Routing

- The internet is a "network-of-networks"
  - A hierarchy of Autonomous Systems (AS), eg. ISPs, each owns routers and links.

- Due to the size of the internet and the decentralised administration of the Internet, routing on the Internet is done hierarchically:
  - Intra-AS routing *
    - Finds a good path between 2 routers within an AS. (Organisation)
    - Commonly used protocols: RIP, OSPF.
  - Inter-AS routing (not covered)
    - Handles the interfaces between ASs

      De facto standard protocol: BGP.

- When discussing routing, we can view a network of routers as a graph, where vertices are routers and edges are physical links between routers
  - ↳ we can associate a cost to each link.
    - ↳ Routing: Finding a least cost path between 2 vertices in a graph.

- Routing Algorithms:

1. "link state" algorithms
   - All routers have the complete knowledge of network topology and link cost.
   - Routers periodically broadcast link costs to each other.
   - Use Dijkstra's algorithm to compute least cost path locally (using global map)

2. "distance vector" algorithms
   - Router know physically-connected neighbours and link costs to neighbours.
   - Routers exchange "local views" with direct* neighbours and update own "local views" (based on neighbours' view)
   - Iterative process of computation.
     1. Swap local view with direct neighbours.
     2. Update own's local view.
     3. Repeat 1-2 till no more change to local view.

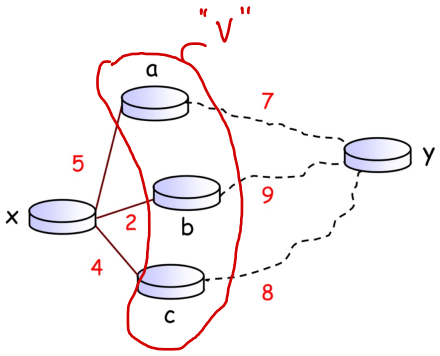
Bellman-Ford

cost of link between x and v, ∞ if not neighbours.

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

cost of least-cost path from x to y (from x's view)

min is taken over all direct neighbours v of x.



$$d_x(y) = min_v \{ \underset{5}{c(x,a)} + \underset{7}{d_a(y)}, \; \underset{2}{c(x,b)} + \underset{9}{d_b(y)}, \; \underset{4}{c(x,c)} + \underset{8}{d_c(y)} \}$$

$$= min\{12, 11, 12\} = 11$$

- 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.

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)

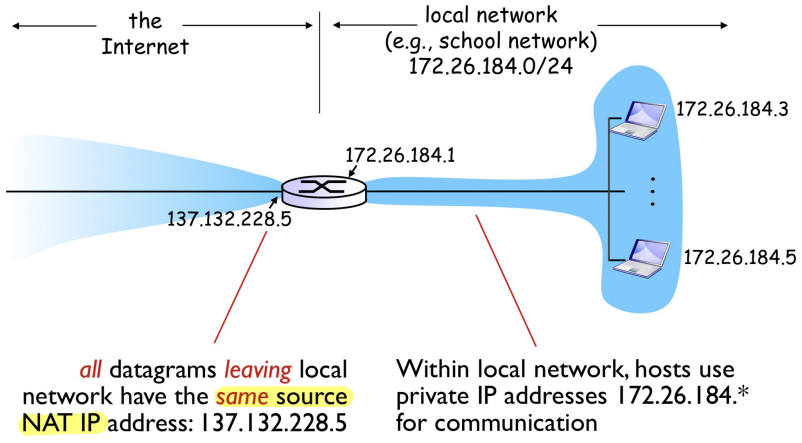
- Implements the Distance Vector algorithm.
- Uses hop count as the cost metric (i.e. insensitive to network congestion)
  - i.e. links

UDP: exchange routing table every 30 seconds over port 520.

"Self-repair": if no update from a neighbour router for 3 mins,
assume neighbour has failed.
(remove corresponding entry from table)
prevent sending message to a neighbour that is not working.

NAT

- Network Address Translation
- limited public addresses ∴ Many uses private IP. (unique in network, not globally unique)
  - Cannot be used as destination IP address.

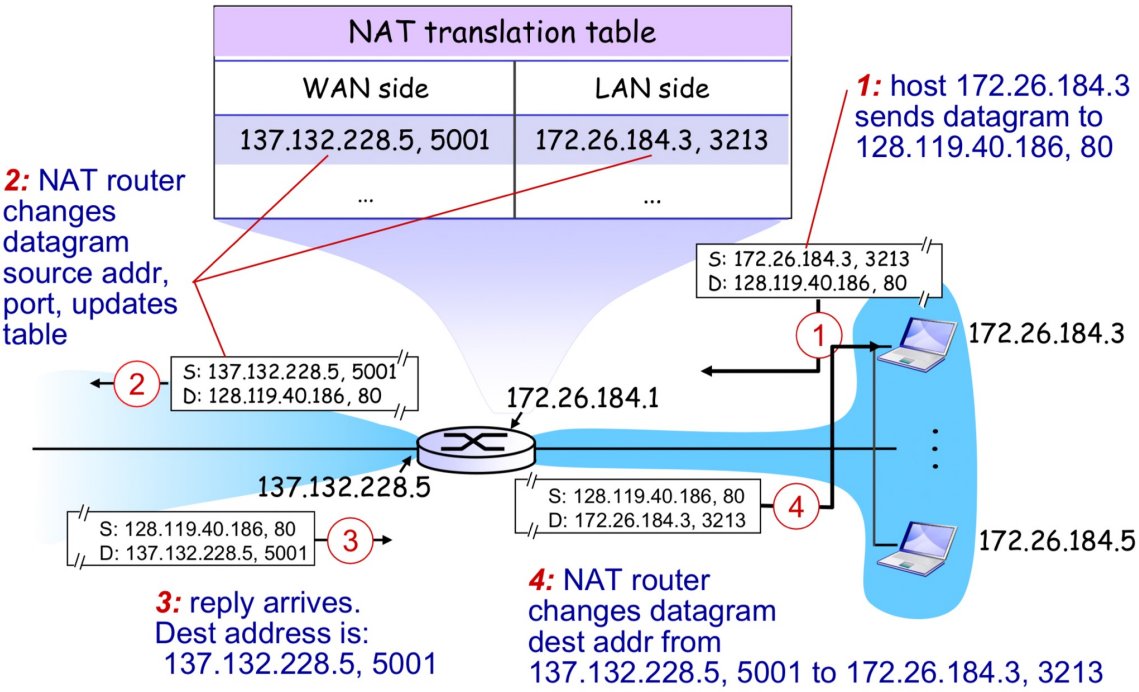


NAT Router:

To deliver reply packets correctly. (no conflict)

- Replace (Source IP addr, port #) of every outgoing datagram to (NAT IP addr, new port #)
- Remember (in NAT translation table) the mapping from (Source IP addr, port #) to (NAT IP addr, port #)
- Replace (NAT IP addr, new port #) in destination fields of every incoming datagram with corresponding (Source IP addr, port #) stored in NAT translation.

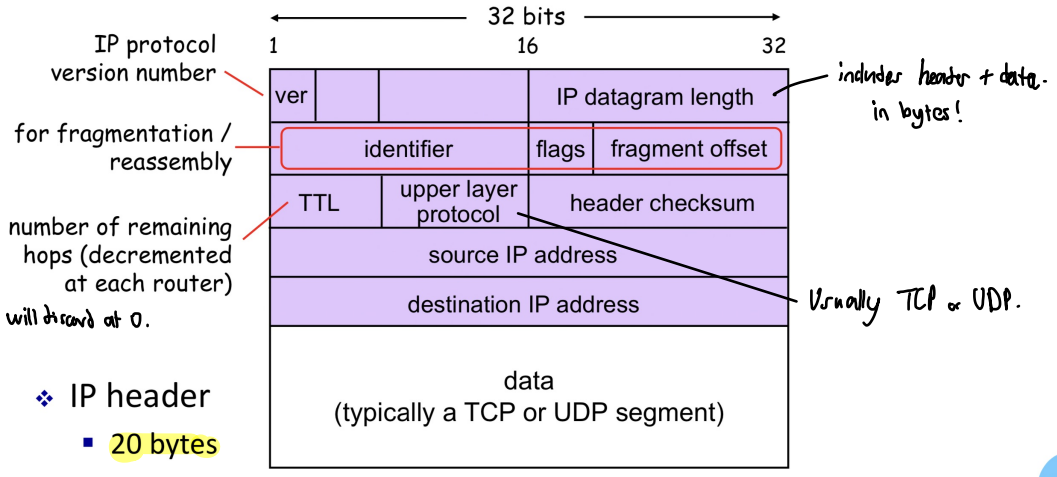
Another Example:



- Packets are modified by Router, invisible to hosts.
- ipconfig/all to see private IP.

IP : Internet Protocol

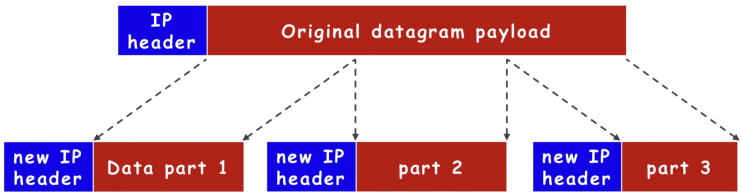
IPv4 Datagram Format

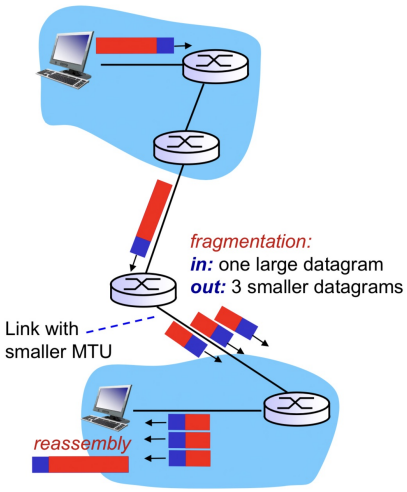


- IP header
  - 20 bytes

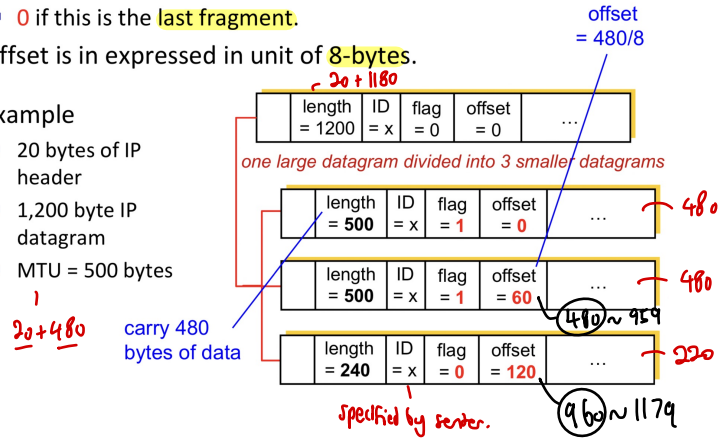
IP Fragmentation and Reassembly.

- Different links may have different MTU (Max Transfer Unit)
  - the max amt. of data a link-level frame can carry (Max size of IP datagram incl. of header)
- "Too large" IP datagrams may be fragmented by routers.





- Destination host will reassemble the packet
- IP header fields are used to identify fragments and their relative order.
  - 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.
  - Example
    - 20 bytes of IP header
    - 1,200 byte IP datagram
    - MTU = 500 bytes



ICMP

- Internet Control Message Protocol
- used by hosts & routers to communicate network-level information.
  - Error reporting: unreachable host/network/port/protocols.
  - Echo request/reply (used by ping)
- ICMP messages are carried in IP datagram
  - ICMP header starts after IP header.

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

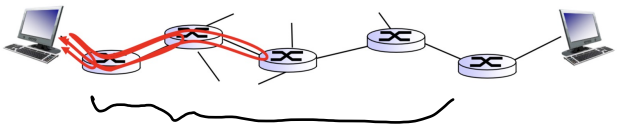
Type	Code	Description
8	0	echo request (ping)
0	0	echo reply (ping)
3	1	dest host unreachable
3	3	dest port unreachable
11	0	TTL expired
12	0	bad IP header

Selected ICMP Type and subtype (Code)

- The command ping sees if a remote host will respond to us – do we have a connection?

– i.e. tracert

- The command traceroute sends a series of small packets across a network, and attempts to display the route (or path) that the messages would take to get to a remote host.



will display all the routers in between.

Summary

- An IP address is associated with a network interface. A device may have multiple network interfaces, thus multiple IP addresses.
- DHCP automates the assignment of IP addresses in an organization's network.
- On TCP/IP networks, subnets are defined as all devices whose IP addresses have the same network (subnet) prefix.
- Subnet mask is useful in checking if two hosts are on the same subnet.
- Routing is the process of selecting best paths in a network.
- NAT maps one IP addresses space into another.
  - Commonly used to hide an entire private IP address space behind a single public IP address.
  - NAT router uses stateful translation tables to remember the mapping.
- ICMP is used by routers to send error messages.
  - E.g. when TTL is 0, a packet is discarded and an ICMP error message is sent to the datagram's source address.

Qns - Network Layer II

Q1

A file is transmitted over TCP. MSS is 1,000 bytes and TCP adds 20 bytes of header to each segment. Each segment is then encapsulated into an IP datagram that has a 20 bytes IP header. Each IP datagram is further encapsulated in to a link layer frame which adds another 18 bytes of header/trailer.

What is the percentage of the bytes of the file in a link layer frame (corrected to 2 decimal places)?

A. 94.5%
B. 98%
C. 100%

Q2

Which of the following fields in an IP datagram header may be changed by a NAT-enabled router?

i. TTL
ii. Checksum
iii. Destination IP address
iv. Source IP address

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

Q3

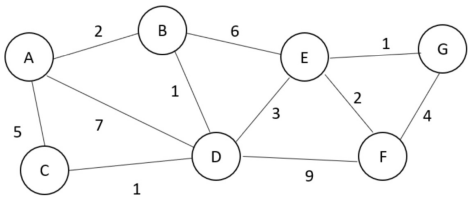
Consider sending a 2,000-byte IP datagram over a link that has an MTU of 100 bytes. Suppose IP header is 20 bytes.

How many IP fragments will be generated?

A. 20
B. 24
C. 25

Q4

Consider the network topology shown below. Routers A … G each runs distance vector algorithm.

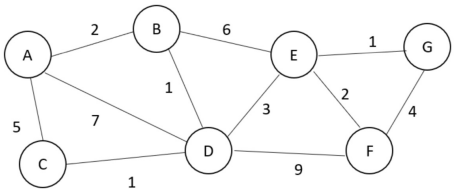


Show the initial distance vector table of router A (i.e. before the distance vectors are exchanged). If A is unaware of another router, write '-' in the corresponding slot.

	cost to A	cost to B	cost to C	cost to D	cost to E	cost to F	cost to G
from A							

Q5

Consider the network topology shown below. Routers A … G each runs distance vector algorithm.



Suppose the distance vector algorithm 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 A.

	cost to A	cost to B	cost to C	cost to D	cost to E	cost to F	cost to G
from A							

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?
   Data (segment) length = 1500 – 20 = 1480 (due to 20 bytes IP header)

   Maximum size of data in each fragment = 500 – 20 = 480

   Number of fragments = $\lceil 1480/480 \rceil$ **= 4**

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

   1st – 3rd fragments have length 500. 4th fragment has length 60.

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

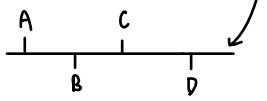
Fragments	ID Number	Offset	Flag
1	422	0	1
2	422	60	1
3	422	120	1
4	422	180	0

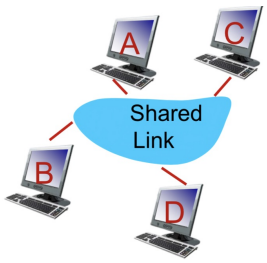
Link Layer

- Aim: Send data between nodes via cable.
  - Communication Channel: the transmission medium of the data signals.
  - nodes: i.e. hosts, routers.

- Solution: Inter-connect N nodes and send data.
  - Each link needs to be addressed
  - Drawback: does not scale, N-1 links needed.

- Solution 2: Inter-connect N nodes via a broadcast link
  - i.e.





  - ✓ Each link needs to be addressed — Framing
  - ✓ Need to define a protocol (due to shared medium) — Link Access Control
  - ✓ Need to handle errors — Detection, Reliability.

- Network layer provides communication service between any 2 hosts
  An IP datagram may travel through multiple routers and links before it reaches destination.

- Link layer sends datagram between adjacent nodes (hosts or routers) over a single link.
  - A single hop connects 2 nodes

- 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

Framing

- encapsulate datagram into a 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.

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

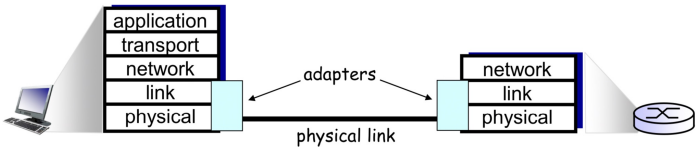
- receiver identifies and corrects bit error(s) without resorting to retransmission.

Reliable delivery (Not in scope)

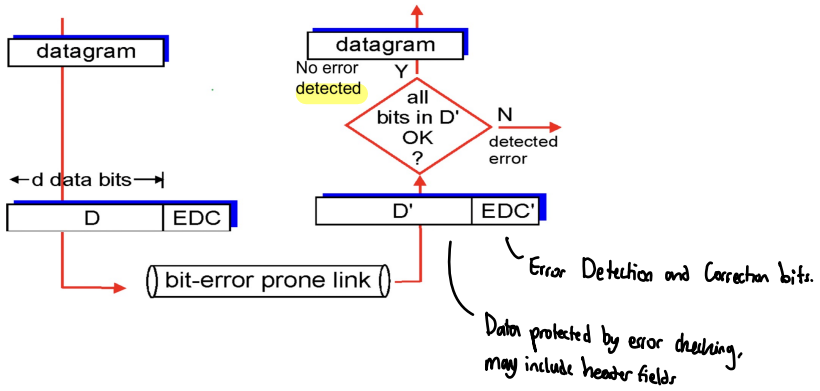
- Seldom used on low bit-error link (eg. fiber) but often used on error-prone links (eg. wireless)

Link Layer Implementation

- Link Layer is implemented in "adapter" (aka NIC) or on a chip. ← Network Interface Card.
  - eg. Ethernet card, Wi-Fi adapter
- Adapters are semi-autonomous, implementing both link & physical layers.



Error Detection

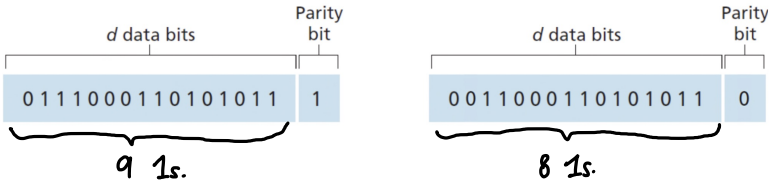


* Error detection schemes are not 100% reliable!
  - may miss some errors, but rarely.
  - usually, longer EDC fields yields better detection (and even correction)
- Error detection schemes:
  - Checksum (used in TCP/UDP/IP)
    - Treat segment contents as sequence of 16 bit integers
    - 1's complement of the sum of segment contents.
  - Parity Checking.
  - CRC

Parity Checking

(single bit)

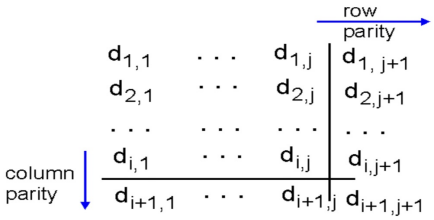
- (even parity) Includes one additional bit, choose its value such that total number of 1s in the d+1 bits is even.
  - i.e. 1 if #1s is odd, 0 if #1s is even



- Can detect single bit errors in data
  - Can detect any odd number of single bit error ("flips").
  - Cannot detect even number of flips.
- Errors are often clustered together in "bursts".
  - probability of undetected errors in a frame can approach 50%

(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

↳ parity bits can also be flipped!

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

Parity bit for the column and row parity bits

$d_{i+1,j+11}$

- Can detect and correct single bit errors in data

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

Cost: Overhead of 9 bits for 15 bits of data.

$\sim 37.5\%$

- reduces overhead of retransmission, throughput increases considerably.

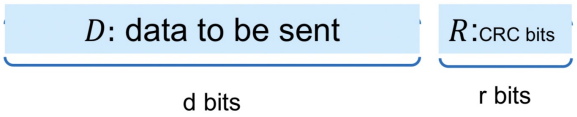
- Can detect any 2-bit error in data.

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

- Cannot be corrected, will throw data away.

CRC

- Cyclic Redundancy Check. Used in link layer)
  - 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.

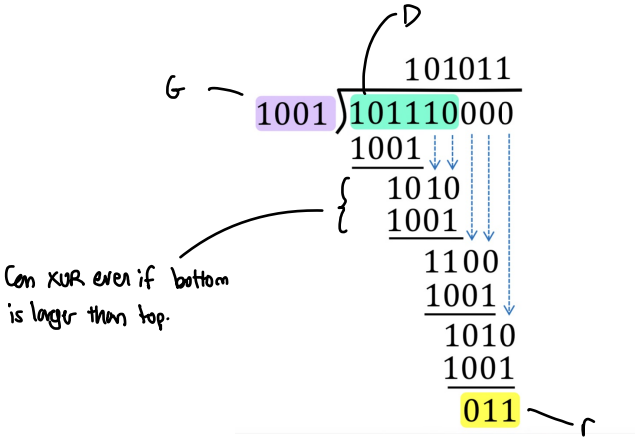


Calculations are done modulo 2.

- It does not have carries for addition or borrows for subtraction.
- Both addition and subtraction are *identical* to XOR

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

- Sender sends (D, R)
  101110011
- Receiver knows G, divides (D, R) by G.
  - If non-zero remainder: error is detected!

- Easy to implement on hardware
- Powerful error-detection coding that is widely used in practice (eg. 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 $1-0.5^r \approx 1$
- CRC is aka 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
    $\Rightarrow 1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0 = x^5 + x^4 + 1$

Link Layer Protocols

Types of Network Links:

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

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.

Multiple Access Protocols

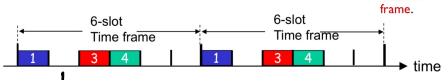
- In a broadcast channel, if 2 or more nodes transmit simultaneously,
  - collision if node receives 2 or more signals at the same time.
- Goal:

Given: broadcast channel of rate R bps,
1. Collision Free
2. Efficient: when one node wants to transmit, it can send at rate R (gets entire bandwidth)
3. Fairness: when M nodes want to transmit, each can send at average rate R/M.
4. Fully Decentralised: no special node to coordinate transmissions (reduces chance of failure)

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

- Channel Partitioning Protocols: TDMA
  - Time Division Multiple Access
  - Access to channel in "rounds"
  - Each node gets fixed length time slots in each round.
    - Length of time slot = data frame transmission time
  - Eg.
    - 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.



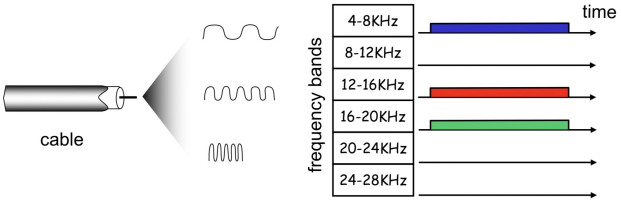
Wasted slot (unless all nodes are active)

Divide channels into fixed pieces, allocate piece to node for exclusive use

✓ Collision Free
✗ Efficiency - unused slots go idle, max throughput for a node is R/N.
✓ Fairness - perfectly fair
✓ Decentralised - only requires clocks to be synchronised.

- Channel Partitioning Protocols: FDMA
  - Frequency Division Multiple Access
  - similar to TDMA

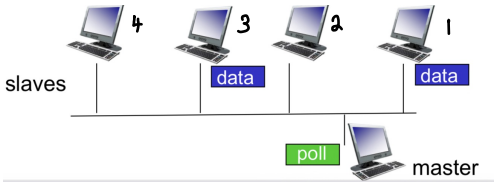
- Channel spectrum is divided into frequency bands.
- Each node is assigned a fixed frequency band.
- Unused transmission time in frequency bands go idle.
- Example: 6 nodes, 1, 3, 4 have frames, frequency bands 2, 5, 6 are idle.



"Taking Turns" Protocols: Polling

Nodes take turns to transmit

- 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 no. of frames**.
  - after node 1 transmits some frames, the master node tells node 2 it can transmit up to the maximum number of frames. (pass the baton)

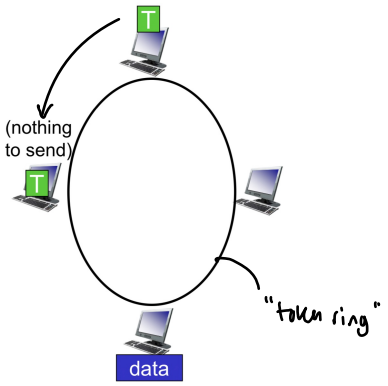


  - The procedure continues in this manner.
- eg. Bluetooth.

✓ Collision Free
✓ Efficiency - higher efficiency, overhead of polling. (no wastage)
✓ Fairness - perfectly fair
✗ Decentralized
- Master node is a single point of Failure.

"Taking Turns" Protocols: Token Passing

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



✓ Collision Free
✓ Efficiency - higher efficiency, ~R overhead of token passing
✓ Fairness - perfectly fair → each node visited equitably.
✓ Decentralized

✗ Downside:
- Token loss can be very disruptive
  - data frame loss
  - system bugs.
  - Added layers of **complexity**.
- Node failure can break the ring.

Random Access Protocols

- Channel is not divided, collisions are possible.
- "recover" from collisions.

- When node has data to send,
  - transmit at full channel data rate R
  - no a priori coordination among nodes.
- 2 or more transmitting node → "collision".
- Random access protocols specify:
  - how to detect collisions
  - how to recover from collisions.

Slotted Aloha

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 (different from TDMA)

- Time is synchronized at each node.

Operation:

- When a node has a fresh frame to send,
  - wait until the beginning of the next slot and transmit 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.

✗ Collision Free

? Efficiency - ✓ when only one node active, it gets a throughput of R.

✗ when many active nodes, the maximum efficiency is only 37%

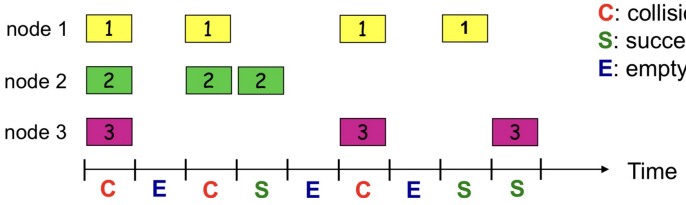
- Slots are wasted due to collision and slots being empty at times.

✓ Fairness

✓ Decentralized - only requires clocks to be synchronized.

Major Drawback: probability of collision in all subsequent time slots remain the same.

↳ can even increase if a new node starts transmitting.



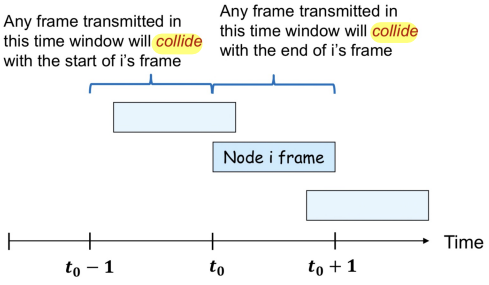
Pure (unslotted) ALOHA

Even simpler than Slotted ALOHA

- no time slots.
- no synchronization.

- Operation:
  - When the node has a fresh frame to send,
    - transmit 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.

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



- Efficiency decreases to 18%, otherwise same as Slotted ALOHA.

CSMA

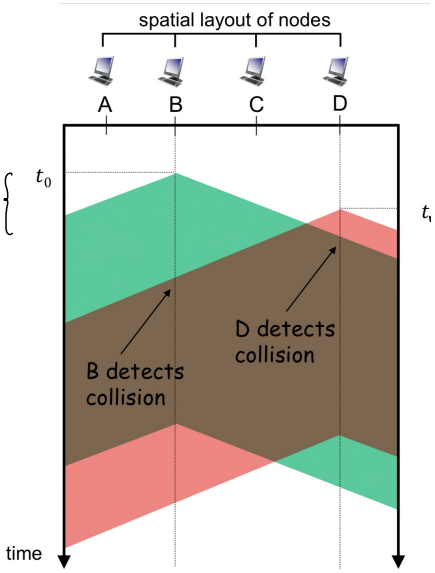
- Carrier Sense Multiple Access

- One major design flaw in ALOHA:
  - a node's decision to transmit is made independently of the activity of other nodes attached to the broadcast channel.
  - a node pays no attention to whether another node happen to be transmitting when it begins to transmit.

✗ Collision Free
✓ Efficiency
✓ Fairness
✓ Decentralized

- CSMA: listen before transmit
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy: defer transmission.

- CSMA Collisions:
  - Can still occur:
    - propagation delay — means 2 nodes may not hear each other's transmission immediately.

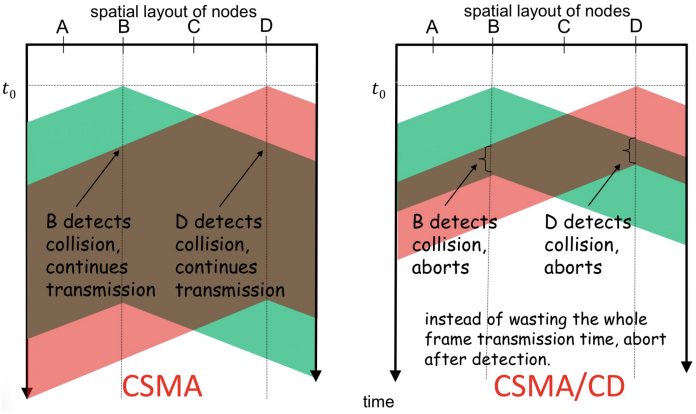


CSMA/CD

Collision Detection

- One major design flaw in ALOHA and CSMA
  - a node does not stop transmitting even when collision is detected.

- 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.
      - ⇓
      - adaptive based on the estimated current load.
        - more collision → heavier load.
        - longer back-off interval with more collisions.
          - ⇓



- Binary Exponential Backoff
  - After 1st collision:
    - choose K at random from $\{0, 1\}$, $p = 1/2$.
    - wait K time units before retransmission.
  - After 2nd collision:
    - choose K from $\{0, 1, 2, 2^2-1\}$, $p = 1/4$
    - wait K time units before retransmission.
  - After m^{th} collision:
    - choose K at random from $\{0, 1, \ldots, 2^m-1\}$, $p = 1/2^m$

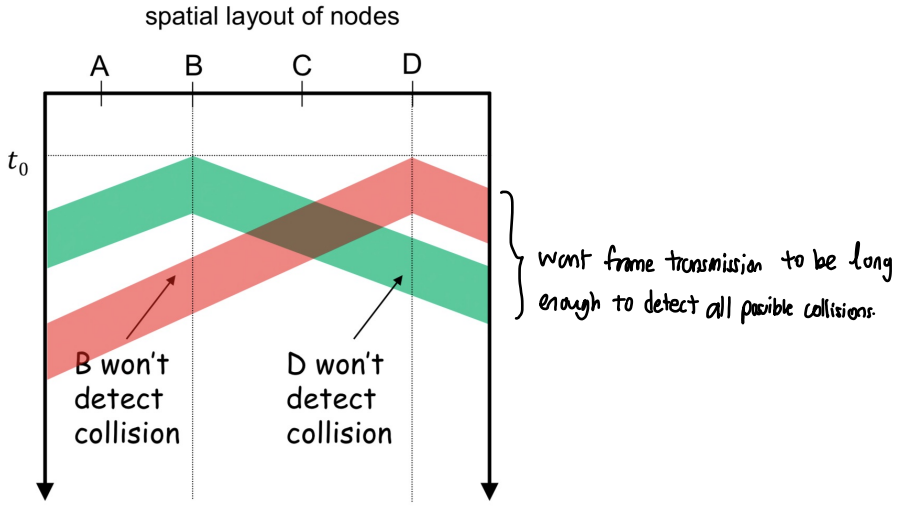
decrease probability of retransmission, ∴ ↓ collision chance.

"Frame time": for Ethernet, 1 time unit is set as 512 bit transmission times.

- Property: retransmission attempts to estimate 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!



- ∴ Ethernet, for example, requires a minimum frame size of 64 byte.

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

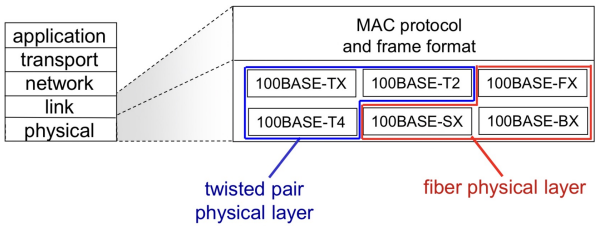
MAC Address

- 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.
- Typically 48 bits burned in NIC ROM (Read-Only Memory). → 6 bytes address.
  - Sometimes software settable
  - Eg. 5C - F9 - DD - EB - E3 - D2 : hexadecimal (base-16) notation.
    (5C: 1 byte.)

  MAC address allocation is administered by IEEE.
  - The first 3 bytes identifies the vendor of an adapter
- Broadcast Address: FF-FF-FF-FF-FF-FF
- Check w/ ifconfig (Linux)

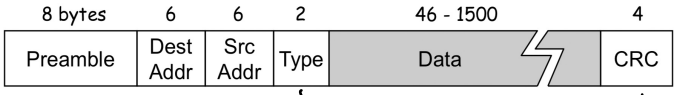
LAN

- LAN is a computer network that interconnects computers within a geographical area such as office building or university campus. (every node is a single hop range)
- LAN technologies:
  - IBM Token Ring (IEEE 802.5)
  - Ethernet (IEEE 802.3)
  - Wi-Fi (IEEE 802.11)
  - Others
- Ethernet:
  - "Dominant" wired LAN technology
  - Developed in 1970s, standardized in 1978.
  - Simpler and Cheaper than token ring and ATM.
  - Ethernet standards:
    - Different speeds – 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps.
    - Different physical layer media – cable, fibre optics.
    - MAC protocol and frame format remain unchanged.



Ethernet Frame Structure

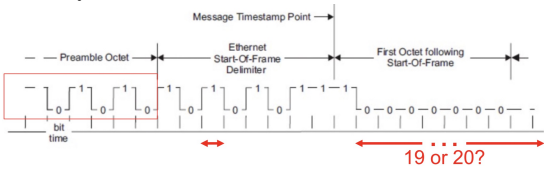
- To send an IP datagram from one host to another, on the same Ethernet LAN,
  → NIC (adapter) encapsulates IP datagram in Ethernet frame.



Type: indicates higher level protocol (network layer: IP, ARP ...)
permits Ethernet to multiplex network-layer protocols.
→ analogous to: protocol field (network layer datagram)
port-number fields (transport layer segment)

CRC: CRC-32, corrupted frame will be dropped.

- 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.
- Data:
  - The maximum size is 1500 bytes.
    - The max size is the link MTU (Maximum Transmission Unit) in IP fragmentation
  - The minimum size is 46 bytes.
    - The min size is to ensure that a collision will always be detected.
- Preamble:
  - 7 bytes with pattern 10101010 (AA_{Hex}) → acts like a "wake up"
  - Followed by 1 byte with pattern 10101011 (AB_{Hex})
    - aka "start of frame"
  - used to synchronize receiver and sender clock rates.

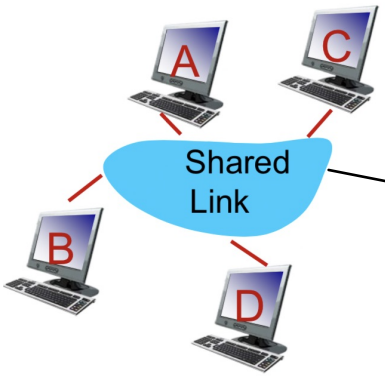


- The preamble provides a "square wave" pattern that tells the receiver the sender's clock rate
  - it tells the receiver the width of a bit
  - which is important if there is a long string of bits of the same value, e.g., 19 or 20 zeros.

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 (eg. TCP), otherwise, dropped data is lost.
- Ethernet's multiple access protocol:
  - CSMA/CD with binary (exponential) backoff
    ↳ retransmission on collision.

Switch



How to interconnect the nodes to create this shared link?

hub - cheap, slow
switch - expensive, fast

Bus Topology

- The original Ethernet LAN used a coaxial bus to interconnect the nodes.
- 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.

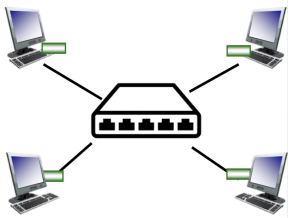
Ethernet with bus topology

Star Topology

- prevalent today

- Hub
  - popular in late 1990s
  - 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
      - boost its energy strength, and
      - transmit the bit onto all the other interfaces.

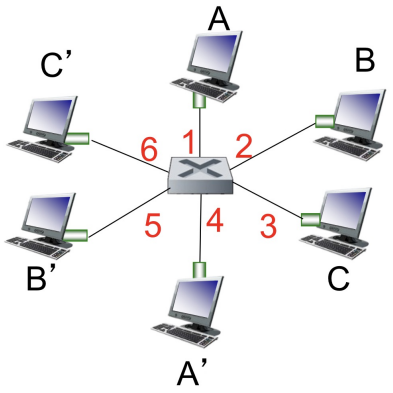
      } "amplifier"
  - Advantages
    - cheap
    - easy maintainance
      - modular design of the network.
  - Drawbacks
    - very slow and not ideal for larger networks
      - Due to collisions.



Ethernet with star topology

Switch

- popular since early 2000's
- nodes are directly connected to a switch
- A layer-2 (link layer) device
  - works and acts on frames rather than individual bits.
    - examines incoming frame's MAC address
      - selectively forward frame to one-or-more outgoing links.
  - no collision → uses CSMA/CD to access link.
  - store-and-forward packet switch
- Transparent
  - hosts are unaware of presence of switches.
- Plug-and-play (self-learning)
  - switches do not need to be configured.

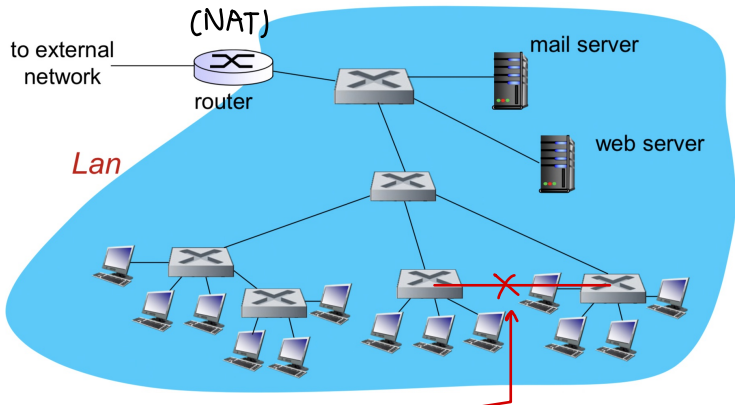


switch with six interfaces (1,2,3,4,5,6)

- Multiple simultaneous transmissions
  - Nodes have dedicated, direct connection to switch.
  - switches buffer packets.
  - Ethernet protocol used on each incoming link
    - but no collisions!
  - Switching:

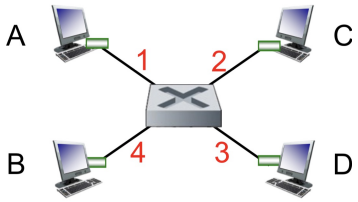
    A to A' and B-to-B' can transmit simultaneously, without collision.

- Interconnecting switches
  - switches can be connected in hierarchy.



- as long there are no loops ...

Selective Forwarding.



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

How does switch know A is reachable via interface 1?

- Each switch has a switch table (similar to routing table)
  - Entry format:

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

(eg. 1,2,3,4...)

Self-learning:

- Switch learns which hosts can be reached through which interface.

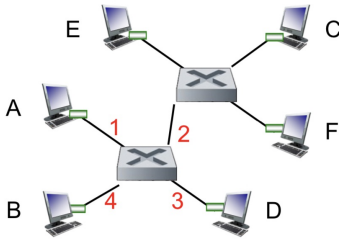
MAC addr	Interface	TTL
A	1	60
D	3	60

Switch table (initially empty)

- 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 the switch table.

Frame filtering / forwarding:

- Forwarding algorithm:
  1. Receive a frame to A on interface 4
     ↳ Forward to interface 1
  2. Received a frame to D on interface 1
     ↳ Forward to interface 2, 3, 4 (all except 1)
  3. 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

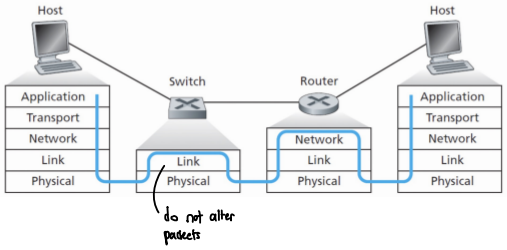
When frame received at switch:

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

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

} LAN, restricted to geographical location



Switch	Router
Layer-2 device	Layer-3 device
Self-learning	Need manual configuration
Forward link layer frames	Forward IP datagrams
Used in a subnet	Used to connect subnets

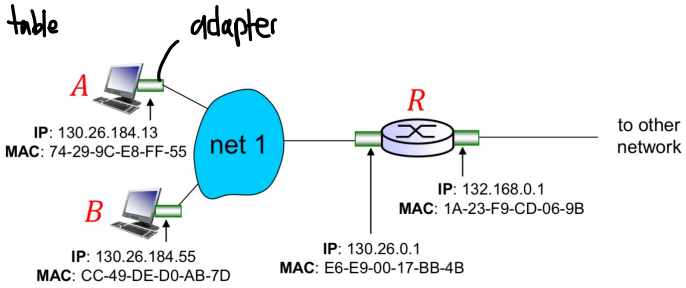
ARP

- Address Resolution Protocol. (arp in cmd)
- 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)

- 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

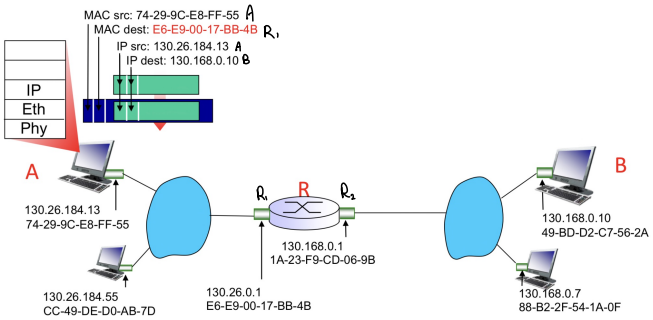


      1. Create a frame with B's MAC address and send it.
      2. Only B will process this frame.
      3. Other nodes may receive but will ignore this frame.
    - ② If A is not aware of B's MAC address
      1. A broadcasts an ARP query packet, containing B's IP address.
         - Dest. MAC addr set to 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 addr.
         - 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"

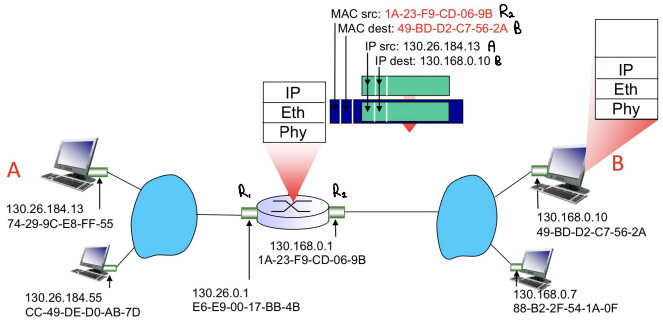
  nodes create their ARP tables without intervention from network administrators.

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.



- frame sent from A to R
- frame received at R, datagram removed, passed up to IP.
- 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 no change, MAC change at router.

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

(globally unique)

- ARP [RFC 826] resolves the mapping from network layer (IP) address to link layer (MAC) address.

Qns - Link Layer

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?

   IP datagrams in the same TCP connection can take different routes in the network, and therefore arrive at receiving host out of order. TCP is still needed to sort out received data in the correct order before passing them to application.
   Also, IP datagrams can be lost due to routing loops, equipment failures, etc. For example, what if a router holding a frame crashes?

5. Nodes A and B are accessing a shared medium using CSMA/CD protocol, with propagation delay of 245 bit times between them (i.e., propagation delay equals to the amount of time to transmit 245 bits onto the link). 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**.

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

   The latest time B can begin transmission is before the signal from A reaches B, which is when $t = 244$ bit time.

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

   Suppose B begin transmission at $t = 244$ bit time. Signal propagates to A at $t = 244 + 245 = 489$ bit time. A is able to detect collision before it finishes transmission (at $t = 512$ bit time).

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?

   After 5th collision, NIC will choose K at random from {0, 1, 2, …, 2^5-1}. The chance to choose $K = 4$ is 1/32.

   NIC will wait for 4 * 512 / 10^7 = 204.8 microseconds.

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).

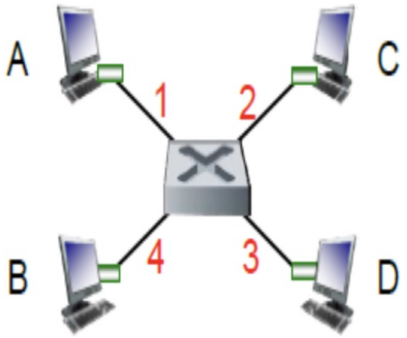
   Suppose that the following events happened in sequence,

   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

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.

   1) **A creates a frame with destination MAC address CC-49-DE-D0-AB-7D (B's address is found in ARP table).**
   2) **This frame travels to switch S and is forwarded towards B (interface to B is found in switch table).**

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

   1) **A broadcasts an ARP query packet, with destination MAC address FF-FF-FF-FF-FF-FF.**
   2) **Switch S forwards this ARP query packet to both B and R since destination MAC address is a broadcast address.**
   3) **R will ignore this ARP query packet but B will reply to A. Switch S forwards the reply frame towards A (interface to A is found in switch table).**
   4) **Subsequently A can send IP datagram to B as in part a).**

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

   1) **A needs to issue an ARP query to know the MAC address of B.**
   2) **The query packet travels to switch S and is forwarded to both B and R. Switch S learns A is reachable via the interface query packet arrives at.**
   3) **R will ignore this ARP query packet but B will reply to A. Switch S forwards the reply frame towards A (interface to A is found in switch table). Switch S learns B is reachable via the interface reply frame arrives at.**
   4) **Subsequently A can send IP datagram to B as in part a).**

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.

   1) **A creates a frame with destination MAC address E6-E9-00-17-BB-4B (R's address is found in ARP table).**
   2) **This frame travels to switch S and is forwarded towards R (interface to R is found in switch table).**
   3) **R checks the destination IP of the datagram and decides to forward it towards external network. It encapsulates the IP datagram in a new frame with source MAC address 1A-23-F9-CD-06-9B (dest MAC address not mentioned in question) and sends it through the interface towards external network.**

3. Which services listed below are NOT usually part of the link layer?

(1 mark) ✗
You scored 0 / 1 mark

- (●) Framing.
- ( ) Link access control.
- ( ) Reliable delivery.
- ( ) Error detection/correction.
- ( ) Name-to-IP resolution.

6. Ethernet uses the CSMA/CD medium access protocol with an exponential backoff algorithm. Why is the factor *K* in the exponential backoff chosen randomly?

(1 mark) ✓
You scored 1 / 1 mark

- (●) If all computers would follow a deterministic algorithm to select K then there would be a chance that two computers would always select the same *K* and so they would continuously collide and could never transmit any data.
- ( ) It is easier to choose a random number than to compute *K*.
- ( ) There is no good reason of why *K* is chosen randomly. We could just come up with another algorithm.

8. The payload of an Ethernet frame is minimal 46 to maximal 1,500 bytes. Which of the following statements is **FALSE**?

(1 mark) ✗
You scored 0 / 1 mark

- ( ) A maximal Ethernet frame payload size makes it easier to implement efficient data buffer management algorithms in the NICs, switches and routers.
- (●) There is a minimum size of 46 bytes to make sure that collisions can be detected.
- ( ) A maximal Ethernet frame payload size helps to make sure that senders do not "hog" shared links and other senders get a chance to transmit as well.
- ( ) If we want to transfer a large file (several MB) over an Ethernet connection and want to achieve high throughput (i.e., high MB/s), then it is best to use a lot of short frames, each with a small payload.

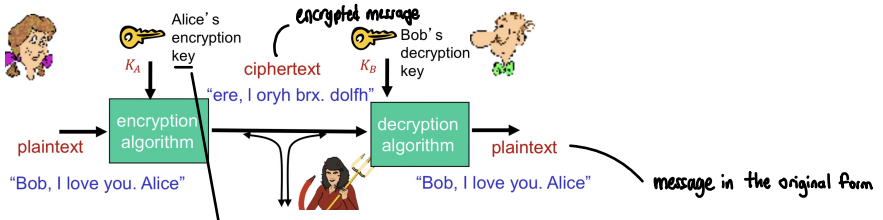
Network Security

- Hosts: Web browser/server for electronic transactions,
  online banking client/server,
  DNS servers,
  Routers exchanging routing table updates.
- Intruder:
  - eavesdrop. intercept messages
  - actively insert message into connection
  - impersonation: can fake (spoof) source address in packet (or any field in packet).
  - hijacking: "take over" ongoing connection by removing sender or receiver, inserting oneself in place (Man in the middle)
  - denial of service: prevent service from being used by others (eg. by overloading resources).

- What is Network Security?
  1. Confidentiality
     - only sender, intended receiver should "understand" message contents.
  2. Authentication
     - sender, receiver want to confirm identity of each other
  3. Message Integrity
     - sender, receiver want to ensure message not altered (in transit, or afterwards) without detection.
  4. Access and Availability
     - services must be accessible and available to all users.

  } cryptographic techniques.

- Principles of Cryptography
  (hidden, writing)
  - Techniques:
    - allow a sender to disguise data so that an intruder can gain no information from the intercepted data.
    - allow the receiver to recover the original data from the disguised data.



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
  - $K_B(K_A(m)) = m$

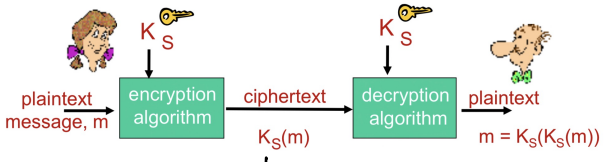
K_A: is an abuse of notation, The Encryption and Decryption algorithm may not be same

Types of Cryptography

1. Symmetric Key Cryptography
   - Sender and receiver use the same key
   - $K_A = K_B$

2. Asymmetric Key Cryptography (aka Public Key Cryptography)
   - Sender and receiver use different key
   - $K_A \neq K_B$

Symmetric Key Cryptography (Confidentiality)



A and B shares same key
↳ need to decide on the common key prior to communication via some other secure means, like face to face meeting.

- Caesar's cipher
  - A form of Substitution cipher: substituting one thing for another.
  - Fixed shift of alphabet.
    e.g., right shift by 3:

```
abcdefghijklmnopqrstuvwxyz
defghijklmnopqrstuvwxyzabc
```

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

key → 25 possible values, easy to break with brute force search.

- Monoalphabetic cipher
  - Substitute one letter for another (Substitution cipher):

```
abcdefghijklmnopqrstuvwxyz
mnbvcxzasdfghjklpoiuytrewq
```

Weakness:
- each letter has only one mapping
  ↳ use multiple mappings (Polyalphabetic encryption)

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

key → mapping from set of 26 letters to set of 26 letters, 26! mappings possible. ($\sim 10^{26}$, not as much ..)
↳ can break it with Statistical Analysis

- *letters e (13%) and t (9%) are the most frequent letters*
- *knowing that particular two-and three-letter occurrences of letters appear quite often together (for example, "in," "it," "the," "ion," "ing,"*
- *If the intruder has some knowledge about the possible contents of the message, then it is even easier to break the code.*
  - *if Trudy the might suspect that the names "bob" and "alice" appear in the text.*
  - *and had a copy of the example ciphertext message above, then she could immediately determine seven of the 26 letter pairings*

Breaking an encryption scheme

- **Ciphertext only** attack: Intruder has ciphertext to analyse.

  Known-plaintext attack: Intruder has plaintext corresponding to ciphertext.
  - eg. In monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o.

- **Chosen-plaintext** attack: Intruder can get ciphertext for chosen plaintext.
  - eg. In monoalphabetic cipher, Trudy gets Alice to send "The quick brown fox jumps over the lazy dog".

Polyalphabetic encryption

- n substitution ciphers, $C_1, C_2, C_3, \ldots, C_n$.
- define a cyclic pattern: (encryption key!)
  - n=4: C_1, C_3, C_4, C_3, C_2
  - → for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern.
  - eg. dog; d from C_1, o from C_3, g from C_4.
- 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: $C_1 C_3 C_3 C_2$

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
  - eg. 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
  - eg. k=3
    - Input: 010|110|001|111
    - Output: 101000111001

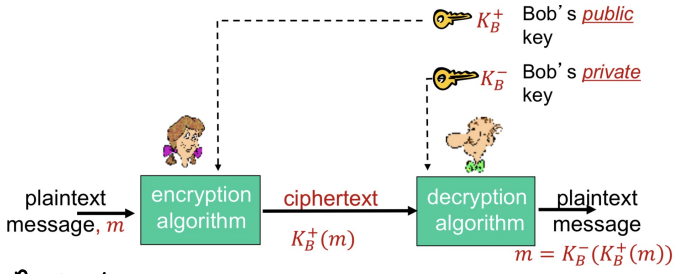
Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

$2^3! = 8!$

- Number of keys: $2^k!$
- **DES**: Data Encryption Standard
  - 56 bit symmetric key, 64-bit block.
  - 3DES: more secure by encrypting it 3 times with 3 different keys.
- **AES**: Advanced Encryption Standard
  - Symmetric key NIST standard, replaced DES
  - 128 bit blocks; 128, 192 or 256 bit keys.

RSA (confidentiality)

- Rivest, Shamir, Adleman algorithm
- Symmetric key crypto drawback: requires receiver know shared secret key.
- Public key cryptography



- Requirements:
  1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
     $$m = K_B^-(K_B^+(m))$$
  2. given public key K_B^+, it should be impossible to compute private key K_B^-
- Notes:
  - 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
    $$(a \bmod n)^d \bmod n = a^d \bmod n$$
  - *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).

- RSA: Creating public/private key pair
  1. Choose 2 large prime numbers p, q.
  2. Compute $n = pq$, $z = (p-1)(q-1)$
  3. Choose e ($e < n$) 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. ($ed \bmod z = 1$)
  5. Public key is (n, e). Private key is (n, d).
     (n, e) = K_B^+, (n, d) = K_B^-

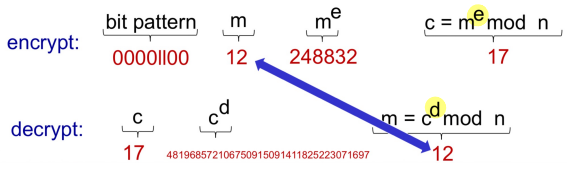
- RSA: encryption/decryption
  0. Given (n, e) and (n, d) as computed above,
  1. To encrypt message m ($m < n$), compute
     $$c = m^e \bmod n$$
  2. To decrypt received bit pattern, c, compute
     $$c^d \bmod n$$

  magic happens! $(m^e \bmod n)^d \bmod n = m$ (where $m^e \bmod n$ = c)

Example:

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.



- Useful property:
  $$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$
  (K_B^-: private key first; K_B^+: public key first.)

  $\longrightarrow$ $(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$
  $= m^{de} \bmod n$
  $= (m^d \bmod n)^e \bmod n$

- Exponentiation in RSA is computationally intensive.
- DES is at least 100 times faster than RSA, but needs prior knowledge of key K_S.
- ∴ use RSA to transfer symmetric key K_S (session key), use K_S in DES for encrypting data.

Message Integrity

- Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection.
- i.e. Error Detection (Checksum, Parity, CRC)

e.g. Internet checksum:
- produces fixed length digest (16-bit sum) of message
- is many-to-one

Consider the given message: "IOU100.99BOB"

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

different messages but identical checksums!

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

- It is easy to find another message with same checksum value
- checksum is designed to detect *accidental* errors not attacks!

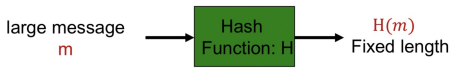
CRC:

Better than Checksum
- Yet poor
- Output is biased to the input
  - Minor changes in input produce minor changes in output

- E.g.
  - "Steven has fifteen white tables." and "Maria has nine red beds."
    - Both have CRC32 checksum = *248210933*
  - "Joe has fourteen magenta things." and "Lars has thirteen black balls."
    - Both have CRC32 checksum = *93832682*

∴ Cryptographic Hash Function

- Hash function: H() that takes an input m and produces fixed-size message digest (fingerprint)
  - many-to-one.
  -

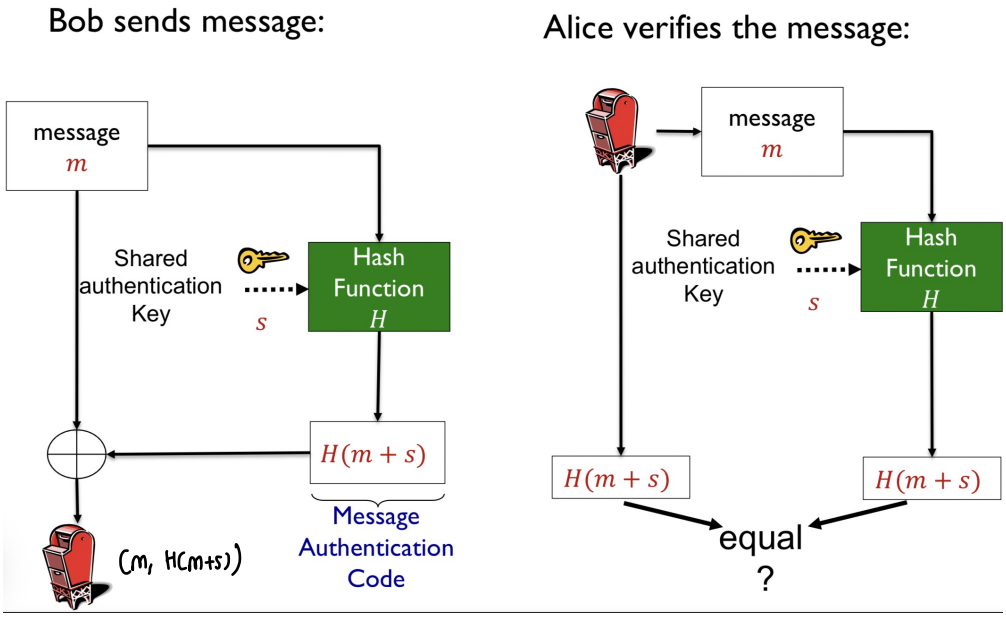


- Cryptographic Hash function:
  - Is a hash function such that it is computationally infeasible to find any 2 different messages x and y such that H(x) = H(y).
  - i.e. Computationally infeasible to substitute one message for another message.
  - MD5 hash function widely used (RFC 1321)
    - computes 128-bit message digest. ← md5sum
  - 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
  - Generate short, fixed length outputs (or digests) – 128 bits
    ↳ useful for longer inputs (fingerprint)
  - A small change in the input should result in a large change in hash output.

Message Authentication Code

- To ensure Message Integrity, not enough to just send (m, H(m)), attacker can just replace it with (m', H(m')), receiver has no way of detecting it.

- The sender and receiver share a "Authentication key" s
- To ensure Message integrity:
  - Send $(m, \underbrace{H(m+s)}_{\text{Message Authentication Code}})$
- 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



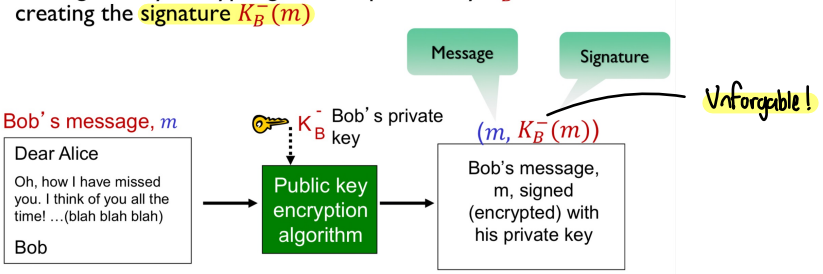
Authentication

- **Digital Signatures**
  - analogous to **hand written signatures**.
  - Sender (Bob) digitally **signs** the document, establishing he is the document owner/creator.
  - Signature must be:
    - **Verifiable**:
      - Recipient (Alice) can check if the signature and the message is generated by Bob.
    - **Unforgable**:
      - No one, other than Bob should be able to generate the signature and the message.

- Exploit RSA property: $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$

simple digital signature for message m:

- Bob signs m by encrypting with his private key K_B^- creating the signature $K_B^-(m)$



- Suppose Alice receives msg m, with signature: $m, K_B^-(m)$.
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$. → note: the order is inverted.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that: — Verifiable!

- Bob signed m
- no one else signed m
- Bob signed m and not m'

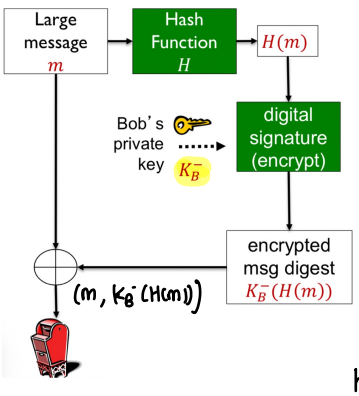
non-repudiation:

✓ Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m

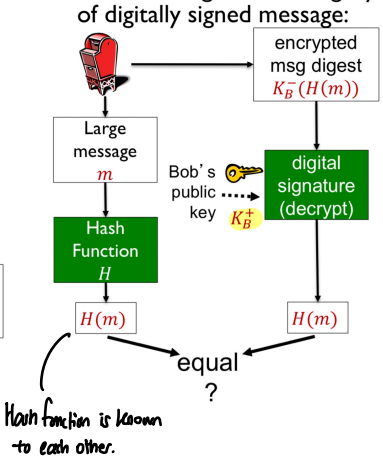
Optimisation:

- Produce a signature of H(m):

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:

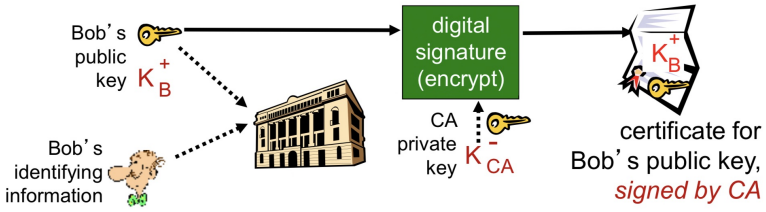


Hash function is known to each other.

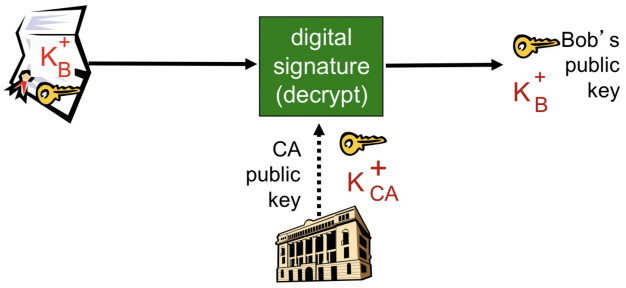
Public Key Certification

- How to know anyone's public key?
  - Certification Authorities (CA) who maintain a public database of everyone's public key.
  - CA signs its messages.
    - ↳ CA's public key?
      - We maintain a list of CAs trusted a priori.
      - OS has a list of "Trusted Root Certification Authorities".

- *certification authority (CA):* binds public key to particular entity, *E*.
- *E* (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.
  - CA creates certificate binding *E* to its public key.
  - certificate containing *E*'s public key digitally signed by CA
    - CA says "this is *E*'s public key"

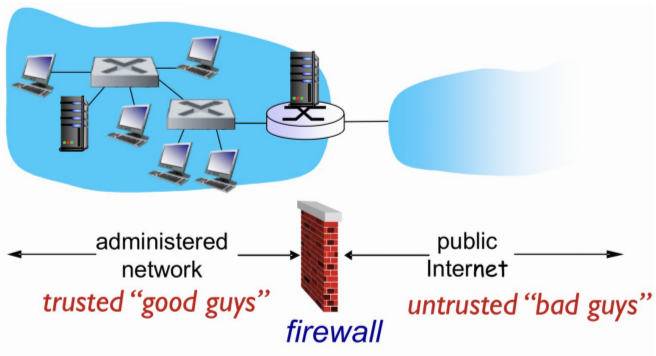


- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere).
  - apply CA's public key to Bob's certificate, get Bob's public key



Access and Availability

- Firewalls
  - isolates organization's internal net from larger internet, allowing some packets to pass, blocking others.



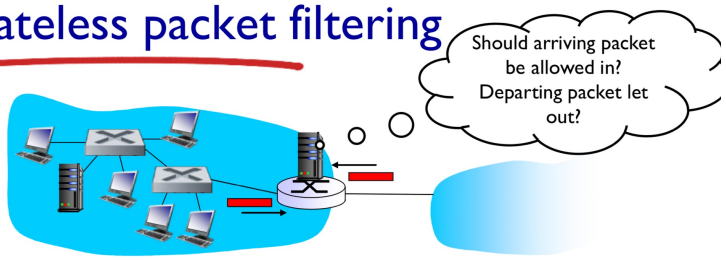
- Why?
  - ❖ Prevent denial of service (DoS) attacks:
    - ❖ SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections
  - ❖ Prevent illegal modification/access of internal data
    - ❖ e.g., attacker replaces CIA's homepage with something else
  - ❖ Allow only authorized access to inside network
    - ❖ set of authenticated users/hosts

- 3 types:
  1. Stateless packet filter
  2. Stateful packet filter
  3. application gateways

Limitations of firewalls

- *IP spoofing:* router can't know if data "really" comes from claimed source
- Can become a bottleneck
- *tradeoff:* degree of communication with outside world, level of security
- Many highly protected sites still suffer from attacks

Stateless packet filtering



- internal network connected to Internet via *router firewall*
- router *filters packet-by-packet,* decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

- *example 1:* block incoming and outgoing datagrams with IP protocol field = 17
  - *result:* all incoming, outgoing UDP flows are blocked

- *example 2:* block inbound TCP segments with ACK=0.
  - *result:* prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Policy	*Firewall Setting*
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists

ACL: table of rules, applied top to bottom to incoming packets: (action, condition) pairs

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Summary

- ❖ *Confidentiality:*
  - ❖ Substitution Cipher
  - ❖ Symmetric Key
  - ❖ Public Key
- ❖ *Authentication:*
  - ❖ Signature
- ❖ *Message integrity:*
  - ❖ Cryptographic Hash
- ❖ *Access and availability:*
  - ❖ Firewall

Qns - Network Security

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?

    There are $N * (N-1)/2$ pairs of people and each pair needs to share a symmetric key. The total number of keys is $N * (N-1)/2$.

    b) Public key encryption is used in each communication?

    With public key encryption, each person has a public key which is known to all, and a private key which is secret and known to the user only. There are thus $2 * N$ 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.

    A file is broken into a number of blocks of identical size. For each block, a hash is calculated (e.g., using MD5 or SHA-1). The hashes for all of the blocks are saved in the `.torrent` file.

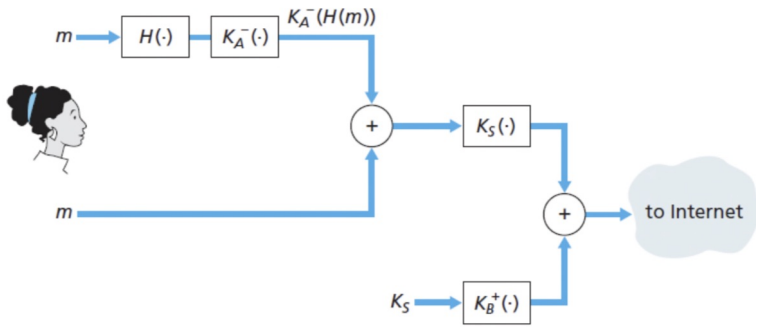
    When a block is downloaded, a peer calculates the hash of this block and compares it to the recorded hash in the `.torrent` file. If the two hashes are equal, this block is error-free. Otherwise, the block is bogus and should be discarded.

5. A digital document that has been signed by Bob with his digital signature has the following properties:

(1 mark)
You scored 1 / 1 mark

- [ ] It is now encrypted and cannot be read by anybody other by a specific receiver (say Alice).
- [x] We can verify that Bob is the person who signed the document.
- [x] We can verify that the document has not been changed.
- [ ] We can verify that the document does not contain any mistakes.

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):



1. generates a random session key K_S
2. encrypts the session key with Bob's public key K_B^+, obtaining K_B^+(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

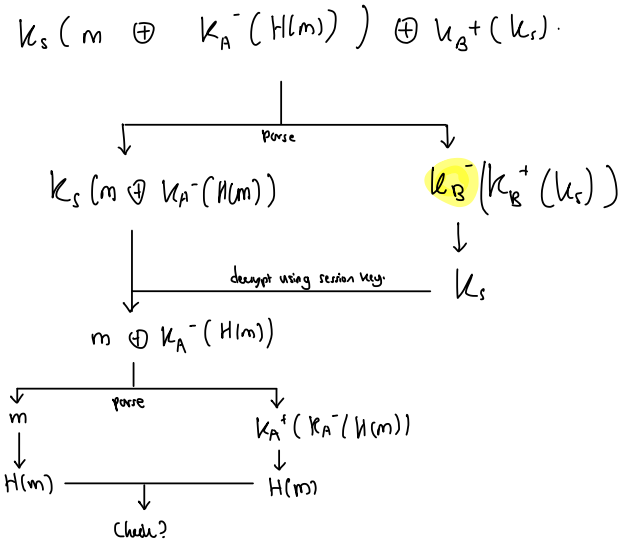
Show what Bob has to do to verify that m is indeed crafted by Alice and has not been modified during transmission.

NOTE: symmetric key crypto (K_S) is used to encrypt m, instead of public key crypto because public key crypto is much slower.

Since only Bob has his private key, the intruder cannot decrypt $K_B^+(K_S)$ and therefore cannot get the session key K_S. Without K_S, the intruder cannot decrypt m.

The intruder cannot impersonate Alice, since Alice digitally signed the message with her private key $K_A^-(H(m))$. Otherwise, Bob cannot get the right $H(m)$ when trying to decrypt with Alice's public key.

The intruder cannot tamper with m, since modifying m would cause its hash to be different and Bob would detect this.



7. You have used a symmetric encryption (like DES) for one of your applications with a key length of 56 bits. Now you have found that you can brute-force break the key on your laptop in about 20 hours. So you decide to increase the key length to 64 bits. Roughly how long would you expect a brute force attack to take with the longer key on the same computer?

(1 mark)
You scored 1 / 1 mark

- Approximately 28 hours.
- Approximately 160 hours.
- **Approximately 213.3 days.**
- Approximately 22.86 hours.

Multimedia Networking

- We defines a multimedia application as any network application that employs audio or video.

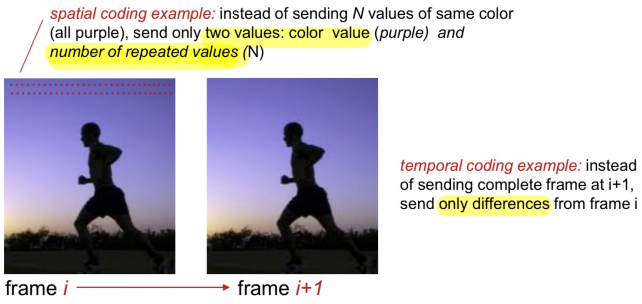
- Application and Transport layers.

- OTT (Over-the-top): user existing internet infrastructure (TCP/IP protocol stack) to send its data (doesn't use anything else)

- 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)
    - · eg. YouTube.
  - * · Conversational ("2 way live") voice/video over IP.
    - · Interactive nature of human-to-human conversation limits delay tolerance.
      - · Delay more than 400ms, intolerable.
    - · eg. Zoom.
  - · Streaming live ("one-way live") audio, video.
    - · Typically done with CDNs (Content Distribution Network.)
    - · eg. live sporting event

- Video: sequence of images displayed at a constant rate, high bit rate.
- Digital image: array of pixels, each pixel represented by bits.

∴ To reduce data usage, we compress the 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)

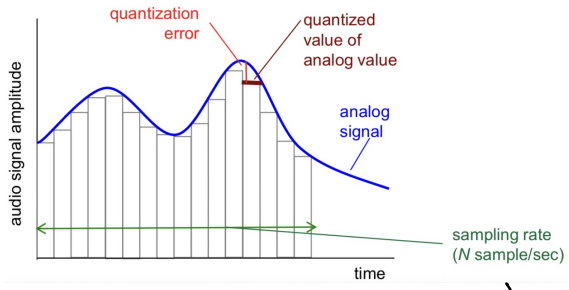


- * · CBR (constant bit rate): video encoding rate fixed.
  - · Not responsive to the complexity of the video
  - · Need to set bitrate relatively high to handle more complex segments of video
  - · The consistency of CBR makes it well-suited for real-time encoding.
    - · For real-time live streaming
- VBR (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes.
  Best suited for on-demand video due to longer time to process the data.

examples:
- MPEG 1, CD-ROM, 1.5 Mbps
- MPEG 2, DVD, 3-6 Mbps
- MPEG 4 / H.264, < 2 Mbps
- H.265, 4K video, > 10 Mbps.

Audio

- Analog audio signal
  - sampled at constant rate
  - Telephone: 8000 samples/sec
  - CD music: 44100 samples/sec
- Each sample quantised, i.e. rounded
  - Each quantised value represented by bits, e.g. 8 bits for 256 (2^8) values.

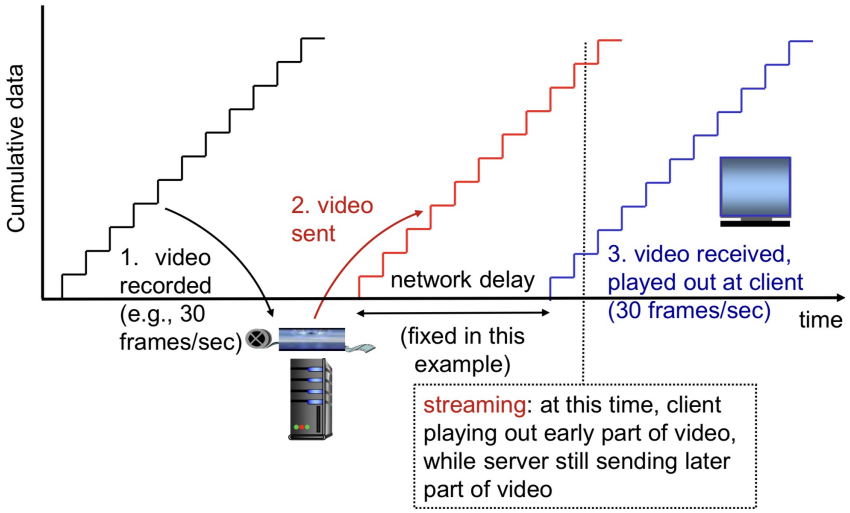


higher → better

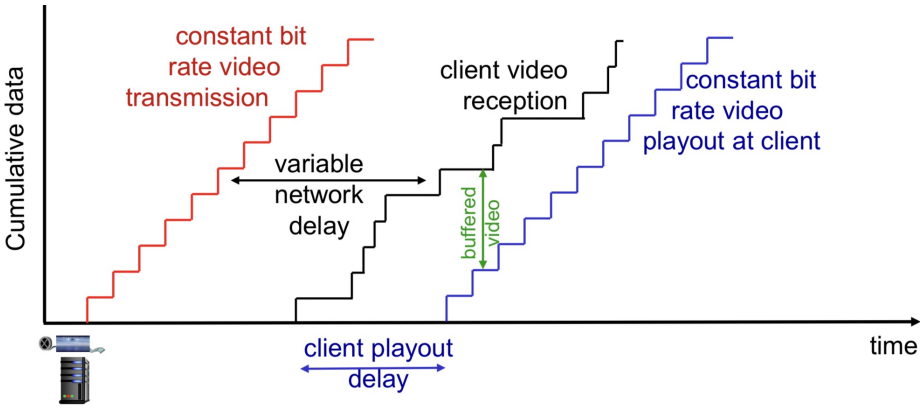
  - Example: 8000 samples/sec,
    256 quantised values (8 bits): 64000 bps.
  - Receiver converts bits back to analog signal (DAC):
    - Some quality reduction.

digital-to-analog converter.
ADC → analog-to-digital converter.

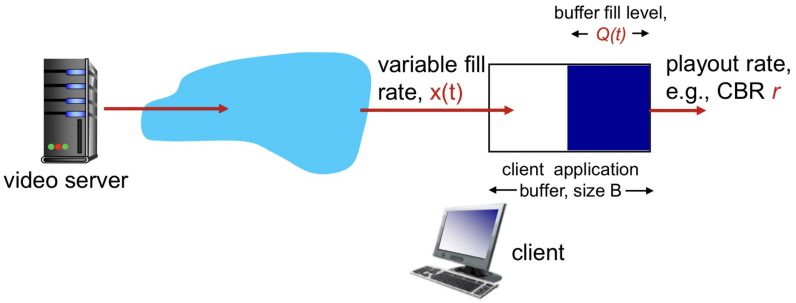
  - example rates:
    - CD: 1.411 Mbps
    - MP3: 96, 128, 160 kbps
    - Internet telephony: 5.3 kbps and up
- Streaming stored video:



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



- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter



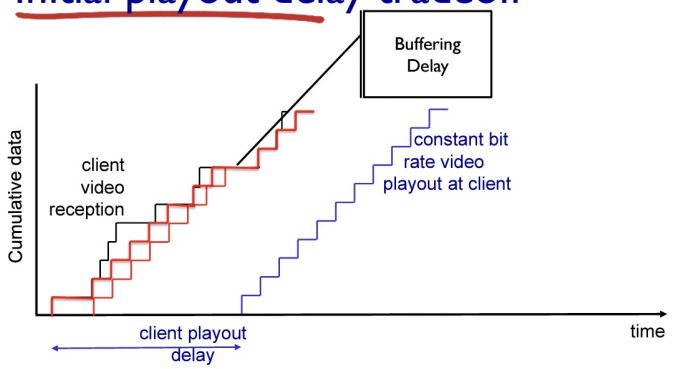
1. Initial fill of buffer until playout begins at t_p
2. Playout begins at t_p.
3. Buffer fill level varies over time as fill rate $x(t)$ varies and playout rate r is constant.

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

- $\bar{x} < r$: buffer eventually empties (causing freezing of video playout until buffer fills again)
- $\bar{x} > r$: buffer will not empty provided initial playout delay is large enough to absorb variability in $x(t)$
  - initial playout delay tradeoff:
    - buffer starvation less likely with larger delay.
    - but larger delay until user begins watching.

i.e. buffer empties.

Initial playout delay tradeoff



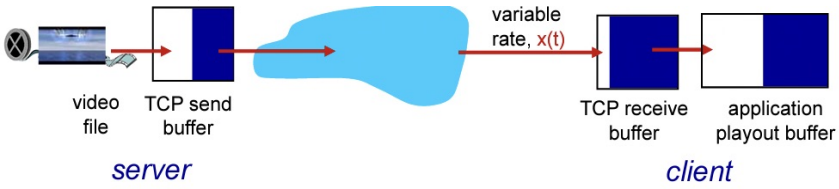
Streaming Multimedia

UDP

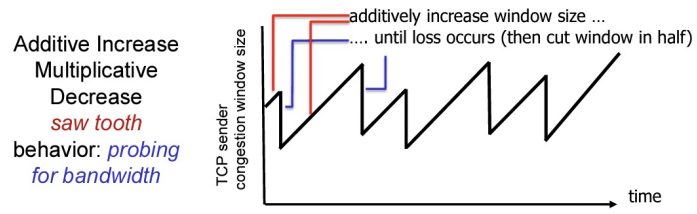
- Server sends at rate appropriate for client
  - usually send rate = encoding rate = constant rate.
    - push-based streaming (server push)
  - UDP has no congestion control
    - Hence transmission without rate control restrictions
- Short playout delay (2-5 seconds) to remove network jitter
- Error recovery: application-level, time permitting.
- Video chunks encapsulated using RTP (Real Time Protocol)
- Control Connection is maintained separately using RTSP (Real Time Streaming Protocol)
  - Is used for establishing and controlling media sessions between endpoints.
  - Client issue commands such as play, record, pause.
- Drawbacks
  - Need for a separate media control server like RTSP, increases cost and complexity.
  - UDP may not go through firewalls.

HTTP

- Multimedia file retrieved via HTTP GET.
  - pull-based streaming (client pull)
- Send at maximum possible rate under TCP.



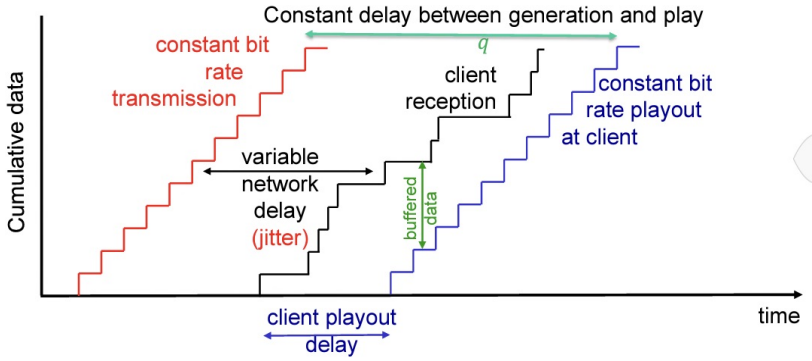
- Advantages
  - HTTP/TCP passes more easily through firewalls.
  - Network infrastructure (like CDNs and Routers) fine tuned for HTTP/TCP.
- Drawbacks
  - fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)
  - larger playout delay: smooth TCP delivery rate.



VoIP

- Voice over IP.
- 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**.

- Characteristics:
  - Speaker's audio:
    - alternating talk spurts, silent periods
    - pkts generated during talk spurts.
    - 20 msec **chunk** at 8 kbps: 160 bytes of data.
  - application-layer header is added to each chunk
  - **Chunk + header** encapsulated into TCP or UDP segment
    - application sends segment into socket every 20ms during talk spurt.
- **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:



- end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

Fixed playout delay

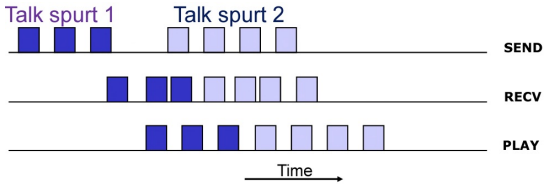
every chunk will have seq. no. & timestamp.

- receiver attempts to playout each chunk exactly q ms 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
- No value of q can guarantee an optimal performance
  We will eventually have a packet loss, or
  - We 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 20ms during talk spurt.



- Adaptively estimate packet delay (EWMA): exponentially weighted moving average.

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

d_i: delay estimate after ith packet
α: small constant, e.g. 0.1
$r_i - t_i$: time received - time sent (timestamp) = measured delay of ith packet

v_i: 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:

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

  - remaining packets in talk spurt are played out periodically

Recovery from packet loss

- Challenge: recover from packet loss given small tolerable delay between original transmission and playout.
  Use ACK/NAK
  → too slow, each ACK/NAK takes one RTT
- Use Forward Error Correction (FEC)
  → send enough bits to allow recovery without transmission. (i.e. 2-D parity).

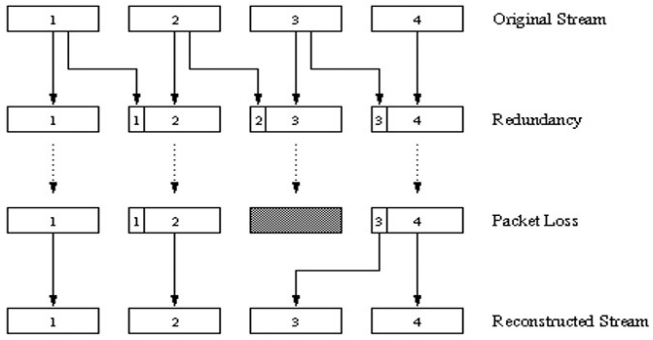
FEC

- 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.

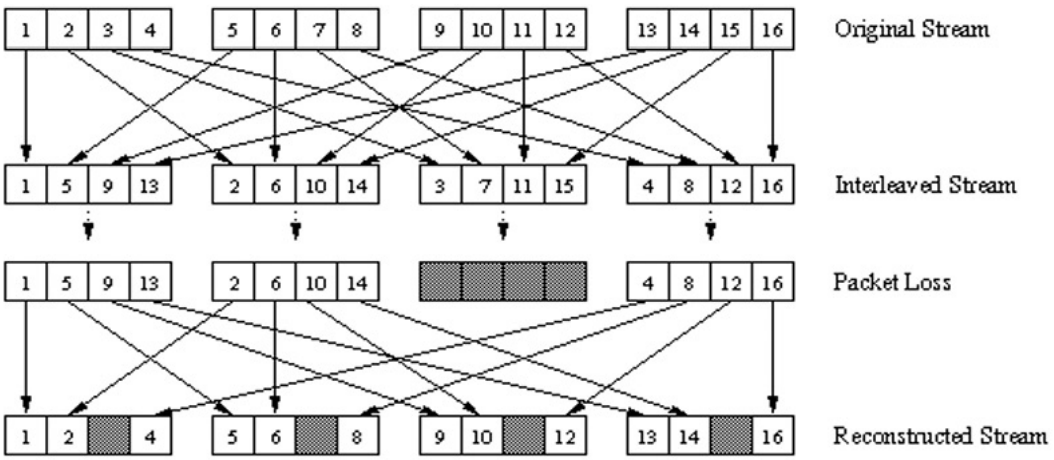
```
1 0 1 0 1 1
1 1 0 1 0 1
1 0 1 0 0 0
-----------
1 1 0 1 1 0
```

  - Drawback
    - Increasing bandwidth by factor 1/n
    - Playout delay is increased during packet loss.
      - Receiver waits for n+1 chunks before playout

- "piggyback lower quality stream":
  - send lower resolution audio stream as redundant information.
    - eg nominal stream at PCM at 64 kbps and redundant stream GSM at 13 kbps.
  - non-consecutive loss: receiver can conceal loss.
  - generalisation: can also append (n-1)st and (n-2)nd low-bit rate chunk.



- Interleaving to conceal loss:
  - Audio chunks divided into smaller units, eg four 5ms units per 20 ms 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.



DASH

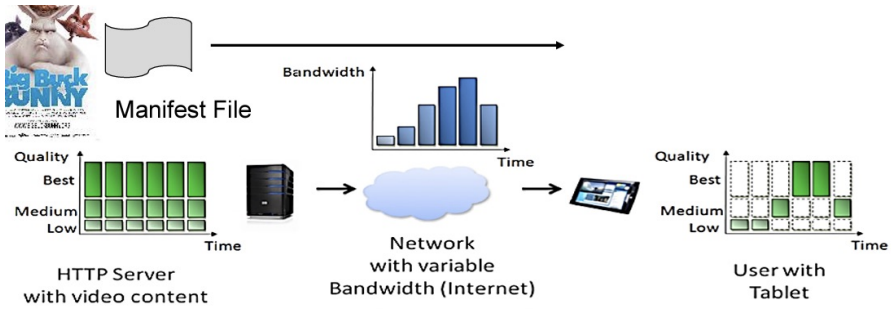
- Video-on-Demand (VoD) video streaming increasingly uses HTTP streaming.
  - Simple HTTP streaming just GETs a (whole) video file from an HTTP server.
- Drawbacks
  - Can be unreliable, needs large client buffer
  - All client receive the same encoding of video, despite the variation in the device/network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

  Dynamic Adaptive Streaming over HTTP. (DASH)

  - 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 that time).
    - "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).



- 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.
- 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 video) to hundred and thousands of simultaneous users?

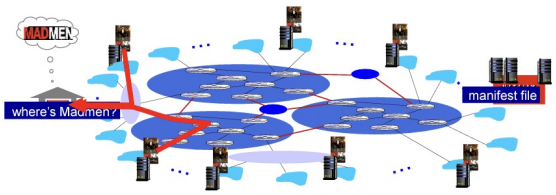
- 1 Single, large "mega-server"
  - Single point of failure.
  - point of network congestion
  - long path to distant client
  - multiple copies of video sent over outgoing link.

  } doesn't scale

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 Providers)
  - close to users.
  - used by Akamai, 1700+ locations.
- bring home. smaller number (10's) of larger clusters in IXPs (Internet Exchange Point) near (but not within) access networks.
  - used by Limelight.

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

Qns – Multimedia

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.
   Spatial redundancy: Redundancy within the same image, e.g. consecutive pixels with the same color. We could compress the video frame by sending just the color value and its count (instead of the color value *N* times).

   Temporal redundancy: Redundancy between multiple images, e.g. consecutive frames that are very similar and only have a small change between them. We could compress the second video frame by sending only its difference from the original frame (instead of the full uncompressed frame).

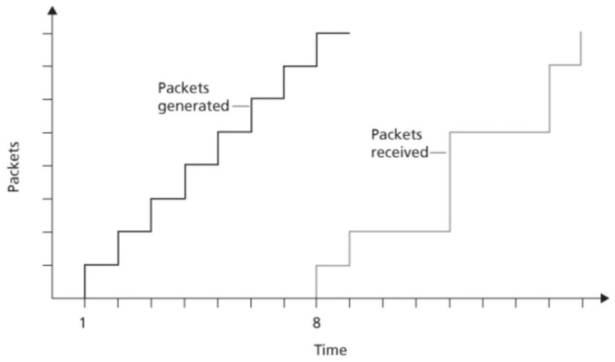
2. **[KR, Chapter 9, R3]** Suppose an analog audio signal is sampled 16,000 times per second, and each sample is quantized into one of 1,024 levels. What would be the resulting bit rate of the PCM digital audio signal?
   Each sample would require $\log_2 1024 = 10$ bits encode its data. Hence, sampling at 16,000 times per second would generate a signal of 160,000 bits per second = 160 kbps.

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?
   No, they are not the same. TCP and application are at different layers of the OSI model. TCP receive buffer (at transport layer) receives the TCP packets from the network layer, which are then processed and forwarded as HTTP packets to the client's application buffer (at the application layer). The client (e.g. Chrome browser), upon receiving these HTTP packets, then processes and renders the video playback to the user.

4. **[KR, Chapter 9, R10]** Why is a packet that is received after its scheduled playout time considered lost?
   When an application receives a packet that is *late*, i.e. received after its scheduled playout time, the application would not be able to play that packet anymore. Hence, from the perspective of the application, the packet has been lost.

5. In practice, RTP tends to be used over UDP while RTSP tends to be used over TCP. Why might this be so?
   RTP sends media flow (e.g. multiple video packets) while RTSP sends playback commands (e.g. "Play" and "Pause"). Hence, it may be preferred that RTP prioritizes speed over reliability (UDP), so that the client could playback the video with minimal delay (losing a few video packets in-between is a relatively small cost since its impact on playback experience is generally tolerable). On the other hand, it may be preferred that RTSP prioritizes reliability (TCP), so that we do not lose any of these commands requested by the client.

6. **[KR, Chapter 9, P11]** Consider the figure below (which is similar to Lecture 10 notes page 26). A sender begins sending packetized audio periodically at $t = 1$. The first packet arrives at the receiver at $t = 8$.



   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.
      Packet 2: 7 units, Packet 3: 9 units, Packet 4: 8 units,
      Packet 5: 7 units, Packet 6: 9 units, Packet 7: 8 units, Packet 8: 8 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?
      Packet 3, 4, 6, 7, 8

   c) If audio playout begins at $t = 9$, which of the first eight packets sent will not arrive in time for playout?
      Packets 3, 6

   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?
      Minimum playout delay is 2, so playout begins at $t = 10$

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?
      Scheme 1: Every 4 original chunks will have 1 redundant chunk = 25% additional bandwidth; Scheme 2: Every chunk will have its redundant low-quality chunk "piggyback" on the next transmission = 25% additional bandwidth. Hence, both schemes will require additional 25% bandwidth.

   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 will be able to reconstruct the original high-quality audio, while Scheme 2 will get a low-quality audio packet in every 5 packets. Hence, Scheme 1 will have better overall audio quality.

   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 will have better audio quality as Scheme 1 is unable to recover from the packet losses while Scheme 2 can (albeit with low-quality). Note, redundancy-based FEC (e.g., XOR in Scheme 1) has a fixed loss limit up to which we can recover the data. If the losses exceed that limit, then all data lost in that group cannot be recovered.

1. CD quality audio is sampled at 44,100 samples per second. What is the uncompressed data rate of CD quality 2-channel stereo audio ?

(1 mark)
You scored 1 / 1 mark

- 44,100 bps
- 88,200 bps
- 705,600 (= 44,100 x 16) bps
- 1,411,200 (= 44,100 x 16 x 2) bps

2. When video is delivered Over-the-Top (OTT) it means the following:

(1 mark)
You scored 1 / 1 mark

- It means that video delivery is so outstanding, it is "over the top."
- It means that a streaming media service is offered directly to viewers via the Internet. OTT bypasses cable, broadcast, and satellite television platforms, which traditionally distribute video content.
- It means that video is delivered at top bitrates (i.e., very high quality).

5. Which of the following statements is FALSE about the Dynamic Adaptive Streaming over HTTP (DASH) protocol?

(1 mark)
You scored 1 / 1 mark

- DASH works well with the existing web caching infrastructure that ISPs and Content Delivery Networks (CDN) have built up over in recent years.
- With DASH, the client (i.e., video player) measures the available network throughput between the server and the client and then requests an appropriate quality version of the media (e.g., low, medium or high).
- The DASH protocol may encounter difficulties to let the media data pass through network firewalls.
- Much of the video-on-demand media streaming on the Internet today uses either DASH or Apple's HLS.

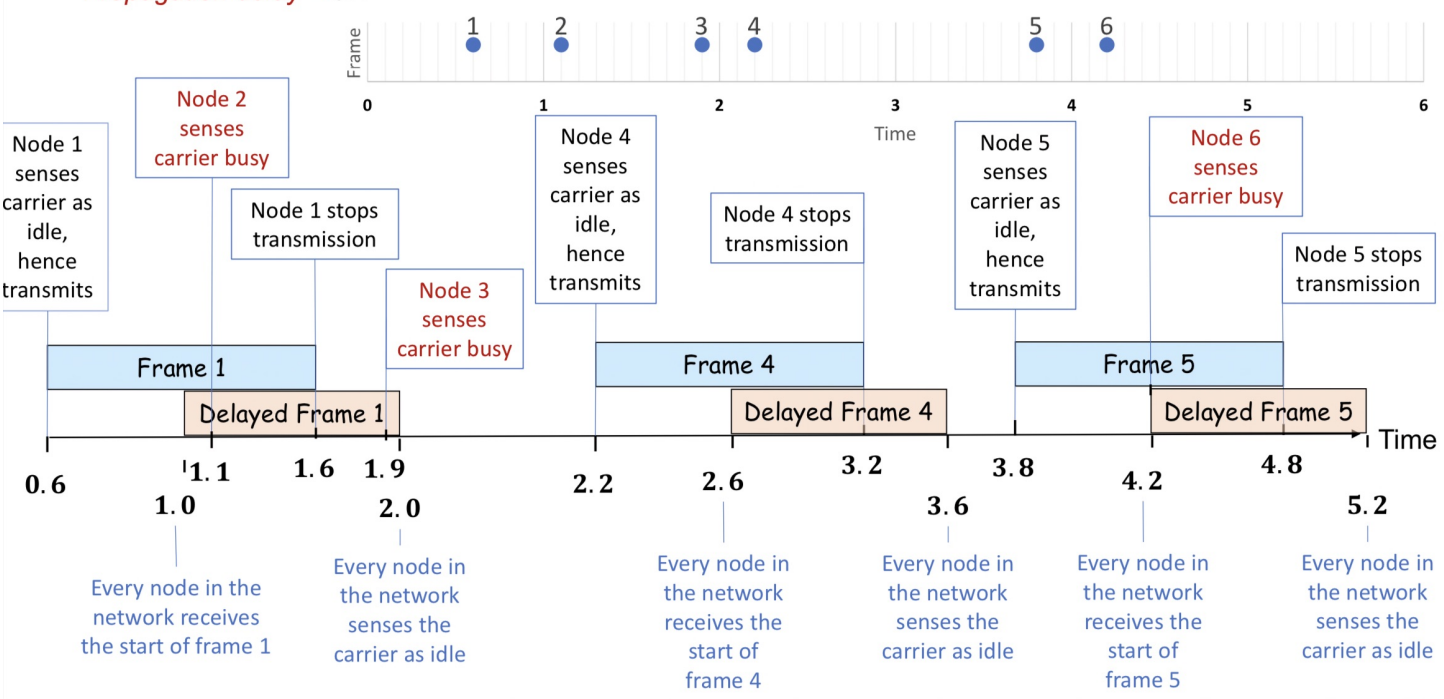
6. In Voice-over-IP (VoIP) applications which of the following statements is FALSE?

(1 mark)
You scored 1 / 1 mark

- Many VoIP applications record, package and send audio data in roughly 20 millisecond chunks.
- In VoIP applications the end-to-end latency (i.e., from microphone to headphones/speakers) should be less than 400 milliseconds, or ideally even less than 150 milliseconds.
- To achieve good voice quality, VoIP systems use a high sampling rate, e.g., 44,100 samples per second.
- VoIP systems may be tolerant to some minor (a few percent) packet loss rates between the sender and the receiver.

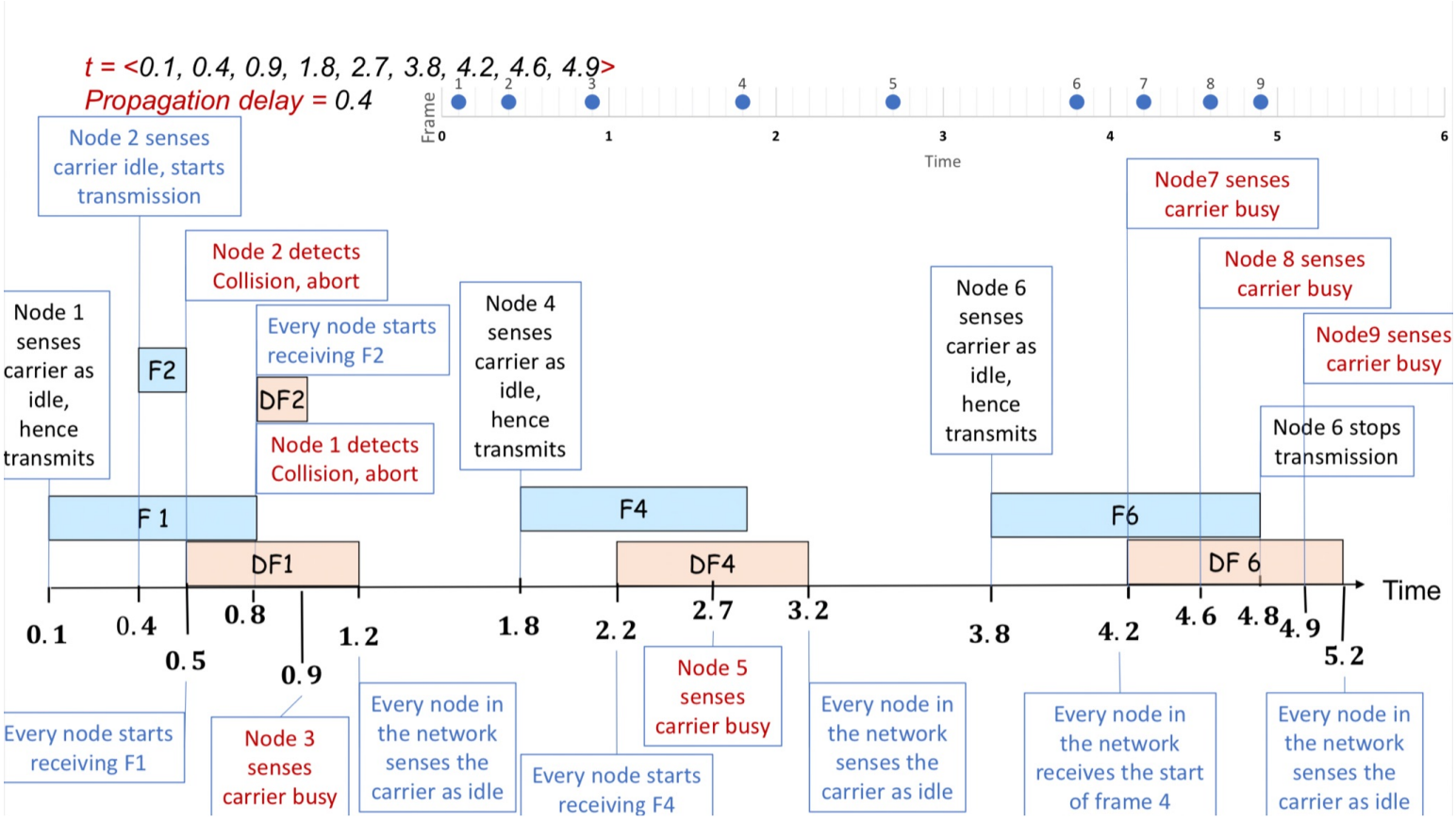
t = <0.6, 1.1, 1.9, 2.2, 3.8, 4.2>

Propagation delay = 0.4

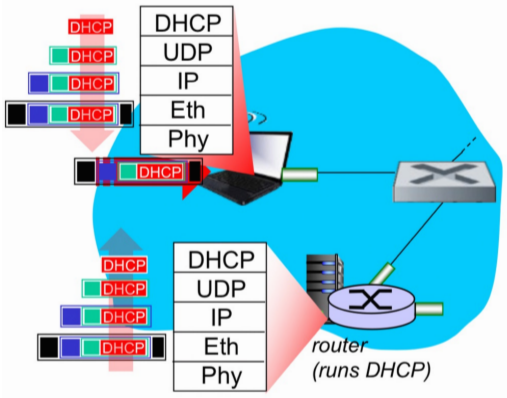


t = <0.1, 0.4, 0.9, 1.8, 2.7, 3.8, 4.2, 4.6, 4.9>

Propagation delay = 0.4



A day in the life… connecting to the Internet

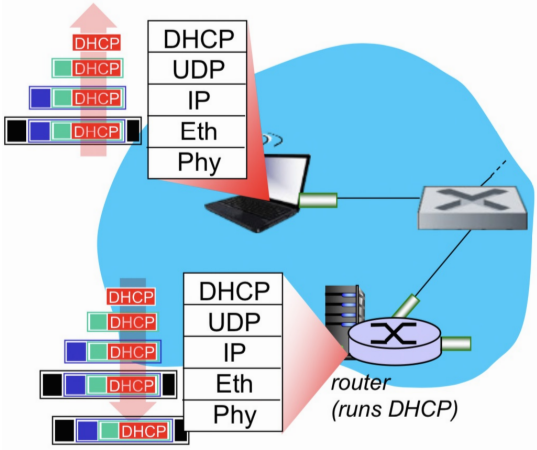


- connecting laptop needs to get its own
  - IP address,
  - addr of first-hop router,
  - addr of DNS server

DHCP

- DHCP request encapsulated in UDP,
  - encapsulated in IP,
    - encapsulated in 802.3 Ethernet
- Ethernet frame broadcast (dest: FF:FF:FF:FF:FF:FF) on LAN, received at router running DHCP server
  - switch learning
- Ethernet demuxed to IP
  - demuxed to UDP
    - demuxed to DHCP

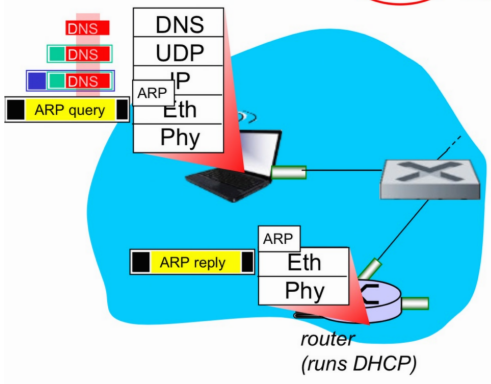
A day in the life… connecting to the Internet



- DHCP server formulates *DHCP ACK* containing
  - client's IP address,
  - IP address of first-hop router for client,
  - name & IP address of DNS server
- encapsulation at DHCP server,
  - frame forwarded (switch learning) through LAN,
- Demultiplexing at client
  - client receives DHCP ACK reply

Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

A day in the life… ARP (before DNS, before HTTP)



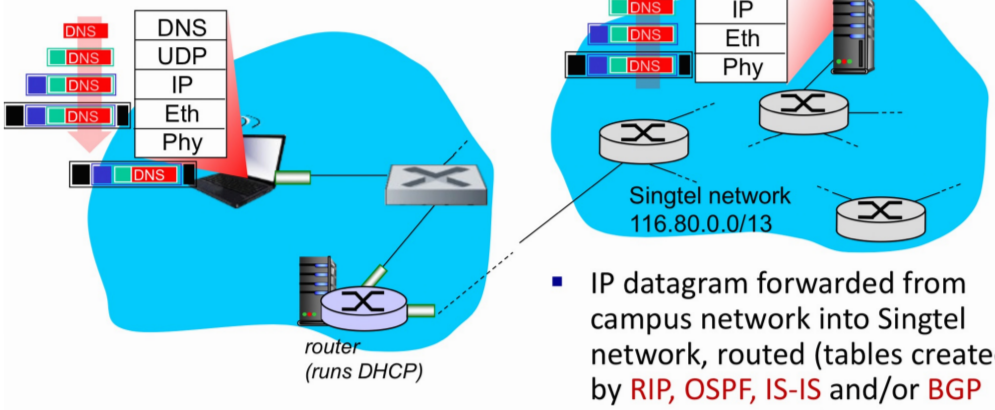
DNS

- before sending *HTTP* request, need IP address of www.google.com
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth.
  - To send frame to router, need MAC address of router interface

ARP

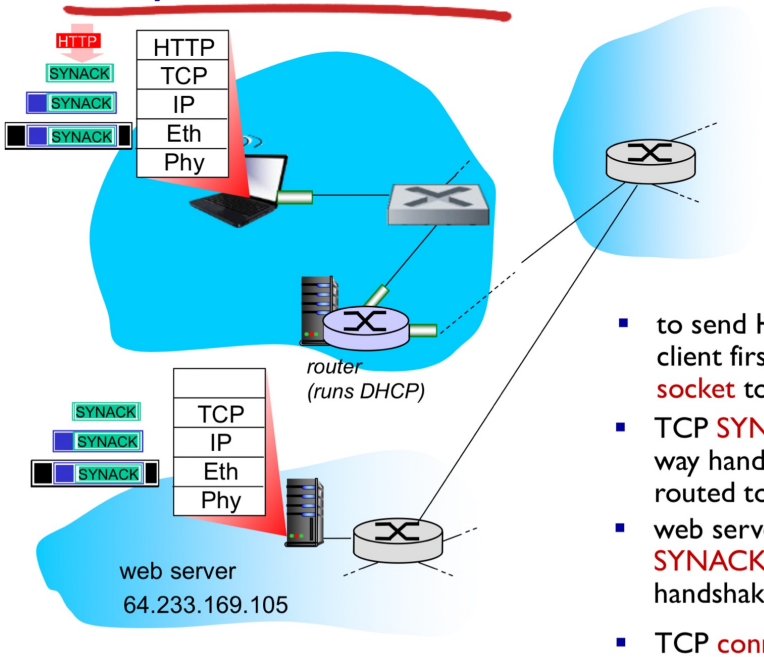
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

A day in the life… using DNS



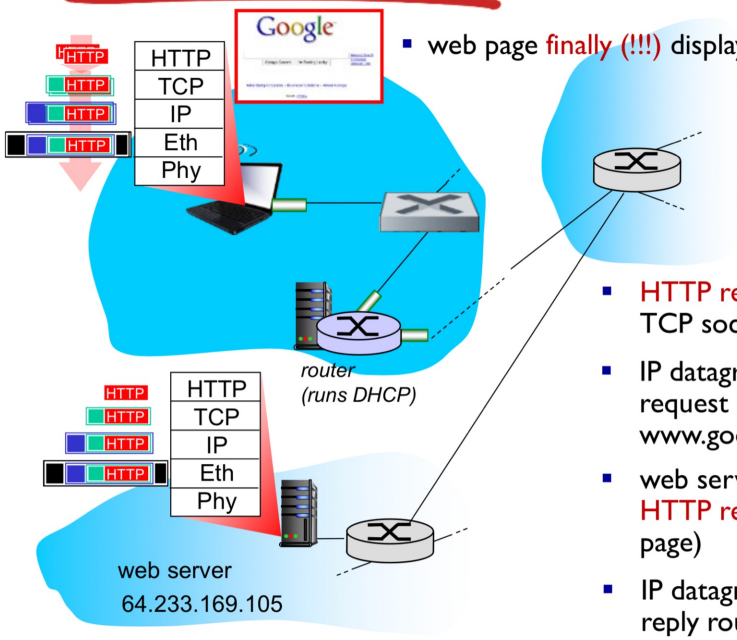
- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
- IP datagram forwarded from campus network into Singtel network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demuxed to DNS server
- DNS server replies to client with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

A day in the life…TCP connection carrying HTTP



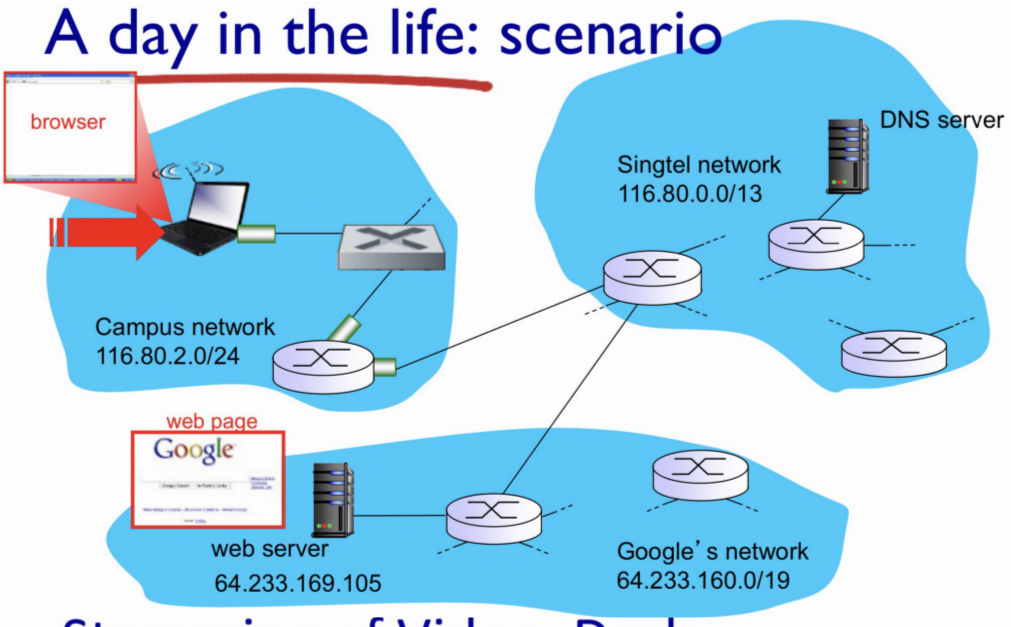
- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in 3-way handshake)
- TCP connection established!

A day in the life… HTTP request/reply

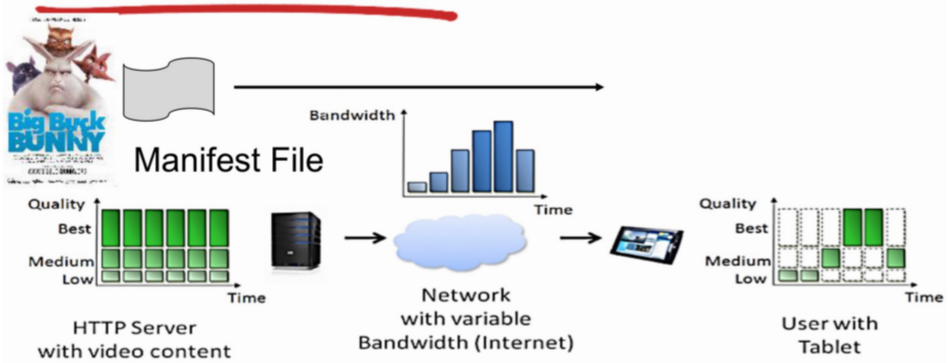


- web page finally (!!!) displayed
- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client

A day in the life: scenario



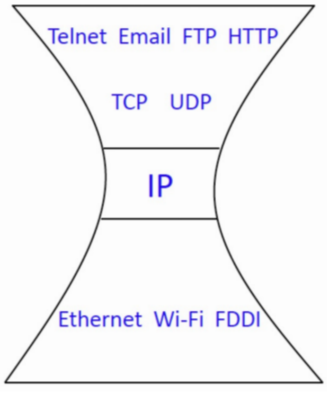
Streaming of Video: Dash



- 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.

Lessons from CS2105

- Network systems are complex!
  - There are many issues to consider, to support different applications running on a large number of hosts through different access technologies and physical media.
- To deal with complexity:
  - Separation of concerns
  - 5 protocol layers
- To deal with scalability:
  - Hierarchical systems



Assorted Qns:

1.10 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.

Q3.

Two hosts *A* and *B* are 2,000 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*.

(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?

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.

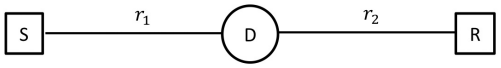
B. To send a packet outside the host's subnet, the host must first discover the IP address of its first-hop router using DHCP.

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.

1.10 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?



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.

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

128 64 32 16 8 4 2 1 x x x x x x x x
1 1 0 1 0 0 0 0

reduce by 1 bit when matching y.

Q4.

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

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

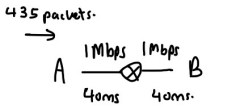
Interface	IP Range	No. of IP
3	1100 0000 – 1100 1111 1110 0000 – 1111 1111	32+16=48
4	1010 0000 – 1011 1111	32
1	1000 0000 – 1001 1111	32
2	1101 0000 – 1101 1111	16
0	0000 0000 – 0111 1111	128

Q5.

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.

packets = $\lceil \frac{400000}{1000-80} \rceil = 435$ ✓

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)?



$d_{Total} = 40 + 40 + \frac{1000}{10^3} + \frac{434800}{10^3}$ (in ms)

d_{prop} - 5 of 6 - d_{trans} A, d_{trans} B.

Q6.

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 integrity using only these 4 keys.

1. Alice encrypts m with her private key to create digital signature $K_A^-(m)$.
2. Alice concatenates message with digital signature $m \oplus K_A^-(m)$, and encrypt the extended message with Bob's public key: $K_B^+(m \oplus K_A^-(m))$.
3. Alice sends $K_B^+(m \oplus K_A^-(m))$ to Bob.
4. Bob decrypts the received message using his private key: $K_B^-(K_B^+(m \oplus K_A^-(m))) = m \oplus K_A^-(m)$.
5. Bob then uses Alice's public key to derive message from digital signature: $K_A^+(K_A^-(m)) = m'$
6. If $m = m'$, message integrity is preserved.
7. Because message is encrypted during transmission, message confidentiality is preserved.

(Another approach is for Alice to send $K_B^+(m) \oplus K_A^-(K_B^+(m))$)