CS2105 Notes.

Date No.

I Overview

Hosts

End systems running applications.

Packet

A package of messages sent between hosts

Store-And-Forward

A packet can only be sent to the next packet switch after it is fully received. Each packet is transmitted at full link capcity (link transmission rate or bandwidth)

Routers VS. Switches

Routers are often at the network core (with high traffic) while switches are often at the network access (boundaries).

Data Centers.

A concentration of physical serves that aggregate as a (logical) server (primarily to handle high traffic).

Connecting End Systems with Edge Routers

① Residential Networks.
② Institional Networks.
③ Mobile Networks

Delays

① Transmission delay

How fast the link can eat/handle.

$$d_{trans} = \frac{L}{R} \quad \frac{bits}{bps}.$$

$$\text{End-to-end delay} = \frac{2L}{R}.$$

② Propagation delay
How fast the packet can travel (capped by the light speed).

$$d_{prop} = \frac{d}{s} \quad \begin{matrix} \text{distance} \\ \text{speed} \end{matrix}$$

③ (Nodal) Processing
Overheads incurred upon receipt, including checksum, destination (forwarding table) lookup.

$$d_{proc} < ms$$

④ Queuing Delay.
Time spent waiting for turn to be sent out.
Depends on the traffic.

The total time needed for Nodal Processing.

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

Routing
Determines source-destination route to be taken by packets.

Circuit-Switching
Each link has N circuits exclusively for a pair of end systems (src, dest). Performance is guaranteed since there is no sharing.

Internet Service Provider (ISP)
Tier I: National/international coverage Tier II: Regional
Internet Structure: Network of Networks
ISP of both tiers.
IXP (Internet Exchange Point) between ISPs and content providers' private networks. Sometimes, a peering link communicates two ISPs.

Date No.

Throughput

Usually measured in bps, the rate of bits actually transferred between 2 end systems.

Network Protocols

Define format, order of messages sent and received among network entities, and actions to be taken on message transmission, receipt.

Layers:

① Application: Supporting network applications
FTP, SMTP, HTTP.

② Transport: Process-Process data transfer
TCP, UDP.

③ Network: Routing of datagrams from source to destination
IP, routing protocols.

④ Link: Data transfer between neighbouring network elements.
Ethernet, 802.11 (Wifi), PPP.

⑤ Physical.

'traceroute'

- 3 probes each element along the path from src to dest to know how delays are distributed along that path.
- The 3 packets may be sent to different routers.
- * * * n indicates no responses or refusal to respond within TTL. for n packets.

Date No.

II Application Layer.

Server-Client Architecture

Server: Always-on host. Well-known, permanent IP.
Client: Might be intermittently connected and may have dynamic IP. Doesn't communicate with each other.

P2P Architecture

Self-scalability: New peers bring in new capacity as well as new demands.
May intermittently connect and change IP.

Server Process & Client Process.

Who waits for connection invitation & who initiate a connection. (Can exist in P2P as well).

Addressing Process/Thread.

Identifier = IP + Port number (a 16-bit int).
Typically, one thread for a port number.

TCP Properties.

- Reliable Data Transfer (RDT).
- Flow control
- Congestion control
- Has connection

UDP Properties:

- Unreliable Data Transfer (UDT).

No guarantee of timing, min throughput and security

Url = Host Name + Path Name
www.sample.edu/someDept/pic.gif

HTTP: HyperText Transfer Protocal

An App-layer protocol for Client-server architecture.
Uses TCP (has connection) (at port 80 for webs).

Date No.

Stateless: servers know nothing about past requests.

Both use TCP.

Non-Persistent HTTP: (v1.0 only has this).
At most one object sent over a TCP connection
(At least 2x RTT for an object: including the RTT for establishing connection.

Persistent HTTP: (Default in v1.1).
Multiple objects over a single connection.
Don't forget the first RTT to establish the connection!).

- When loading a website, one RTT (request) for getting the base html file, then one RTT (request) for each referenced objects.
- For HTTP, non-parallel and parallel flavours are offered.
- Aside from RTTs, also consider file transmission time.

? HTTP Methods.

① GET: Retrieve an object specified by a URI
② HEAD: Same as GET, but the response will contain the status line and the header section only.
③ POST: Submit data for processing
④ PUT: Update data. Need to include the entire body.
⑤ DELETE
⑥ CONNECT: Establishes a tunnel
⑦ OPTIONS: Describe communication options available.
⑧ TRACE: For debugging. Loop-back test.

NUS National University of Singapore

Date No.

? Methods for Form Uploading.
① POST: The submission is included in the message body.
② URL: Use GET to upload the object specified in the URL field of the request line. E.g.: https://www.example.com/res?monkey?banana.

HTTP Response Status Code.
200 OK
301 Moved permanently
400 Bad Request.
404 NOT FOUND
305 HTTP Version. Not Supported
304 Not Modified.

Cookies
① Cookie header line of the response message (initial cookie identifier created by server).
② Cookie header line of subsequent request messages (so that the server can identify).
③ Cookie file managed by browser (so it knows what to send next time).
④ Cookie and past behaviors in database at the Web.

Web Caches (Proxy Server).
A server that acts as a cache.
Object in cache → return.
Else → get from the original server.

Conditional GET.
The request message includes the last modification date of an object. If up-to-date, the server returns 304 with an empty message body.

Date No.

Domain-Name System (DNS). — An App-Layer Protocol
IP address ⟷ Domain name. (many-to-many)
Over UDP. (but also has TCP options)

Local DNS Name Server
"Default name server" that each ISP has one.
Acts as a proxy and may send outdated mapping.
Perform recursive query.

Root Name Servers (13 logical servers worldwide with many mirrors)
⇕
.com, .edu, .org Servers. Top-Level Domain/ Tier I
⇕
.yahoo.com, .amazon.com, poly.edu, umass.edu Servers Tier II.
(Could be authorative) DNS servers

DNS Query: (Port No. 53 at server) (nslookup) (dig -t)
Iterative: The local DNS server asks every server along a path from the root down.
Recursive: Each lower server asks its parent server. The queries can be cached in servers for some time-to-live (TTL).
※ Change of IP may not known until all TTL expire

UDP Sockets
- No handshaking.
- Data can be lost or received out of order
- Sender explicitly attaches dest IP and port number to every packet.
- Uses SOCK_DGRAM
- Uses clientSocket.recvfrom(bufsize)
Uses bind() for server to listen to a tuple (IP, port).

Date No.

TCP Sockets.

- Each server has a 'welcoming' socket that listens to any request for connection.
- Exclusive socket is created for a single client upon connected.
- Reliable, in-order byte-stream transfer (error-checked, continuous), there are no app message boundaries, and the app layer needs to figure them out.
- Different sockets do not imply different port numbers.
- Uses SOCK_STREAM
- Uses bind() and listen().
- Uses accept() to create exclusive sockets
- Uses recv(bufsize).

Demultiplexing:

TCP identifies the socket by (src IP, src port, dest IP, dest port).

UDP identifies the socket by only (dest IP, dest port).

TCP & UDP port number: 16-bit unsigned

III. Transport Layer

Transport VS. Network.

Transport: Logical communication between processes.

Network: Logical communication between hosts.

~~TCP~~ UDP checksum

including header = 0 ∴ checksum = 0 while computing it.

① Add all 16-bit integers. Wrap around when there is a carry at the MSD.

② Flip 0 and 1 (1's complement)

Date No.

Complications of RDT

Problem	Fix
① Bit flips in body	Retransmission and acknowledgement sequence number
② Bit flips in ACK Resent duplicates are not recognised	
③ NAK is unnecessary.	Repeat the last Ack as an NAK.
④ Packets might be lost.	Timeout and resend
a) Duplicate Ack (means the one just sent is corrupted).	Ignore duplicate ACK. Resend upon timeout anyway It is still a wait-and-stop protocol. The sender will ensure the current pkt is well received before sending another
⑤ Wait-and-Stop is very inefficient.	($U_{sender} = \frac{D_{trans}}{RTT + D_{trans}}$ neglecting time taken by receiver) Introduce pipelined protocols.
⑥ In pipelining, packets may be lost or received out-of-order	Go-back-N and selective repeat.

Go-Back-N.

- Sender maintains a single clock for the earliest unacknowledged packet. P. (the send base).
- Receiver sends the highest continuous sequence number every time it receives a packet.
- Receiver discards not-in-order packets.
- Sender resends a whole sliding window starting from P upon timeout.

- Receiver acknowledges duplicate packets.
- Sender ignores duplate acknowledgements.
- The sliding window moves as soon as another consecutive ACK comes. Moves = send new packets in the window.
-

Selective Repeat

- Receiver acknowledges individual correctly received pkt. and buffers out-of-order pkt for delivering in-order messages to the upper layer.
- Sender maintains a timer for each unacknowledged pkt. Upon timeout, resend.
- On receiver's side, duplicate (already buffered) pkt also need to be acknowledged. (the sender may not think the pkt is not acknowledged by the time it resends or the ACK might be lost as well.).

TCP

- Point-to-point.
- Has handshaking.
- Full duplex data (bi-directional data flow)
- In-order byte-stream (no app-level message boundaries).
- Pipelined (dynamic window size set by congestion/flow control).
- Has 2 buffers, one at sender and the other at receiver
- Sequence # = byte number of the first byte in the body.
- ACK # = the next expected byte # = the last seq # + the length of the last message.
- Cumulative ACK
- Initial seq # is randomly chosen.

Sender and receiver both need to give an ack # as well as a seq # in a message. The seq # A sends = ack # B sends to A last time.

Date No.

Flow control:

- Receiver 'advertises' an acceptable rate of transmission.
- Sender controls the number of packets on the fly.
- Such that the receiver's buffer won't overflow

Connection Establishment: 3-way Handshake

① C → S SYN · bit = 1, seq # = x.

② S → C SYN · bit = 1, seq # = y, ack # = x+1 even if the segment is empty. ACK · bit = 1.

③ C → S ACK · bit = 1. ack # = y+1. This is an ack of ②

Connection Teardown: 4-way Handwave.

① C → S FIN · bit = 1, seq # = x.

② S → C ACK · bit = 1, ack # = x+1. } Can be combined

③ S → C FIN · bit = 1, seq # = y } Can be combined

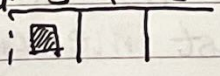
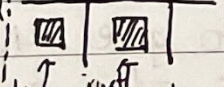
④ C → S ACK · bit = 1, ack # = y+1.

Server and still send data.

Client cannot send data, but can receive data

Client waits for 2 max segment lifetime just in case server did not receive ④ and keeps sending things.

Receiver Actions.

Event	Action	Purpose
Ack'ed: 3 4 5 	Wait for 500 ms If no second segment, send ack # = 4 (ack'ing 3)	Not to send too many ack to overwhelm the net
Ack'ed: 3 4 5  pending, just arrived	Send ack # = 5 immediately	To avoid time out

Event	Action	Purpose
ack'ed: 3 4 5 (5 received)	Send duplicate seq# =3 immediately	For fast transmit, telling the sender there is probably a loss.
ack'ed: 3 4 5 (4 arrived, 5 pending)	Send seq# =4 immediately	To avoid unnecessary timeout and retransmission

TCP doesn't specify how to handle out-of order segments.

Sender Actions.

Pretty much the same as Go-Back-N, only that the sender only resends the segment with the lowest seq# upon timeout.

RTT measurement & setting

- SampleRTT = $T_{ack\text{-}receipt} - T_{seq\text{-}trans}$ ignoring re-trans
- $EstRTT_{k+1} = (1-\alpha)\,EstRTT_k + \alpha\,SampleRTT_k$.
- $DevRTT_{k+1} = (1-\beta)\,DevRTT_k + \beta\,|SampleRTT_k - EstRTT_k|$.
- Typically, $\alpha = \frac{1}{8}$ and $\beta = \frac{1}{4}$. Timeout = EstRTT + 4 DevRTT

TCP Fast Transmit.

Don't wait for timeout.
Resend as soon as 4 ack's (3 duplicates) are received.

App. Protocol	Tpt. Protocol	Port.
HTTP	TCP	80 (default)
HTTPS	TCP	443 (default).
DNS	UDP	53
SMTP	TCP	25
DHCP	UDP	67 (svr) 68 (client).
RIP	UDP	520.

CS2105 Notes. II

IV Network Layer.

- Network layer handles host-to-host communication.

Functions

- Forwarding: determines output port on router based on the input port.
- Routing: determines the overall route.

Data Plane.

- Local, per-router
- Forwarding function.

Control Plane

- Network-wide logic
- Two approaches:
  - Traditional Routing Algorithms: Embedded in Routers
  - Software-defined networking (SDN): Implemented in remote servers.

IP addressing

- 32 bits for IPv4, 128 bits for IPv6.
- Interface: connection between host/router.
  - Routers ~~can~~ typically have multiple.
  - Hosts can have multiple. (different media)

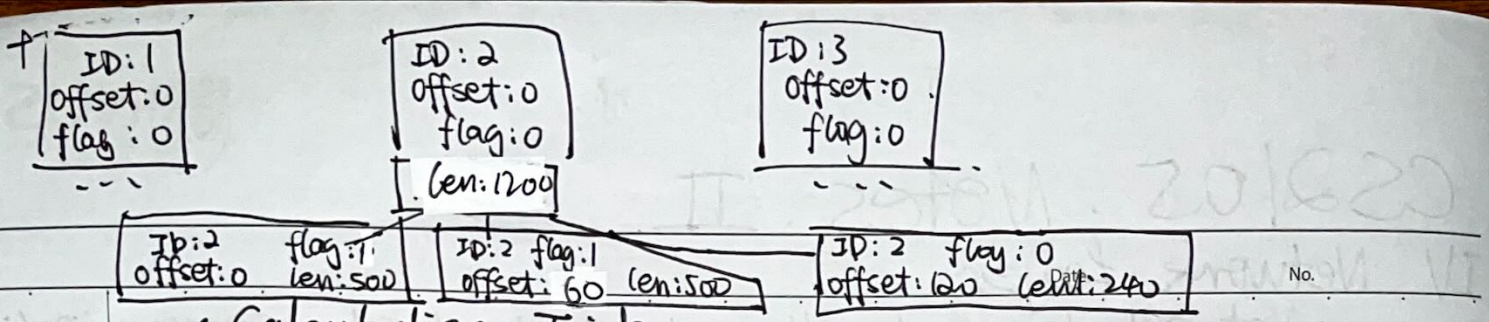
Different interfaces have different IP addresses
⇒ A router can have different IP addr.

Subnets.

- Hosts in the same subnet can, logically, "directly" communicate, without routers.
- Have the same IP prefix.

CIDR Classless InterDomain Routing

- a.b.c.d/x.
  - x: length of IP subnets Prefix.



- Calculation Trick:
  2^{32-x} public IP addr can be allocated to hosts in this subnet.
  Assume $0 \le x \le 8$, then given a.b.c.d/x,
  IPs that are in the same subnet are
  from $a-(a \bmod 2^{8-x}).0.0.0$
  to $(a-(a \bmod 2^{8-x})+2^{8-x}-1).255.255.255$.
  Remember to subtract the number of reserved addr if encountered in the range.
- Subnet mask is x 1's followed by 0's.
- Special IP addresses.
  0.0.0.0/8 Meta-addr.
  127.0.0.0/8 Loopback
  10.0.0.0/8
  172.16.0.0/12 } Private.
  192.168.0.0/16
  255.255.255.255 Broadcast.
- ICANN: Internet Corporation for Assigned Names and Numbers.
  Allocates IPs to ISPs.

Longest Prefix Matching.
- Forward the packet to the router whose subnet IP has the longest shared prefix with the destination IP.
  ↳ guaranteed to be unique.
- Note: Two IP addr can have the longest common prefix even if their decimal representations look very different.

Getting an IP.
- Hardcoded
- DHCP.
  - Plug-and-play; Dynamically gets an IP.

DHCP { server / client } port number { 67 of.

A yiaddrr has a lifetime -

Date No.

- Can renew lease on addr in use.
- Allows reuse of addr (only when connected).
- Support mobile users.

• Execution of DHCP.
① Host broadcasts "DHCP discover" [optional].
② Server "DHCP offer" [opt].
③ Host "DHCP request"
④ Server "DHCP ack"

IP Datagram Fragmentation.
An arriving IP datagram can be decomposed into several smaller IP datagram at a router. They will only be resembled at dest. †

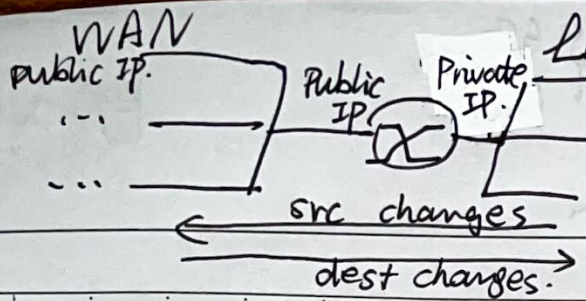
On the length field at header.
UDP len = header (8 bytes) + body, in bytes
TCP len = header (20 - 60 bytes) only.
4 bits ⇒ 2^4 values for 20 ~ 60?
Express $\frac{\text{len in bytes}}{4}$ ⇒ 5 ~ 15.
∴ in 4-byte words.

multiple of 8 except the last.

IP len = header (20 bytes) + data, in bytes.
IP offset = displ (first byte in the same fragment).
in 8-bytes

IP Format.
• Frag: 1: not the last. 0: the last.
• offset: relative position to the original beginning
$offset_{i+1} = offset_i + (length_{i+1} - 20)/8$.
∴ in 8-byte words.

NAT:
• Intuition: exploit the port number space to make up for shortage of IP addr space



Date No.

- A "façade" router that sits between WAN and LAN has a public IP and a private IP.
- Implementation: Maintain a 1-to-1 mapping.
  - Replace (private IP, port) with (router IP, dedicated port) when sending out
  - Remember the mapping.
  - Replace (router IP, dedicated port) with (private IP, port) when sending in.
- Also a security plus in the sense that the outside world cannot directly address a private IP.
- Challenge: "NAT mare" Peer-to-peer apps don't work directly.

Routing.
- Intra-As: No policy decisions. Focus on performance.
- Inter-AS: Policy may dominate.

Cost:
- Could be constant.
- Or inversely related to bandwidth
- Or related to congestion.
- Or distance
- Or financial.

$C_{x,y}$ = cost (x, y).
= ∞ if x and y don't neighbour.

$D_x(y)$ = cost of the least-cost path from $x \to y$

Bellman-Ford Equation.

$$D_u(z) = \min_{a \in N} \{c(u,a) + D_a(z)\}.$$ N: 1-hop neighbours

distance to every node.

Routing Algorithm.

① (wait for) Receive distance vectors from neighbours

② Compute its own distance vector.

$$\begin{bmatrix} a_1 \\ \dots \\ a_m \end{bmatrix} = \begin{bmatrix} \min(C_{a,b} + b_1, \dots, C_{a,z} + z_1) \\ \dots \\ \min(C_{a,b} + b_m, \dots, C_{a,z} + z_m) \end{bmatrix}$$

③ Notify neighbours only when local distance vector changes, possibly due to:

a) Link cost changes.

b) Topology changes.

c) DV update from neighbours.

Note: The above algorithm terminates/stabilizes in finite steps from $t = t_0$ if there are no further a) or b) changes from to onwards. However, the number of steps may be greater than the diameter of the graph.

RIP.

- Uses hop count as the cost → not sensitive to traffic.
- Exchange routing table every 30 sec.
- "Self-repair": assume a neighbour has failed if no update for 3 min.
- Distributed, iterative, and asynchronous.

ICMP.

- Error reporting: unreachable host/net/port/protocol.
- Echo request/reply: ping. (uses a connection).
- Exchange info about net.

For example, a packet can be dropped due to TTL expired ⇒ send an ICMP to notify the sender.

V Link Layer.

Handles neighbour-to-neighbour communication, over a single link.
Deals with addressing, protocol, and errors.

Concerns and Services.
- Framing.
- Link Access Control.
- Error Dection.
- Error Correction.
- Reliable Delivery: typically for error-prone links such as wireless links.

Link layer is implemented in "adapter" or a chip.
Adapters are semi-autonomous, implementing both link & physical layers.

(Even) Parity Checking (Single Bit)
- Number of 1's in the d+1 bits is even.
- Can detect odd number of single bit errors but not even number of single bit errors.
- In practice, Pr[undetected | error] ~> 50%.
- Works well if errors are independent.

2D Parity Checking.
- For an m×n matrix, there are m+n+1 parity bits.
- The (m+1, n+1) bit should be the same regardless of whether it is evaluated by row or col.
- Can detect and correct single-bit errors.
  Can only detect a two-bit error.
  Can detect odd number of single-bit errors.

Date No.

Cyclic Redundancy Check.

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

① Append r 0's to D, resulting D'.
② R := D' mod G under Modulo 2 Arithmetic.

$$x+y = x-y = x \oplus y.$$

③ Send and receive (D, R), meaning D concat. R.
④ Receiver checks if (D, R) mod G = 0. If not, some error exists

Intuition: divisor | (dividend − remainder)
dividend − remainder = dividend ⊕ remainder
= dividend + remainder = D' + R = (D, R).
∴ G | (D, R).

- Can detect ☑ all odd number of single-bit errors.
  ☑ all burst errors of ≤ r bits
  ☑ all burst errors of > r bits with prob $1-(\frac{1}{2})^r$
- A.K.A. polynomial code. k-bit frame = k coefficients for a polynomial of degree k-1.

Types of Network Links.

1. Point-to-Point Link: PPP, SLIP.
2. Broadcast Link (shared medium): Every node sharing the link receives a copy.

Multiple Access Protocols.

- Random Access: Possible collisions ⇒ recovery is concerned
- Taking Turns:
- Channel Partitioning: Time/Frequency.

Complexity ↑

Date No.

Ideal Multiple Access Protocol. (given total capacity R)
1. Collision Free (Mutual Exclusion).
2. Efficient: If only 1 node is transmitting, it can achieve R. (Progress).
3. Fairness. Each node has an average of R/N. (Starvation Free).
4. Fully Decentralized.
   - No out-of-band communication.

Channel Partitioning.

TDMA/FDMA.
- Fixed timeslots/frequency bands that are pre-agreed /hardcoded.
- For FDMA, the average is still R/N because the difference in frequency on different bands are negligible.
- Satisfies Property 1, 3, 4.

Taking Turns.

Polling.
- The master node polls every slave node in round-robin. Admitted slaves take turn to send at most m packets/frames.
- Need to consider polling overhead.
- Satisfies 1, 2, 3.
  ↳ not completely.

Token Passing.
- A special frame, called token, is passed along a (logical) ring topology (the underlying physical topology does not have to be a ring).
- When holding the token, can send something. Forward it if finished.
- Satisfies 1, 2, 3, 4
  ↳ not completely.
- Need to consider token loss and node/link failure.
  - token loss → data loss, system bug.
  - node/link failure → broken topology.

On Random Access: transmit at full rate R.
no a prior coordination.
On ALOHA collision detection: Listen while sending.
For some propagation delay too long for a collision to be detected while transmitting → relies on ACK.

Date No.

Slotted ALOHA.

- All frames are L bits long.
- (Time Quantum) = time for sending a frame.
  = Timeslot.

① If there is a frame to send, wait for the beginning of the next slot.

② No collision ⇒ success.
Collision ⇒ failure ⇒ retransmit in each following slot with prob p. until success.

- Satisfies ✗ 2, 4.
- About 2: a) Full rate once a node starts.
  b) Only 37% ($\frac{1}{e}$) of R if there are many active nodes.
  c) Slots are wasted due to collisions or being empty (the prob leads to this).

Pure/Unslotted ALOHA.

- No timeslots. + No synchronisation.

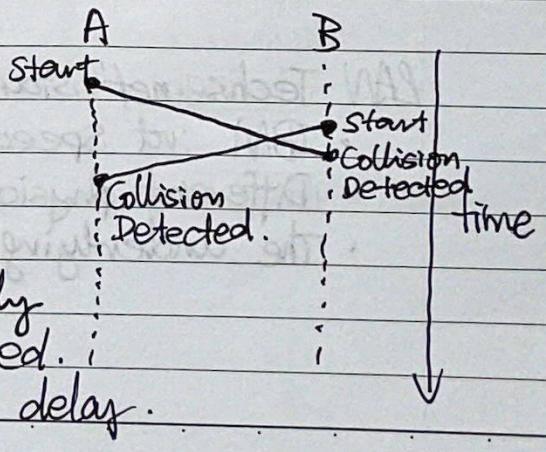
① Transmit immediately.

② For failure, wait for 1 unit time, retransmit the frame with prob p.

- Chance of collision increases, resulting in halved overall capacity (18%).
- Satisfies ✗ 3, 4.

CSMA.

- Listen before speaking.
- Collisions are still posible.
- Unslotted.



CSMA/CD.

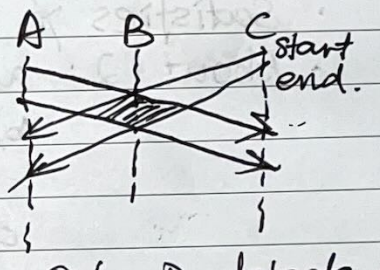
- Abort transmission immediately if there is a collision detected.
  Retransmit after a random delay.
- Satisfies 2, 3, 4.

※ Constand delay prob p only burdens the future. It can even cause more collisions.

Date No.

Back-off Algorithms. (Binary Exponential Backoff).
- After the m-th collision.
  choose $k \in \{0, \ldots, 2^m - 1\}$ at random.
  such that the prob of retransmitting is $p = \frac{1}{2^m}$.
- Intuition: More collisions imply heavier loads.
- Possible Improvement: Can start over again from m=0 after a period of time, otherwise old past history influnces now too much.

Minimum Frame Size.
- Ethernet: 64 bytes.



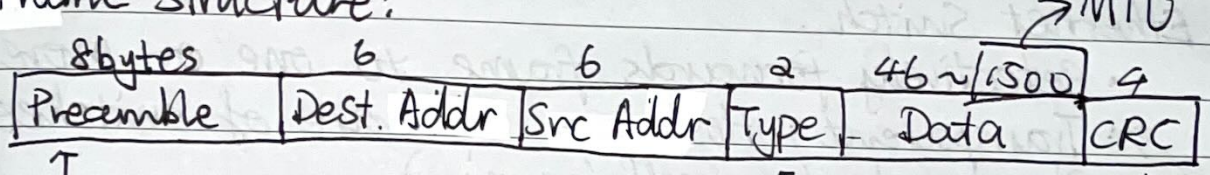
Only B detects the collision.

MAC Address.
- A.k.a., physical/LAN address.
- Upon receipt of a frame, a device checks if the destination is its MAC addr
  yes → extract datagram and hand over it.
  no → discard.
- Typically 48 bits, burnt in ROM (sometimes software settable).
- Broadcast Address: FF-FF-FF-FF-FF-FF.

802.3 Ethernet Standards.
- Different speeds: (2M, 10M, 100M, 1G, 10G, 100G) bps.
- Different physical layer media: cable, fiber optics.
- The underlying protocol is CSMA/CD with backoff.

Date No.

- Frame Structure:



- 10101010×7 $(AA)_{16}$, 10101011×1 $(AB)_{16}$. used to synchronise clock rates. (the width of a bit), important when there are 19 or 20 0's.

The higher layer protocol, e.g., Novell IPX, AppleTalk, ARP.

No retransmission if corrupted. ∴ unreliable dropped frames will only recovered when the higher layer offers retransmission

Ethernet Topology.

- Bus
  - All nodes can collide with each other.
  - If the backbone cable fails, the entire network will fail.
  - Very slow due to collisions → not ideal for large networks.
- Hub (Star).
  - A physical layer device that acts on individual bits rather than frames.
  - Re-create and boost the bit signal, send it to all other interfaces → logically still a bus.
  - Cheap, easy maintenance, very slow, not ideal for large networks.
- Switch (Star).
  - A link layer device.
  - Acts on individual frames in a store-and-forward fashion.
  - No collisions.

Q: How can A know if B is in the same subnet?
A: Via DHCP and subnet name and mask.

Date No.

Ethernet Switch.

- Selectively forwards frame to one or more links.
- Transparent: Hosts are unaware of switches.
- Plug-and-play (self-learning).
- Multiple simultaneous transmissions with buffer but without collisions.
  "simultaneous" in the sense that A→A', B→B' can be done in parallel.

That's why nodes in an LAN are logically one hop apart.

Frame Filtering/Forwarding.

- Forwarding table:
  <MAC addr, Interface, TTL> → renewed upon every receipt.
  Each tuple means: "I have seen MAC x from Interface y." This tuple is valid for TTL.
- Suppose the table has only one record <x, y, 60>.
  - On receipt of x on y: discard.
  - On receipt of x on z≠y: send to y.
  - On receipt of w on z: send to every interface except z. "flood"

ARP

- A link layer protocol.
- Each IP node has an ARP table about hosts in the same subnet
  <IP addr, MAC addr, TTL> → typically a few min.
- Plug-and-play.

Sending Frame in the Same Subnet. (A→B)

① If A knows B's IP. in the ARP table → trivial.
② Otherwise, use ARP.
- A broadcasts to (FF ×6) with B's IP.
- B replies with B's MAC.
- A caches B's MAC until TTL expires.

* LAN ≠ subnet.
LAN may contain multiple subnets.

Date No.

Sending Frame to Another Subnet.

- Cannot just set dest = B's MAC, otherwise it will be discarded by all nodes in the same subnet including (the link layer of) the router, due to MAC mismatch.
- Correct approach: set IP dest = B's IP, MAC dest = R's MAC. Upon receipt at R, R changes MAC dest = B's MAC, MAC scr = R's MAC.
- ∴ <MAC src, MAC dest> changes when subnet changes

* What if a node does not even know about its gateway router?

① There is a default gateway in the ARP table, usually.

② Via DHCP, get the subnet name and mask, and hence the gateway router

③ Use ARP proxy reply. The ARP broadcast query with B's IP will now be handled by the router as a proxy. The router will reply A that it can reach B by its MAC. ⇒ From now, every frame from A to B will have MAC dest = R's MAC, IP dest = B's IP.

VI Security.

Malicious behaviors include:

- Eavesdrop: intercept.
- Insert
- Impersonation.
- Hijack: Man in the middle. Take over the connection
- DoS.

Date No.

Symmetric key
Signature & certificate
Hash, MAC

Aspects of Security.

- Confidentiality: only someone can understand
- Authentication: confirmed identity.
- Message Integrity: not altered without detection.
- Access & Availability.

Notation

$K_A K_B(m) = m$ → the plaintext.

Caesar's Cipher

- An example of substitution cipher.
- Key: the shift number.
  right shift by n letters: every char is moved to the right by n letters in the alphabet.

plaintext	ciphertext for n=3
a	d
b	e
...	...

- Easy to do brute-force.

Monoalphabetic Cipher.

- A one-to-one mapping of letters.
- 26! possible keys, 26!-1 of which are meaningful
- Easy to break with statistical analysis. (frequency analysis). Some two-and-three-letter occurrences are often together (is, are, of, to, -ion ...).

Polyalphabetic Encryption.

- Addresses the problem, also the fundamental weakness, of having only one mapping.
- Uses multiple mappings. Select n substition ciphers and a pattern $C_1C_3C_2...$
- $K(m_1 m_2 m_3 m_4 m_5) = C_1(m_1)C_3(m_2)C_2(m_3)...$

NUS National University of Singapore

Date No.

Block Cipher

- Processes m in blocks of K bits.
- Uses a 1-to-1 mapping. e.g., 001 → 111, 010 → 100.
- Number of keys $(2^K)!$
- Hard to break with frequency analysis if K is a good choice.

DES:

- 56-bit symmetric key, 64-bit block.
- Broken in < a day.
- 3DES: encrypt 3 times with 3 different keys.

AES

- 128-bit blocks, 128-, 192-, or 256-bit keys. (+64, +64)
- If a computer can break DES in 1 sec, it will break DES in 149×10^{12} yrs.

Setting of Attacks.

- Ciphertext Only.
- Known-plaintext. (may not be totally known)
- Chosen-plaintext (use the encryption algo as a black box arbitrarily).

Public key (Asymmetric) Encryption.

- $K^+(\cdot) \neq K^-(\cdot)$, but $K^-(K^+(m)) = m$
  Given K^+, it is merely impossible to find K^-
- RSA is an example.

Modulo Arithmetic.

$(a \overset{+}{\times} b) \bmod n = [(a \bmod n) \overset{+}{\times} (b \bmod n)] \bmod n.$

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

Date No.

RSA:

① Choose 2 large prime numbers p, q.
② $n := pq$, $z = (p-1)(q-1)$
③ choose $e < n$ that is co-prime with z
④ choose d such that $ed \bmod z = 1$.
⑤ $K^+ = (n, e)$, $K^- (n, d)$.

Encryption:

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

Decryption:

$$m = c^d \bmod n = (m^e \bmod n)^d \bmod n$$

Note that K^+ and K^- are commutative, meaning

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

The underlying challenge: factorising a big number is hard.

Session Key:

- DES is ×100 faster than RSA, but need Ks in advance
- Use RSA to deliver Ks.
- Use Ks to en/decrypt messages for the session.

Motivation for Alteration Detection:

1. Checksum only detect accidental errors not attacks
2. CRC clashes are common and easy to find.
   Minor change in input → minor change in output

Cryptographic Hash Function:

- Message of arbitrary length → fixed-length digest. ("fingerprint")
- Computational infeasible to find $x \neq y$ s.t. $H(x) = H(y)$.
- MD5 → $\{0,1\}^{128}$ = $\{0,1\}^{128}$ } Cryptographically broken, deprecated.
- SHA-1 → $\{0,1\}^{160}$ = $\{0,1\}^{160}$

Message Integrity:

(Does not have confidentiality)

- Use a shared "Authentication Key" s.
- Send $(m, \underbrace{H(m+s)}_{\text{Message Authentication Code.}})$; Receiver checks if $H(m'+s) = H'$

Date No.

Password.
- Saved in hash → can only be reset

Digital Signature.
- Desired properties.
  - Verifiable → both parties can check
  - Unforgeable. → only authorized parties can generate.
- Send $(m, K^-(m))$ or $(m, K^-(H(m)))$.
  signature = message encrypted with private key
  ~~message~~ fingerprint.
- Receiver verifies $K^+(K^-(m)) = m'$ or
  $K^+(K^-(H(m))') = H(m')$.

Certification Authorities.
- A list of CAs for receiver to query trusted public keys
- These CAs' messages are also signed. ⇒ their signatures are maintained as universal knowledge by Trusted Root CAs.
- Entity E provides CA with its proof of identity. ⇒ the certificate contains $(K_E^+, K_{CA}^-(K_E^+))$.

Firewalls
- Isolate an organisation's internal network from the outside world.
- Re-construct segments/datagrams/messages if needed.
- Three types.
  1. Stateless packet filters ☆.
  2. Stateful packet filters
  3. Application gateways.
- Limitations: IP spoofing, performance bottleneck.

Stateless Packet Filter.
- Decides to forward/drop based on addr, port #, ICMP message type, SYN/ACK bits.

Date No.

Access Control List.
- Conditions are evaluated top-down.

VII Multimedia Networking.

At App-layer and Transport Layer.

3 Types of Applications.

1. Streaming stored audio, video.
   Streaming: playout begins before downloading the entire file.
   Stored (at servers/CDNs): can transmit faster than it is played. ⇒ buffering.
2. Conversational
   Interactive, delay-sensative.
   Intolerable with >400 ms
3. Streaming live
   Typically done with CDNs.

Video.
- Frames played in sequence at constant rate.
  ↓
  Arrays of pixels → bits.

Exploits Redundancy:
- Spatial Coding: repeated colors → count the occurrences
- Temporal Coding: similar frames → send differences only
- CBR
  - Not responsive to complexity change.
  - High rate to handle sudden increase in complexity
  - Suitable for real-time encoding.
- VBR.
  - Changes according to how spatial/temporal coding changes
  - Suitable for on-demand video.

NUS National University of Singapore

Date No.

Audio:
- Sampling (quantized to bits)
- At constant rate:
  - Telephone: 8000 /sec.
  - CD 44100 /sec
- Bit rate = sample size × sampling rate.
  - CD: 1.411 Mbps.
  - MP3: 96, 128, 160 Kbps.
  - Internet telephony: ≥5.3 Kbps

Motivates:

Continuous Playout Constraint:
- Once client starts playing, the speed must match
  - but there are jitters.

Other Challenges
- Client Interactivity: pause, fast-forward, rewind, jump etc.
- Lost/retransmitted packets.

→ Buffering.
1. Initial fill of buffer until playout at t_p.
2. Playout begins.
3. Buffer level = $x(t) - r$ varies.
   - $x(t)$: fill rate varies
   - r: constant playout rate.

Consider the average, $\bar{x}(t)$
- $\bar{x}(t) > r$ → eventually full
- $\bar{x}(t) < r$ → eventually empty. (buffer starvation)
  - less likely with larger delay

∴ The initial fill should be large enough to absorb the variance of $x(t)$.

Calculation Tricks:
- Measure everything in "chunks"

Date No.

Push-Based Streaming.

- Uses UDP. → no congestion control → no max rate restrictions.
- Short playout delay. (2-5 sec).
- Error recovery at app-level if time permits.
- Video chunks encapsulated using RTP (seq #, timestamp, encoding ...).
- Control Connection maintained by RTSP (usually over TCP)
  - Handles client command (pause, play ...).
  - Establishes/controls media sessions.
- Drawbacks.
  - More cost/complexity due to a separate RTSP.
  - May not go thru firewalls.

Pull-Based Streaming.

- Over HTTP/TCP.
- Easier to get thru firewalls.
- Infrastructure (CDNs, Routers) fine-tuned for it.
- Fill rate fluctuates.
- Greater playout delay.
- Congestion Control exists.
  - Additive increase V.S. Multiplicative decrease.

VoIP

- End-end-delay requirement.

Delay loss →
- < 150 ms : good. } includes app-level (packetization + playout) delay.
- > 400 ms : bad. }

Network loss → - Loss over 10% → unintelligible.

- Packets only generated during talk spurts.
- 20 ms chunks at 8 Kbps = 160-byte chunks
- (Chunk + header) ∈ TCP or UDP. → more common.
  - header: { seq # timestamp.

Date No.

- Receiver has a constant q_r. Chunks generated at t will be played at $t+q$.
  - Large q_r → less loss (fewer "too-late" chunks).
  - Small q_r → better e-t-p.
  - But no q_r guarantees optimal performance.

Adaptive Playout Delay.

- Estimates net delay, adjusts playout delay (yes there are many playout delays for a conversation, unlike videos). of each talk spurt.
- Silent periods are compressed/elongated.
- EWMA:

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

d_i: delay est. after ith chunk. α: small. e.g. [0.1]. r_i: time received. t_i: time sent.

Apply after receiving the i-th chunk.

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

v_i: est. avg deviation of delay. β: small. $r_i - t_i$: last actual delay. d_i: predicted next delay.

$$(\text{playout time})_i = t_i + d_i + 4v_i$$

Only applicable for the first packet in a spurt!!

Remaining packets in the same spurt are played out periodically.

Forward Error Correction (FEC) → VoIP

1. Simple FEC.
   - 1 extra chunk XOR-ed by n chunks.
   - Lower performance by multiplying $\frac{n}{n+1}$.
   - Longer playout delay → n+1 must arrive before playing.

Date No.

2. "Piggyback" FEC.
   - Piggybacks low-quality stream of the last (and second last ...) chunk. → conceal loss

3. Interleaving.
   - Chunks are split into n small units, disordered, and reassemblied to n-unit chunks
   - If packet lost, still have most of every chunk.
   - Longer playout delay because of the need to wait for different chunks to be generated.

DASH.
- Deals with 3 things.
  1. When to request
  2. What encoding rate (quality)
  3. Where to request.
- Data encoded into different qualities, cut into short segments (streamlets, chunks).
- Manifest file.
- ABR.
- Advantages
  - Simple server
  - No firewall problems
  - standard (image) web caching works (due to use of HTTP).
- Disadvantages
  - 2-10-sec media segments → not suitable for two-way / interactive apps.

CDN
- Option ①: Large "mega-server"
  - does not scale (with other shortcomings)
- Option ②: Store copies on geographically distributed sites.

Date No.

- Enter deep: get deep into access networks, close to end users.
- Bring home: larger clusters at IXPs near access networks.

Execution:

① CDN stores copies at different nodes
② Client requests for content and gets manifest.
③ Client retrieves the highest quality affordable.
④ May dynamically choose different rate/copy in case of high-latency/congestion/loss/failure

Appendix

About DHCP.

- Implemented at internet edge.
- Gives the following:

1. IP Addr of first-hop router.
2. Name & IP addr of DNS server.
3. Network Mask.

Subnet Mask Numbers:
128, 192, 224, 240, 248, 252, 254, 255.

Autonomous Systems.

- Can have multiple routers and subnets.
- Belong to an organization/institution/entity, which can have multiple ASes.