery from packet loss

Interleaving to conceal loss:

- Audio chunks divided into smaller units, e.g., four 5 msec units per 20 msec audio chunk
- packet contains small units from different chunks
- if packet lost, still have *most* of every original chunk
  - Concealed by packet repetition or interpolation
- no redundancy overhead, but increases playout delay, even without error



So you can brazen it out, user will not feel the difference that data has been lost

Multimedia networking: outline

9.1 multimedia networking applications
9.2 streaming *stored* video
9.3 voice-over-IP
2.6 dynamic adaptive streaming over HTTP (DASH)

HTTP Streaming

- Video-on-Demand (VoD) video streaming increasingly uses HTTP streaming
  - Simple HTTP streaming just GETs a (whole) video file from an HTTP server
- Drawbacks
  - can be wasteful, needs large client buffer
  - All client receive the *same* encoding of video, despite the variation in the device/network bandwidth

Solution:
Dynamic Adaptive Streaming over HTTP

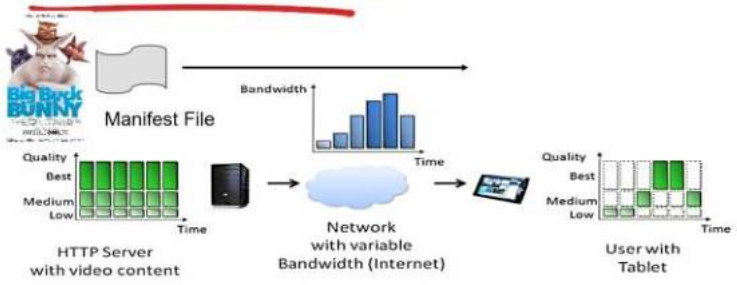
Streaming multimedia: DASH

- *DASH: Dynamic, Adaptive Streaming over HTTP*
- *server:*
  - divides video file into *multiple* chunks
  - each chunk stored, encoded at *different rates*
  - *manifest file:* provides URLs for different encodings
- *client:*
  - periodically *measures* server-to-client bandwidth
  - consulting manifest, requests one chunk at a time
    - chooses maximum coding rate *sustainable* given current bandwidth
    - *can choose* different coding rates at different points in time (depending on available bandwidth at time)

Streaming multimedia: DASH

- *DASH: Dynamic, Adaptive Streaming over HTTP*
- *"intelligence" at client:* client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
  - *what encoding rate* to request (higher quality when more bandwidth available)
  - *where* to request chunk (can request from URL server that is "close" to client or has high available bandwidth)

How DASH works



- Data is encoded into different qualities and cut into short segments (streamlets, chunks).
- Client first downloads *Manifest File*, which describes the available videos and qualities.
- Client/player executes an adaptive bitrate algorithm (ABR) to determine which segment do download next.

Streaming multimedia: DASH

- Advantages of DASH
  - Server is simple, i.e., regular web server (no state, proven to be scalable)
  - No firewall problems (use port 80 for HTTP)
  - Standard (image) web caching works
- Disadvantages
  - DASH is based on media segment transmissions, typically 2-10 seconds in length
  - By buffering a few segments at the client side, DASH does not:
    - Provide low latency for interactive, two-way applications (e.g., video conferencing)

Content distribution networks

- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of *simultaneous* users?

- *option 1:* single, large "mega-server"
  - single point of failure
  - point of network congestion
  - long path to distant clients
  - multiple copies of video sent over outgoing link

….quite simply: this solution *doesn't scale*

Content distribution networks

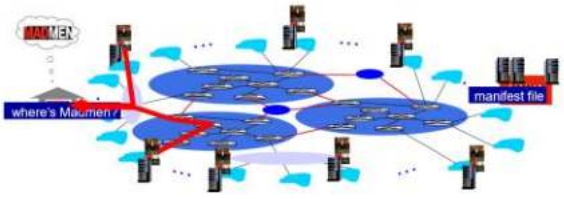
- *challenge:* how to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

- *option 2:* store/serve multiple copies of videos at multiple geographically distributed sites *(CDN)*
  - *enter deep:* push CDN servers deep into many access networks
    - Usually at ISP (Internet Service Provides)
    - close to users
    - used by Akamai, 1700+ locations
  - *bring home:* smaller number (10's) of larger clusters in IXPs near (but not within) access networks
    - used by Limelight

Jargon Alert:
- IXP: *Internet* exchange point

Content distribution networks (CDNs)

- CDN: stores copies of content (e.g. MADMEN) at CDN nodes
- Client requests content
  - service provider returns manifest
- using manifest, client retrieves content at highest supportable rate
- may choose different rate or copy if network path congested



Summary

- Encoding exploiting
  - Spatial redundancy
  - Temporal redundancy
- Client-side Buffering
  - Playout delay
  - Congestion Control
- VoIP
  - FEC
  - Error concealment
- Video Streaming
  - UDP
  - HTTP
  - DASH
  - CDN

There are various media applications on the Internet, even though it provides (few) guarantees

Tutorial_9_qns

Friday, November 10, 2023 2:48 AM

Tutorial_9_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 9** Question paper

1. [KR, Chapter 8, P1] Using the substitution cipher (monoalphabetic cipher) shown in Lecture 10, on page 15, notes:
   a) encode the message "this is a secret message" vasi si m icbocu hciimzc
   b) decode the message "tcow ihmou" very smart

2. [KR, Chapter 8, R6] Suppose N people each want to communicate with $N-1$ other people. All communication between any two people, i and j, is visible to all other people but no other person should be able to decode their communication. In total, how many keys are required in this group if:
   a) Symmetric key encryption is used in each communication?
   b) Public key encryption is used in each communication?

Symmetric key for each pair. n*(n-1)/2

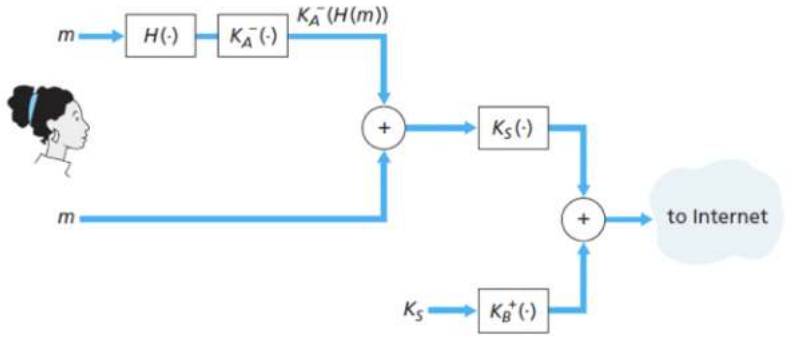
2N keys

3. [KR, Chapter 8, P13] In the BitTorrent P2P file distribution protocol, the seed breaks a file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn't redistribute bogus blocks.
   Assume that when a peer joins a torrent, it initially gets a .torrent file from a *fully trusted* source. Describe a simple scheme that allows peers to verify the integrity of blocks.

30%mcq

4. Suppose Alice wants to send a secure email m to Bob, and wants to ensure its confidentiality and integrity. Alice performs the following steps (Figure 8.21 on textbook which is reproduced below):

Page 1 of 2



1. generates a random session key K_S
2. encrypts the session key with Bob's public key K_B^+, obtaining $K_B^+(\text{KS})$
3. hashes the message m with a cryptographic hash function H, obtaining message digest $H(m)$
4. encrypts the hash with Alice's private key K_A^-, obtaining digital signature $K_A^-(H(m))$
5. encrypts the message m, concatenated ($\oplus$) with $K_A^-(H(m))$, using the session key K_S to obtain $K_S(\, m \oplus K_A^-(H(m))\,)$
6. finally, sends $K_S(\, m \oplus K_A^-(H(m))\,) \oplus K_B^+(K_S)$ to Bob

Show what Bob has to do to verify that m is indeed crafted by Alice and has not been modified during transmission.

Decrypt the session key using bob's private key
Use Alice's public key to decrypt

Tutorial_10_qns

Friday, November 17, 2023 1:04 PM

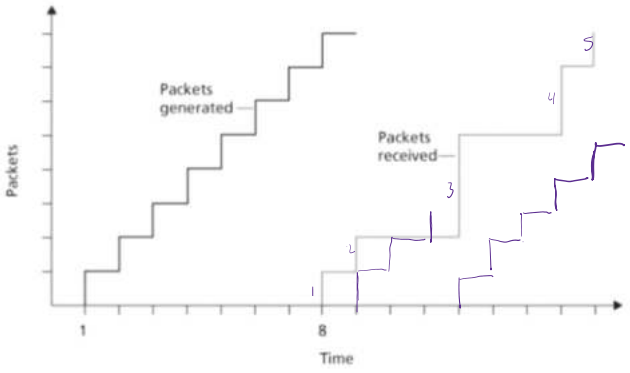
Tutorial_10_qns

National University of Singapore
School of Computing

CS2105 **Tutorial 10** Question paper

1. **[KR, Chapter 9, R2]** There are two types of redundancy in video. Describe them, and discuss how they can be exploited for efficient compression.
   redundancy within frames and between frames
   spacial compressed using vectors
   temporal compressed by sending only difference
2. **[KR, Chapter 9, R3]** Suppose an analog audio signal is sampled 16,000 times per second, and each sample is quantized into one of 1,024 levels. What would be the resulting bit rate of the PCM digital audio signal (i.e. uncompressed digitized audio signal)?
   For 1024 values of quantized
   $2^{10} = 1024$ so 10 bits needed
   $10 \cdot 16000 =$ 160000 bps
3. **[KR, Chapter 9, R7]** With HTTP streaming, are the TCP receive buffer and the client's application buffer the same thing? If not, how do they interact?
   diff buffers serve diff purposes
   TCP buffer sends to application buffer
4. **[KR, Chapter 9, R10]** Why is a packet that is received after its scheduled playout time considered lost?
   its discarded
5. In practice, RTP tends to be used over UDP while RTSP tends to be used over TCP. Why might this be so?
   RTCP provides inf.
6. **[KR, Chapter 9, P11]** Consider the figure below. A sender begins sending packetized audio periodically at $t = 1$. The first packet arrives at the receiver at $t = 8$.

Page 1 of 2



pkt 3 - 2+7
pkt 4 - 2+7

a) What are the delays (from sender to receiver, ignoring any playout delays) of packets 2 through 8? Note that each vertical and horizontal line segment in the figure has a length of 1, 2, or 3 time units.

b) If audio playout begins as soon as the first packet arrives at the receiver at $t = 8$, which of the first eight packets sent will *not* arrive in time for playout? pkt 3

c) If audio playout begins at $t = 9$, which of the first eight packets sent will not arrive in time for playout? pkt 3

d) What is the minimum playout delay at the receiver that results in all of the first eight packets arriving in time for their playout? t=10

7. **[Modified from KR, Chapter 9, P13]** Recall the two FEC schemes for VoIP described in lecture. Suppose the first scheme (Scheme 1) generates a redundant chunk for every four original chunks. Suppose the second scheme (Scheme 2) uses a low-bit rate encoding whose transmission rate is 25 percent of the transmission rate of the nominal stream. (Note: we ignore the effects of playout delay in this question as we assume that all packets, including FEC packets, will be received prior to reconstruction and playback)

a) How much additional bandwidth does each scheme require?

b) How do the two schemes perform if the first packet is lost in every group of five packets? Which scheme will have better audio quality? Scheme 1

c) How do the two schemes perform if the first packet is lost in every group of two packets? Which scheme will have better audio quality? Scheme 2 A B E P
cuz A and C cannot be determine

3103 networds
4226 network arch
3235 cybersec

joining network - only know own mac
learn IP of gateway router + IP of ourself

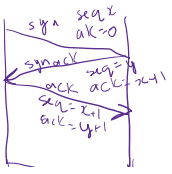
ARP
0 - (2^10 - 1) port # received

DNS port # - 53

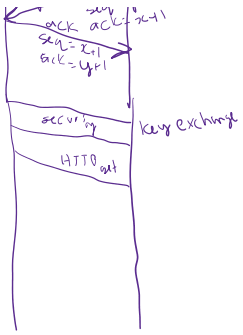
RIP determines path intra domain

BGP determines path inter domain

A B



mac



Hashing
non repudiation
authentication
integrity

mac
integrity
authentication
shared key
non repudiation X
could be
A or B

sample_paper_1

Sunday, November 26, 2023 2:10 AM

sample_paper_1

NATIONAL UNIVERSITY OF SINGAPORE

CS2105 – INTRODUCTION TO COMPUTER NETWORKS

Sample Exam Paper 1

Please DO NOT upload questions and answers onto the Internet.

Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains 7 questions and comprises 8 printed pages, including this page.
2. This is a **CLOSED BOOK** assessment. You may bring in one piece A4 size help sheet.
3. Calculators are allowed, but not laptops, PDAs, or other electronic devices.

Q1. Multiple Choice Questions (MCQs)

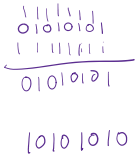
1.1 Which of the following protocols run at the application layer?

i. HTTP ✓
ii. UDP
iii. DHCP ✓
iv. DNS ✓

A. (i) and (iii) only
B. (i) and (iv) only
C. (i), (ii) and (iv) only
D. (i), (iii) and (iv) only
E. None of the above

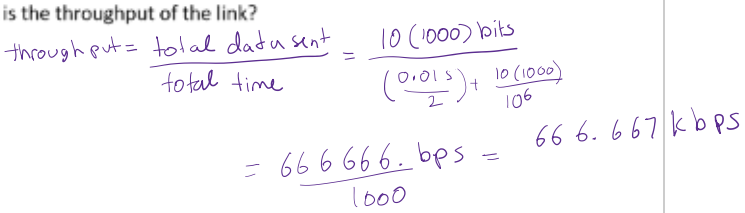
1.2 1s complement is used as checksum in _____. Given two bytes 01010101 and 11111111, the 1s complement checksum is _____.

A. TCP but not UDP, 10101010
B. UDP but not TCP, 10101010
C. Both TCP and UDP, 10101010
D. Both TCP and UDP, 01010101
E. None of the above



1.3 10 packets are continuously sent over a 1 Mbps link. Each packet is of 1,000 bits long and RTT is 10 ms. What is the throughput of the link?

A. 511.856 bps
B. 511.856 Kbps
C. 500 bps
D. 500 Kbps
E. 666.667 Kbps



1.4 If the baud rate for n-PSK signal is 1000 and the bit rate is 5000, what is n?

A. 5
B. 4
C. 32
D. 2
E. None of the above

1.5 Which of the following statement about IP datagram is FALSE?

A. Routing protocols determine the routes that datagrams take between sources and destinations.

B. TTL field of IP header prevents a datagram from circulating in the network forever.

C. When a big datagram is fragmented into a series of smaller fragments, transport layer header will be replicated in each fragment.

D. On the Internet, datagrams from the same source may take different routes towards the destination.

E. MTU of the link-layer protocol places a limit on the length of a datagram.

1.6 In a subnet, the first IP address is 172.18.176.0 and the last IP address is 172.18.183.255. What is the length of network prefix of this subnet?

A. 28

B. 29

C. 21

D. 22

E. None of the above

1.7 A host uses a variety of protocols to discover information about the network it is connected to. Which of the following statements is FALSE?

A. To perform a DNS lookup, a host must first discover the IP address of its local DNS server using DHCP.

B. To send a packet outside the host's subnet, the host must first discover the IP address of its first-hop router using DHCP.

C. To send a packet outside the host's subnet, a host must first discover the IP address of the destination host using DNS.

D. To get an IP address assigned, a host must first discover the IP address of its DHCP server using DNS.

E. To send a packet to another host in the same subnet, a host must first discover the MAC address of the destination host using ARP.

1.8 An IP address block 192.168.208/20 can be further divided into x subnets, each supporting a maximum of y hosts. Which of the following is NOT a valid assignment?

A. $x = 4$ and $y = 1022$

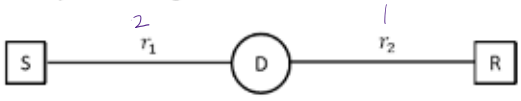
B. $x = 32$ and $y = 126$

C. $x = 64$ and $y = 62$

D. $x = 256$ and $y = 30$

E. $x = 1024$ and $y = 2$

1.9 A device (D) is used to connect a sender (S) and a receiver (R). Transmission rates of the links between sender and the device and between the device and receiver are r_1 and r_2 ($r_1 > r_2$) respectively. Ignore other types of delay, what is the end-to-end delay to send a packet of length L?



D R
at t= 2 .
3 . .
4 .

A. $\frac{Lr_1r_2}{r_1+r_2}$, if this device is a store-and-forward packet switch.

B. $\frac{L}{2r_1}+\frac{L}{2r_2}$, if this device is a store-and-forward packet switch.

C. $\frac{L(r_1+r_2)}{r_1r_2}$, if this device acts on individual bits and repeats every bit to receiver once receives it from sender.

D. $\frac{L}{r_1}+\frac{1}{r_2}$, if this device acts on individual bits and repeats every bit to receiver once receives it from sender.

E. $\frac{1}{r_1}+\frac{L}{r_2}$, if this device acts on individual bits and repeats every bit to receiver once receives it from sender.

Q2.

Suppose there is a 10 Mbps microwave link between a geostationary satellite and its base station on Earth, which are 3.6×10^7 meters apart. The satellite takes a digital photo once in a while and then sends it to the base station. Assume a propagation speed of 2.4×10^8 meters/second.

(a) What is the propagation delay (in seconds) of the link?

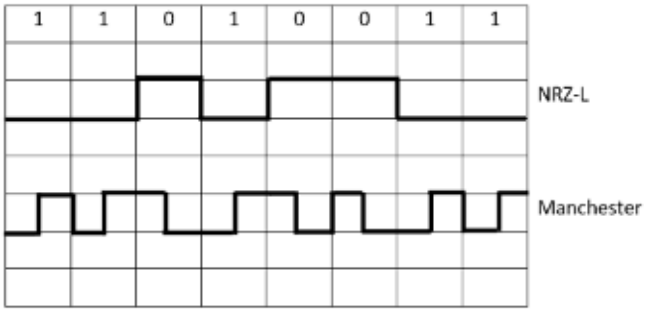
(b) Suppose the satellite takes a photo every 24 seconds and let x denote the size of the photo. What is the minimum value of x (in bits) for the microwave link to be fully utilized (i.e. always busy transmitting)?

a. $\frac{3.6\times10^7}{2.4\times10^8} = 0.15\text{sec}$

b. $\frac{L}{10^5} = 24$

Q3.

(a) The correct drawing of NRZ-L for bit pattern 11010011 is shown in the grid below. Is corresponding Manchester encoding correctly drawn? Answer "Yes" or "No".



(b) A channel has bandwidth in the range between 200 KHz - 260 KHz, and a signal to noise ratio of 31. What is the Shannon capacity of the channel?

(c) Suppose the propagation delay between furthest nodes is d and link rate is r. What is the minimal frame size L to ensure collision will always be detected in CSMA/CD protocol?

(d) Source and destination are connected by a single link that has packet loss probability of p. If at most k (re)transmissions are allowed until the source gives up, what is the probability that a packet would be successfully delivered to destination?

Q4.

On **Sunfire** server, we type the command

```
dig -t a www.duke.edu +trace
```

and observe the following outputs:

```
; <<>> DiG 9.6-ESV-R8 <<>> www.duke.edu +trace
;; global options: +cmd
.                   155852  IN      NS      b.root-servers.net.
.                   155852  IN      NS      c.root-servers.net.
.                   155852  IN      NS      d.root-servers.net.
.                   155852  IN      NS      e.root-servers.net.
.                   155852  IN      NS      f.root-servers.net.
.                   155852  IN      NS      g.root-servers.net.
.                   155852  IN      NS      h.root-servers.net.
.                   155852  IN      NS      i.root-servers.net.
.                   155852  IN      NS      j.root-servers.net.
.                   155852  IN      NS      k.root-servers.net.
.                   155852  IN      NS      l.root-servers.net.
.                   155852  IN      NS      m.root-servers.net.
.                   155852  IN      NS      a.root-servers.net.
;; Received 492 bytes from 137.132.85.2#53(137.132.85.2) in 13 ms

edu.                172800  IN      NS      a.edu-servers.net.
edu.                172800  IN      NS      f.edu-servers.net.
edu.                172800  IN      NS      d.edu-servers.net.
edu.                172800  IN      NS      c.edu-servers.net.
edu.                172800  IN      NS      g.edu-servers.net.
edu.                172800  IN      NS      l.edu-servers.net.
;; Received 265 bytes from 192.33.4.12#53(192.33.4.12) in 182 ms

duke.edu.           172800  IN      NS      avallone.stanford.edu.
duke.edu.           172800  IN      NS      dns-auth-01.oit.duke.edu.
duke.edu.           172800  IN      NS      dns-auth-02.oit.duke.edu.
;; Received 194 bytes from 192.31.80.30#53(192.31.80.30) in 259 ms
```

```
www.duke.edu.      21600   IN      CNAME   duke.edu.
duke.edu.          21600   IN      A       54.191.241.8
duke.edu.          21600   IN      A       54.68.155.51
duke.edu.          21600   IN      NS      dns-auth-02.oit.duke.edu.
duke.edu.          21600   IN      NS      dns-auth-01.oit.duke.edu.
;; Received 164 bytes from 152.3.105.232#53(152.3.105.232) in 270 ms
```

Answer the following questions.

(a) Write down one IP address of a local DNS server. 137.182.85.2

(b) Write down one IP address of a root DNS server. 192.33.4.12

(c) Write down one IP address of a top-level domain DNS server. 192.31.80.30

(d) Write down one IP address of `www.duke.edu`. 54.191.241.8

(e) What is the canonical name of `www.duke.edu`? duke.edu

(f) What port number does a DNS server listen to? 53

Q5.

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

Prefix Match	Interface
11	3
101	4
100	1
1101	2
otherwise	0

$(4-1)\cdot 2^4 = 48$

32

32

16

128

For each of the five interfaces, give the associated range of destination IP addresses and the number of destination IP addresses in that range.

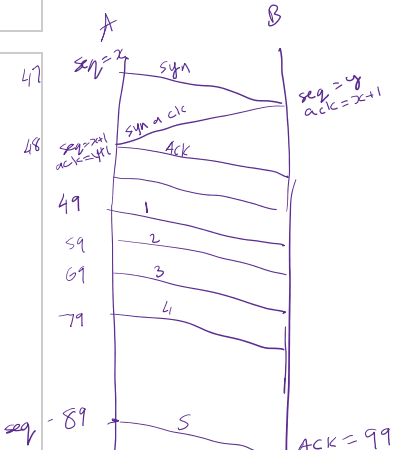
Q6.

Host A sends 5 data packets to host B using TCP protocol. Each data packet contains 10 bytes of application data.

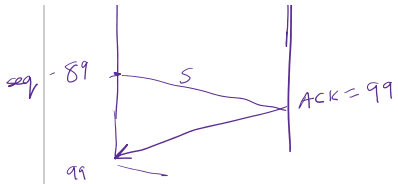
Answer the following 3 questions. They are independent of each other.

(a) Suppose 5 data packets arrive in order and all are accepted by B. The last ACK packet sent by B has ACK number 99. What is the sequence number of the first data packet sent by A? 49

(b) Suppose 5 data packets arrive at B out of order. Their sequence numbers (shown in the



sent by B has ACK number 99. What is the sequence number of the first data packet sent by A? 49



(b) Suppose 5 data packets arrive at B out of order. Their sequence numbers (shown in the order of arrival) are 200, 240, 210, 230 and 220 respectively. Assume B will buffer out-of-order packets for later in-order delivery to application. Write down ACK numbers of the corresponding ACK packets sent by B. 210 210 220 220 250

(c) Within 100 ms duration, B receives 5 in-order data packets and accepts all of them. How many ACK packets will B send out?

Q7.

10 students want to communicate with each other confidentially (i.e., messages between any two students shouldn't be understandable to a third student).

Answer the following 3 questions. They are independent of each other.

(a) In symmetric key cryptography, how many secret keys are needed in total? $\binom{10}{2} = 45$

(b) Suppose every student trusts the teacher. If a student needs to send a message to another, he first sends it to the teacher; the teacher then sends the message to the other student. The teacher is allowed to understand all messages sent to her. At a minimum how many keys are needed in total? State clearly if symmetric or public key cryptography is used.

10 key symmetric

(c) Suppose every student has a pair of public/private keys, so does the teacher. Now the teacher has a short announcement for all the students. In no more than 80 words, write down the steps the teacher performs to ensure confidentiality and authenticity of this announcement.

To ensure confidentiality
if Stu A send to Stu B
encrypt with B's public key
so only B can decrypt

To ensure authenticity
A will also encrypt m wit
A's private key and send it
also
B can use A's public key to
decrypt & match m

=== END OF PAPER ===

sample_paper_2

Sunday, November 26, 2023 11:10 PM

sample_paper_2

NATIONAL UNIVERSITY OF SINGAPORE

CS2105 – INTRODUCTION TO COMPUTER NETWORKS

Sample Exam Paper 2

Please DO NOT upload questions and answers onto the Internet.

Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains 7 questions and comprises 7 printed pages, including this page.
2. This is a **CLOSED BOOK** assessment. You may bring in one piece A4 size help sheet.
3. Calculators are allowed, but not laptops, PDAs, or other electronic devices.

Q1. Multiple Choice Questions (MCQs)

1.1 CSMA/CD is a (an) _____ layer protocol.

A. application
B. transport
C. network
D. link
E. physical

1.2 Ethernet provides an unreliable service. Therefore,

A. CRC is not used for error checking.
B. Ethernet sends a negative acknowledgement to the sender to indicate packet loss.
C. Ethernet drops a frame that fails error checking without retransmission.
D. Ethernet does not function correctly when bit errors in frames are detected.
E. Applications that require reliable delivery cannot run over Ethernet.

1.3 Which of the following is an INVALID subnet mask?

A. 255.255.255.0
B. 255.255.252.0
C. 255.192.0.0
D. 255.255.255.224
E. 255.255.244.0

1.4 Which of the following IP addresses belong to the subnet 137.132.96/20?

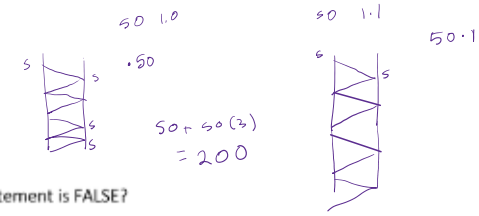
i. 137.132.96.96
ii. 137.132.104.104
iii. 137.132.112.112
iv. 137.132.120.120

A. (i) only
B. (i) and (ii) only
C. (i), (ii) and (iii) only
D. (iii) and (iv) only
E. (i), (ii), (iii) and (iv) only

1.5 A Web server supports both HTTP/1.0 and HTTP/1.1. So far 100 clients have downloaded a web page from the server, which contains 1 HTML file and 2 images. Half of the clients run HTTP/1.0 and the other half run HTTP/1.1.

How many sockets has the Web server ever created?

A. 201
B. 200
C. 100
D. 101
E. None of the above

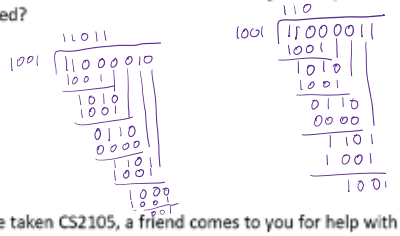


1.6 Which of the following statement is FALSE?

A. When a router receives an IP datagram with destination address 255.255.255.255, it must broadcast this IP datagram on all the interfaces except the interface this datagram is received.
B. One of the benefits of segmenting a big chunk of data into smaller packets for transmission in the Internet is to lower end-to-end delay.
C. MSS specifies the maximum size of a TCP segment, exclusive of the size of TCP header.
D. Hosts in the same subnet communicate with each other without intervening a router.
E. The maximum size of an IP datagram that can be transmitted over a link is restricted by the link MTU.

1.7 Given the CRC generator **1001**, which of the following bit sequence received by receiver is not corrupted?

A. 11000010
B. 11000111
C. 11000110
D. 11000011
E. 11000001



1.8 Knowing that you have taken CS2105, a friend comes to you for help with his laptop. He says that he cannot access the Web page hosted at **`www.example.com`**. Using the tools you have learned in CS2105, you run the following commands on his laptop to troubleshoot what could be the reason.

Which of the following is NOT the correct use of the corresponding tool?

A. You run **`telnet`** to check if **`www.example.com`** is listening on port 80.
B. You run **`traceroute`** to check if there is a route from the laptop to **`www.example.com`**.

C. You run **dig** to check if his DNS server is able to resolve the IP address of host name **www.example.com**.
D. You run **ping** to check if you can establish a TCP connection to **www.example.com**.
E. You run **curl** to check if **www.example.com** is responding to a HTTP request correctly.

1.9 A Go-back-N sender just receives an ACK packet with sequence number t. Before this ACK is received, sender's window is $[k, k+N-1]$ where N is the window size. Suppose $k > N$, packets may be lost or corrupted but will not be reordered. What is the smallest possible value of t?

A. $k-1$
B. k
C. $k-N+1$
D. $k-N$
E. None of the above

Q2.

(a) After issuing the command **nslookup www.yahoo.com**, the following output is observed.

```
Server:         203.211.152.66
Address:        203.211.152.66#53

Non-authoritative answer:
www.yahoo.com       canonical name = fd-fp3.wg1.b.yahoo.com.
Name: fd-fp3.wg1.b.yahoo.com
Address: 106.10.138.240
Name: fd-fp3.wg1.b.yahoo.com
Address: 106.10.139.246
```

What are the IP address and port number of the DNS server that answers this DNS query?

(b) The _____ field in IP header prevents an IP datagram from continuously wandering through the Internet.

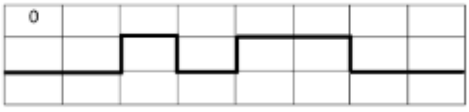
(c) A subnet has 16384 IP addresses and one of the IP is 58.26.177.105. What is the first IP address in this subnet?

Q3.

(a) A modem is designed for use over a telephone link, for which the available channel bandwidth is 3 kHz, and the average signal to noise ratio on the channel is 511. What is the maximum error-free data rate that can be supported on this channel?

(b) 1.8 Mb of data is transmitted in 60 seconds using 8-PSK. What is the baud rate of the signal?

(c) What bit pattern does the following NRZ-I diagram represent? Suppose the first bit is 0.



400000 bits 920 bit/pkt
= 435 pkts

total bits = 435·80 + 400k = 434 800
last pkt = 808

Q4.

10^6 bps

Two hosts A and B are connected via a router. The link rate is 1 Mbps and propagation delay is 40 ms per link. The maximum size of a packet is 1 Kb and packet header is 80 bits. Suppose sender sends as much data as possible in a packet, packets are sent continuously and no packet is corrupted or lost during transmission.

How long (in milliseconds) does it take to send a 400 Kb file from A to B (from when the first bit of the first packet leaves A to when last bit of the last packet arrives at B)?

A ——○—— B

$\frac{1000}{10^3} + 40 + \frac{434800}{10^3} + 40 = 515.8$ ms

Q5.

Public key cryptography uses both public and private keys. Let Alice's public key be K_A^+ and private key be K_A^-, Bob's public key be K_B^+ and private key be K_B^-. Alice sends a message m to Bob. Describe how they can ensure message confidentiality and message authenticity using only these 4 keys.

Alice → Bob

$K_B^+(m, K_A^-(m))$

Bob decrypt msg

$K_B^-[K_A^-(m, K_A^-(m))]$

get $(m, K_A^-(m))$

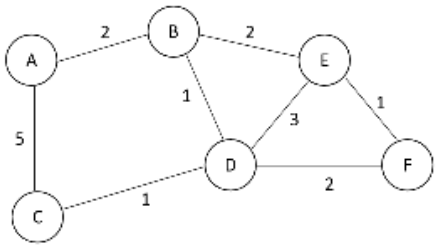
now use K_A^+

$K_A^+(K_A^-(m)) \Rightarrow m_1$

check $m = m_1$

Q6.

Consider the network topology shown below. Each link is labelled with the cost (in dollars) of using that link. Every router runs distance vector routing protocol.



	A	B	C	D	E	F
A	0	2	~~5~~4	3	4	5
B	2	0	2	1	2	3
C	~~5~~4	2	0	1	4	3

(a) Fill in the distance vector table for the initial distance vectors of routers A to C (i.e. before the distance vector protocol is executed). If a router is unaware of another router, write '–' in the corresponding slot.

(b) Suppose the distance vector protocol has terminated and each router knows the cost of the least cost path to every other router. Fill in the distance vector table for the final distance vectors of routers A to C.

(c) Routers C to F are the gateway routers to the following subnets.

Subnets	Gateway routers	Next Hop
137.132.58.128/28	D	B
137.132.89.0/26	C, D	B
137.132.80.128/25	C, F	B
137.132.82.0/24	E	B

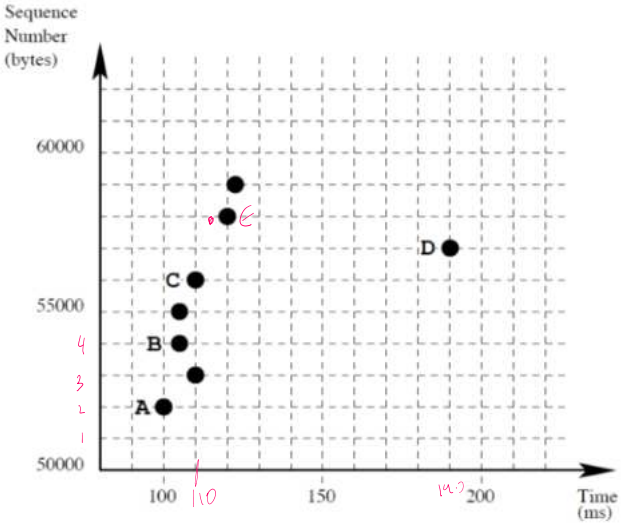
Use the result you get in (b) to derive the forwarding table of router A.

(d) Assume all the links have the same transmission rate. In no more than 50 words, explain why the forwarding table in (c) leads to inefficient use of network bandwidth?

A always moves thru B to get to gateway routers whereas A–C link is underused

Q7.

The following graph shows the time sequence graph for a TCP connection between host X and host Y. Each dot represents a TCP segment received at host Y, plotting the sequence number of the segment, versus the time at which it is received. A set of dots stacked above each other represents a series of packets that are received back-to-back by the receiver. The packet labelled with A is the first data packet sent by X. The packet labelled with D is a re-transmitted packet.



(a) How many bytes of data are there in each TCP segment? 1000

(b) Suppose an acknowledgment is sent by Y at time 105ms, after receiving the packet labelled with B. What should be the acknowledgement number in this feedback packet? 53000

(c) Does Y buffer out-of-order packets or discard them? Justify your answer in no more than 100 words. buffer — example, pkt with seq # = 53000 is out of order. if Y did not buffer pkt D would not be retransmit cuz first 53000 seq would need to be retransmit cuz TCP sender has only 1 timer and resends oldest unacked pkt

(d) Suppose X sets a reasonable timeout value for this TCP connection. Retransmission is always successful. Estimate the timeout value that X chooses. Justify your answer and state clearly any assumptions you make. 80

=== END OF PAPER ===

cack timeout resend
d + 2d → d

4d = 80
d = 20
timeout ~ 40 sec

sample_paper_3

Monday, November 27, 2023 11:40 PM

sample_paper_3

NATIONAL UNIVERSITY OF SINGAPORE

CS2105 – INTRODUCTION TO COMPUTER NETWORKS

Sample Exam Paper 3

Please DO NOT upload questions and answers onto the Internet.

Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains 7 questions and comprises 7 printed pages, including this page.
2. This is a **CLOSED BOOK** assessment. You may bring in one piece A4 size help sheet.
3. Calculators are allowed, but not laptops, PDAs, or other electronic devices.

Q1. Multiple Choice Questions (MCQs)

1.1 Which of the following is NOT a network protocol?

A. RIP
B. ICMP
C. ARP
D. PAP
E. CSMA

1.2 Which of the following command will NOT cause a DNS query to be issued?

A. `dig www.comp.nus.edu.sg`
B. `nslookup www.nus.edu.sg`
C. `telnet localhost 9000`
D. `ping sunfire.comp.nus.edu.sg`
E. `traceroute ivle.nus.edu.sg`

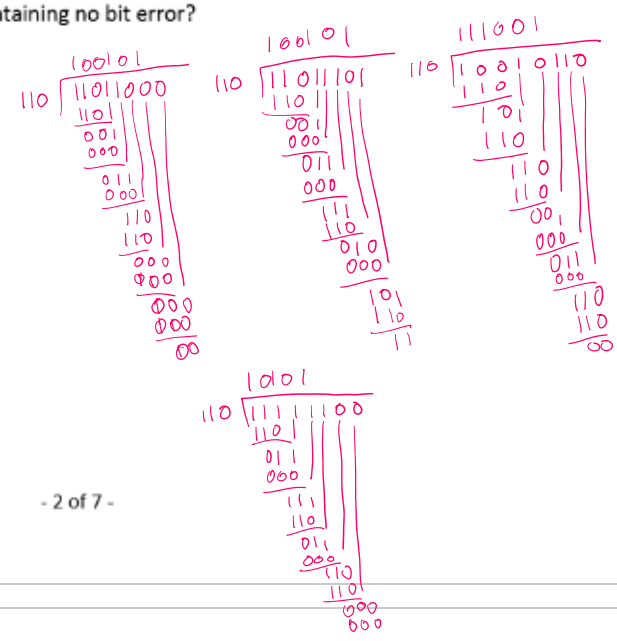
1.3 Which of the following statement about 2-dimensional parity bits is FALSE?

A. It can detect any one-bit error.
B. It can correct any one-bit error.
C. It can detect any two-bit error.
D. It can correct any two-bit error.
E. It may not be able to detect a four-bit error.

1.4 Two hosts are communicating using CRC as an error detection scheme, with a generator of 110. Every byte sent consists of six bits of data and two bits of the CRC value. Suppose the following four bytes are received. Which bytes would pass the CRC test and considered as containing no bit error?

i. 11011000
ii. 11011101
iii. 10010110
iv. 11111100

A. (i) and (ii) only
B. (i) and (iv) only
C. (i), (iii) and (iv) only
D. (iii) and (iv) only
E. (i), (ii) and (iii) only



1.5 Which of the following statement about IP header is TRUE?

A. The source and destination port numbers in the IP header determine which application on the receiving host will process the datagram.

1.5 Which of the following statement about IP header is TRUE?

A. The source and destination port numbers in the IP header determine which application on the receiving host will process the datagram.

B. The TTL field in the IP header determines the time period within which the source IP address is valid.

C. The 16-bit identifier field in the IP header are not changed during IP fragmentation.

D. The checksum field in the IP header allows the receiver to check if the IP header is corrupted.

E. The protocol field in the IP header determines which link layer protocol should be used to transmit the datagram.

1.6 Consider a noisy channel with a Shannon capacity of 100 kbps and a bandwidth of 10 kHz. The signal-to-noise ratio of this channel is

A. 3.3

B. 5

C. 9

D. 10

E. 1023

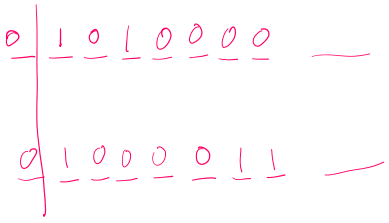
1.7 A subnet contains two hosts with IP addresses 137.132.80.16 and 137.132.67.94 respectively. Which of the following is/are possible address block assigned to the subnet?

i. 137.132.64.0/18

ii. 137.132.64.0/19

iii. 137.132.64.0/20

iv. 137.132.0.0/17



A. (i) only

B. (i) and (ii) only

C. (i), (ii) and (iii) only

D. (iii) and (iv) only

E. (i), (ii) and (iv) only

1.8 Which of the following digital-to-analogy modulation scheme can support the highest data rate?

A. PSK at 8000 baud
B. QPSK at 8000 baud
C. 4-QAM at 6000 baud
D. 8-QAM at 4000 baud
E. 16-QAM at 2000 baud

1.9 Consider the following Java implementation of the rdt 3.0 protocol, using classes and methods similar to your Assignment 2. A student has implemented the sender correctly according to the state diagram of the protocol. In the receiver, the student implemented the following:

```
byte[] recv() throws Exception {
   DataPacket p = udt.recv();
   while (p.isCorrupted || p.seq != seq) {
      p = udt.recv();
   }
   udt.send(new AckPacket(p.seq));
   seq = 1 - seq;
   return deliverData(p);
}
```

Here, **seq** is the expected sequence number (either 0 or 1) at the receiver, and **deliverData** is a method that extracts and returns the payload from packet **p**.

The protocol is used over a channel that may lose or corrupt a packet, but always delivers packets in the order that they are sent.

We say that the receiver *waits forever*, if it is blocked at the call **udt.recv()**, waiting to receive a packet that will never be sent. We say that the sender *loops forever*, if it repeatedly retransmits the same packet over and over again.

Which of the following statement CORRECTLY describes the behavior of the protocol implemented above?

A. A single corrupted data packet is sufficient to cause the sender to loop forever.
B. A single corrupted data packet is sufficient to cause the receiver to wait forever.
C. A single loss ACK packet is sufficient to cause the sender to loop forever.
D. A single loss ACK packet is sufficient to cause the receiver to wait forever.
E. A single premature timeout is sufficient to cause the sender to loop forever.

Q2.

Two hosts A and B are communicating over a wireless channel with a signal to noise ratio of 15 and a bandwidth of 100 MHz. The nodes are 300 meters apart. The signal propagation speed over the air is the $3 * 10^8$ m/s.

(a) What is the maximum data rate that can be supported by the wireless channel?

(b) Suppose that A transmits at 20 MBaud using 64-QAM as the modulation scheme. What is the transmission rate of A in Mbps?

(c) Suppose that A transmits a frame of size 1000 bytes at 100 Mbps, starting at time t = 0. At what time will the frame reach B completely? Give your answer in the unit of μs (Note: $1\ \mu s = 1 * 10^{-6}$ s).

Q3.

Two hosts A and B are 2000 km apart and are connected directly using a link with propagation delay of 800 bit times and propagation speed of $2.5 * 10^8$ m/s. A is sending a sequence of packets, each is 100 bytes in size, to B.

pkt = 100 bytes = 800 bits

$800 + \frac{2000 \cdot 1000}{2.5 \cdot 10^8 m/s}$

0.008

(a) How long does it take for B to receive a packet?

(b) A is using a sliding window protocol to communicate with B. What is the minimum window size A should use for the link to be fully utilized?

3

Q4.

To preserve message confidentiality and authenticity, the following information is contained in a secured message sent from Alice to Bob.

- Encrypted hash of the message encrypt with Alice's privak key, digital signature
- Encrypted message encrypt with session key
- Encrypted session key encrypt with Bob public key, to share session key used going forward

Briefly describe the purpose of each piece of information and the key used in generating that information.

Q5.

Figure 1 shows the finite state machine of a protocol designed to run over a channel with the following properties: (P_1) can corrupt packet, (P_2) can lose packets, and (P_3) has an unknown round trip time.

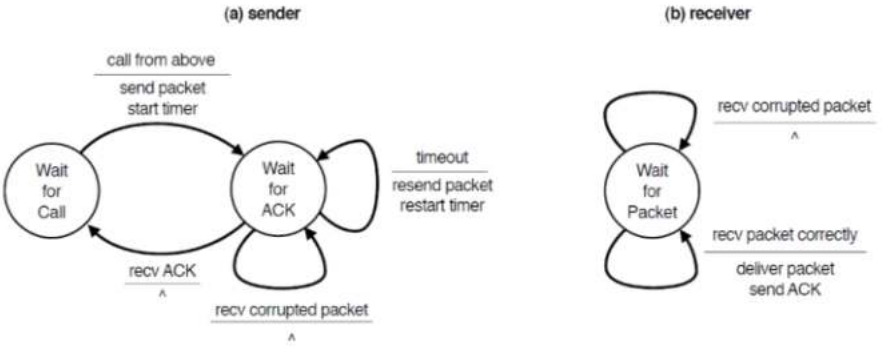


Figure 1: Finite State Machine of a Protocol

yes timeout.

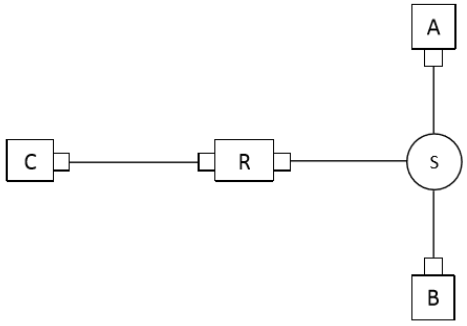
(a) Is it possible for this protocol to deliver the same packet twice to the application? Either give an example where the same packet is delivered twice by drawing a timing diagram, or argue why every packet will only be delivered once.

(b) Is it possible for this protocol to not detect a lost data packet? Either give an example where a lost packet is not detected by drawing a timing diagram or argue why a packet loss is always detected.

(c) Can we remove only one of the network properties P1, P2, P3 so that the protocol works as intended without modification? Justify your answer.

Q6.

The diagram below shows a small network with five entities: hosts A and B are connected to a router R through a switch S. Host C connects to R directly. There is no other host, switch, or router in the network.



(a) What is the maximum number of entries that could be in the switching table of S? 3

(b) What is the maximum number of entries that could be in the ARP table of A? 2

(c) What is the maximum number of entries that could be in the ARP table of C? 1

(d) How many IP addresses are used in this network? 5

Q7.

A node x is part of a network running distance vector routing protocol. x has three entries in its routing table:

Destination	Cost	Next Hop
w	4	w
y	α	z
z	β 10	w

α and β are two unknown values (unknown to you, but known to x). Assume that the distance vector routing protocol has converged and the minimum cost from x to every other node has been found. We denote $c(x, y)$ as the link cost between x and y, and $d_x(y)$ as the cost of the minimum cost path from x to y. The link cost is a positive integer.

We know that $c(x, w)$ is 4, and $c(x, z)$ is 10.

(a) What is the minimum possible value for α? 11

(b) What is the value for $d_w(z)$? 6

=== END OF PAPER ===

1819SEM1-CS2105(Ques 9-14)

Tuesday, November 28, 2023 5:28 PM

1819SEM1-CS2105(Q...

NATIONAL UNIVERSITY OF SINGAPORE

CS2105 — INTRODUCTION TO COMPUTER NETWORKS

Semester 1, 2018/2109

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. Please write your Student Number only. Do not write your name.
2. The assessment has one question booklet and one answer booklet.
3. The question booklet contains **FOURTEEN (14) questions** and comprises **EIGHT (8) pages** including this cover page.
4. The answer booklet contains **FOUR (4)** pages.
5. Weightage of questions is given in square brackets. The maximum attainable score is 50.
6. This is a **CLOSED** book assessment, but you are allowed to bring **ONE (1)** double-sided A4-size, sheet of notes.
7. The use of electronic calculators is permitted for this assessment.
8. Write all your answers legibly in the **ANSWER BOOKLET**.

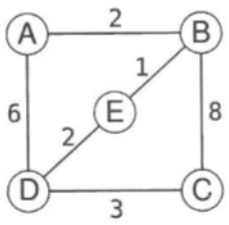
This page is intentionally left blank.

It may be used as scratch paper.

Short Answer Questions [Total: 34 marks]

Short Answer Questions [Total: 34 marks]

9. **[Total: 6 marks]** Consider the following network topology where each router runs the distance vector algorithm with ***poisoned reverse***:



A B C D E
A 0 2 6
B
C
D
E

(a) [2 marks] Show the contents of router *A*'s distance vector table before any distance vector exchange takes place.

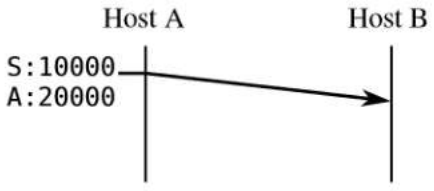
(b) [2 marks] All routers exchange distance vectors with their neighbours before updating their tables. Show the contents of A's distance vector table after the exchange and update takes place.

(c) [2 marks] At this point, router A exchanges its distance vector with its neighbours. Show the contents of the distance vector that A exchanges with B.

10. **[Total: 6 marks]** A TCP connection has been established between two hosts *A* and *B*. The MSS of the connection allows each TCP packet to contain at most 1,000 bytes of application data. Suppose at this point in time, both hosts each has 3,000 bytes of data to send to the other host and it was done by sending the least number of segments.

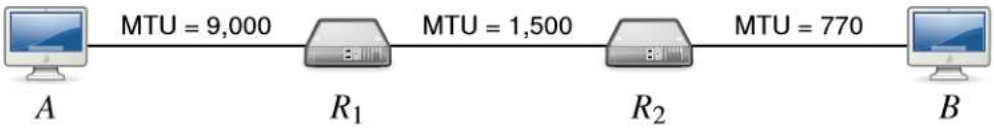
(a) [2 marks] What is the total number of TCP segments that was sent by both hosts?

(b) [4 marks] Suppose the first segment was sent by host A and has sequence number 10,000 and acknowledgement number 20,000 as shown. Complete the trace of the conversation by showing the sequence and acknowledgement number of all the segments sent and received between *A* and *B*.



11. **[Total: 6 marks]** Two hosts are separated by two routers with links of different MTU size as shown below. Host *A* sends a non-fragmented IPv4 datagram of Id 54 and length 3,000 bytes (inclusive of the 20 bytes IP header) to host *B*. Assume there are no data corruption or packet losses.

(inclusive of the 20 bytes IP header) to host B. Assume there are no data corruption or packet losses.

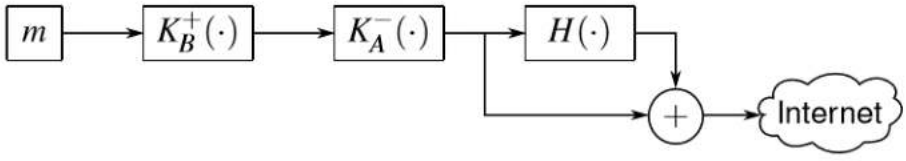


(a) [2 marks] List the IP fragments along with the information in their header sent by R_1 in increasing order of offset.

(b) [2 marks] List the IP fragments along with the information in their header sent by R_2 in increasing order of offset.

(c) [2 marks] Many Gigabit Ethernet networks support *jumbo frames* where the MTU can be as large as 9,000 bytes even though the IEEE 802.3 Ethernet only mandates that devices support an MTU of up to 1,500 bytes. What is an advantage of doing so?

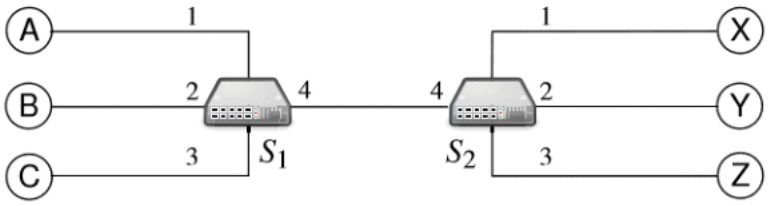
12. **[Total: 6 marks]** Alice is sending a message m to Bob using public key cryptography. Let Alice's public and private key pair be K_A^+ and K_A^-, and Bob's public and private key pair be K_B^+ and K_B^-. Assume that all the public keys as well as the hash algorithm H are known by everyone including Trudy the intruder. Suppose Alice is sending the message m using the protocol shown below.



(a) [2 marks] Show what Bob needs to do to read the message m.

(b) [4 marks] Does the protocol provide confidentiality, integrity, authenticity, and non-repudiation? For each of those properties, explain your answer.

13. **[Total: 6 marks]** Consider the network below with two switches S_1 and S_2 where both switches currently have empty switch tables. We denote the MAC address of a node with a subscript. For instance, MAC address of B is MAC_B.



(a) [2 marks] Node A wants to communicate to node Z, but having just joined the network, it does not know the MAC address of Z (i.e., MAC_Z). What kind of message should node A send? State the source and destination MAC address (i.e., MAC_{src} and MAC_{dst}) of the message.

(b) [2 marks] What will be the state of the switch table for both S_1 and S_2 **after** node Z replies node A its MAC address?

(c) [2 marks] At this point, switch S_1 is restarted and loses all data in its switch table while switch S_2 retains its state given in part (b). Node B now sends a frame with a source address of MAC_B and destination addresss of MAC_Z. Which nodes will receive this frame? Note that a node may receive a frame but ignore it if the destination address does not match its own.

14. **[Total: 4 marks]** Two stations, A and B, connected in a bus topology. They are separated 1,000 m away with the propagation speed of 2×10^8 m/s and transmission rate of 10 Gbps. Assume that they are using CSMA/CD. Consider the scenario where station A starts transmitting at time $t = 0$ s.

(a) [2 marks] At what time does station B receives the first bit?

(b) [2 marks] Assume that the time you computed in part (a) is time t_1. Consider the case that station B starts sending a frame just before time t_1 causing a collision. What should the minimum frame size be such that station A will detect the collision?

— E N D O F P A P E R —

Scratch Paper

– H A P P Y H O L I D A Y S ! –