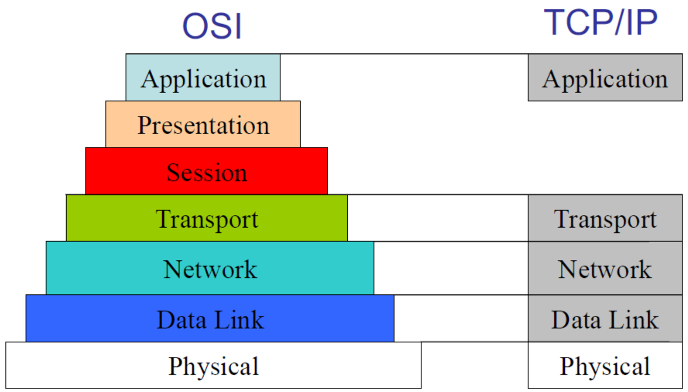
Why need Cross-Layer: 1.the unique problems created by wireless links, the possibility of opportunistic communication on wireless links, and the new modalities of communication offered by the wireless medium. The key is to self-adaptive.

pros and cons of cross layer design

Communication system design is basically a tradeoff of performance versus complexity. Cross layer design leads to performance gains at the cost of complexity. This complexity can be at run-time or at the design stage. Since cross layer design violates the layered architecture, it could also lead to stifling of innovation and difficult maintenance.



The TCP/IP Model

Application Layer – concerned with how data at both ends is handled, user interface
Transport Layer – manages end-to-end flow of data, reliability, congestion control
Network Layer – which performs routing and provides hierarchical addressing
Data Link Layer – manages transmission of data on a link-by-link basis, link-level reliability
Physical Layer – used for transmitting data on the physical medium

OSI Model

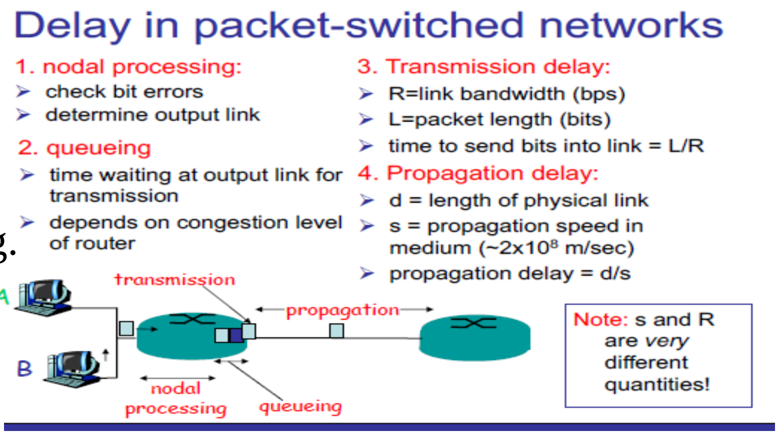
Application Layer – It is concerned with how data at both ends is handled, user interface. The application layer establishes the availability of intended communication partners, synchronizes and establishes agreement on procedures for error recovery and control of data integrity.
Presentation Layer – It converts the data into a format compatible with the receiver's system format and suitable for transmission. Translates between multiple data formats by using a common format. Provides encryption and compression of data. Examples :- JPEG, MPEG, ASCII.
Session Layer – The session layer defines how to start, control and end conversations (called sessions) between applications. It also synchronizes dialogue between two hosts' presentation layers and manages their data exchange. It offers provisions for efficient data transfer. SSL, SQL
Transport Layer – It manages end-to-end flow of data, reliability, congestion control. It contains TCP/UDP. The transport layer segments data from the sending host's system and reassembles the data into a data stream on the receiving host's system. It ensure end-to-end reliability.
Network Layer – Defines end-to-end delivery of packets. Defines logical addressing so that any endpoint can be identified. Defines how routing works and how routes are learned. Routers operate at Layer 3. Find a set of reliable links which form a path from source to destination, routing.
Data Link Layer – It manages transmission of data on a link-by-link basis, link-level reliability. The data link layer provides reliable transit of data across a physical link by using the Media Access Control (MAC) addresses. It create a reliable link.
Physical Layer: The physical layer deals with the physical characteristics of the transmission medium. It learns how to use the medium of communication, and create a link.

Circuit switching VS Packet Switching

Circuit switching: dedicated circuit per call: telephone net – call setup required

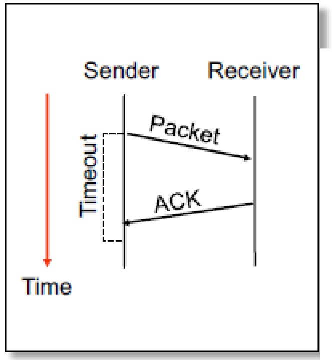
Packet switching: data sent through net in discrete “chrunks”– each end-to-end stream divided into packets - each packet uses full link bandwidth resource contention – aggregate resource demand can exceed amount available. Congestion- packets queue, wait for link use.

store and forward: packets move one hop at a time. Node receives complete packet before forwarding.



ARQ: Automatic Repeat Request

ARQ provides link layer reliability at the hop-by-hop level. TCP is at the transport layer and provides end-to-end reliability. Receiver sends acknowledgment(ACK) when it receives packet. Sender waits for ACK and timeout if it does not arrive within some time period.



Multiplexing and Demultiplexing

UDP

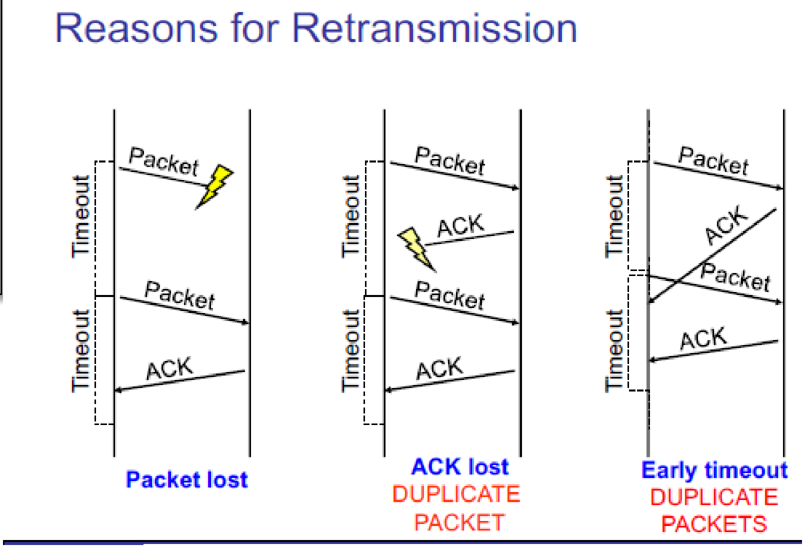
UDP applications: 1. multimedia streaming: telephone calls, video conferencing Gaming. 2. Simple query protocols like Domain Name System.

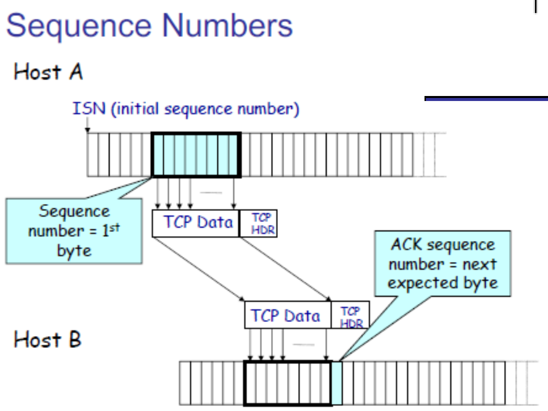
TCP properties:

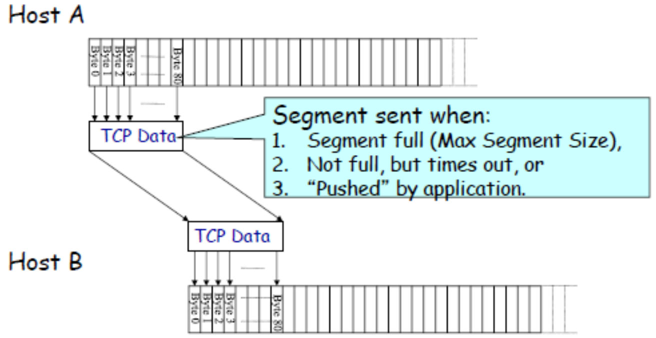
1. connection oriented – explicit set-up and tear down of TCP session
2. stream-of-bytes service – a stream of bytes, not messages
3. Reliable, in-order delivery – checksums to detect corrupted data. Ack & retransmission for reliable delivery. Sequence numbers to detect losses and record data.
4. Flow control – prevent overflow of the receiver’s buffer space
5. Congestion control. – Adapt to network congestion for the greater good.

TCP segment: TCP header is 20 bytes long, IP header is 20 bytes long.

ISN – Intial Sequence Number



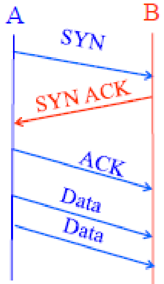




TCP Three-Way Handshake

How to establishing a TCP Connection:

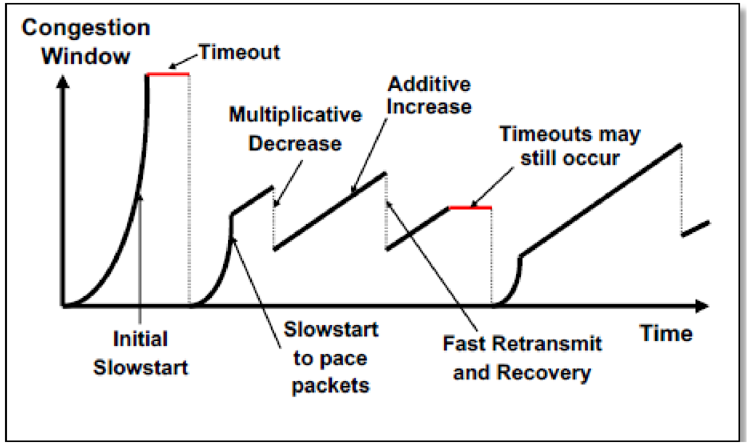
1. Host A sends a SYN (open) to the host B
2. Host B returns a SYN acknowledgment (SYN ACK)
3. Host A sends an ACK to acknowledge the SYN ACK



How to close a TCP connection:

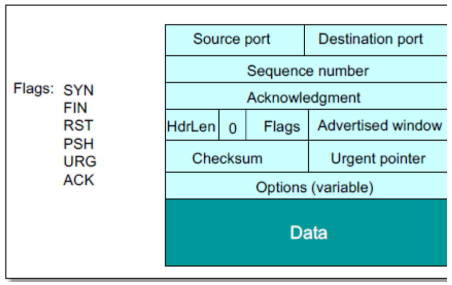
1. Host A send a Finish (FIN) to close and receive remaining bytes
2. Host B sends a FIN ACK to acknowledge
3. Host B sends a Finish(FIN) to close and send remaining bytes
4. Host A receive remaining bytes from Host A and send ACK

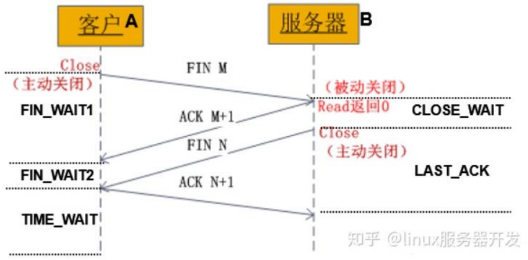
TCP Saw Tooth Behavior (TCP Reno)



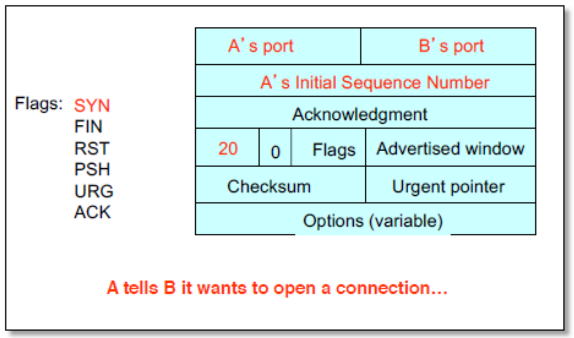
TCP Header

Each host tells its ISN to the other host.

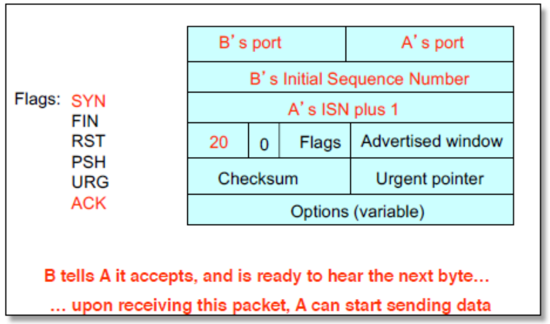




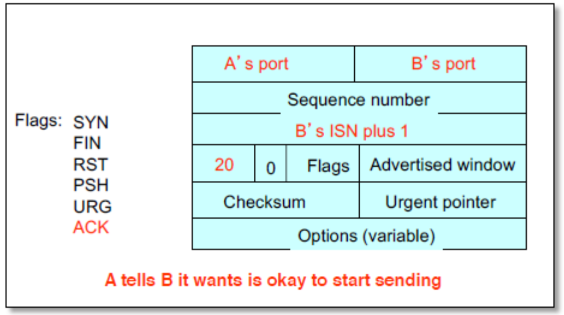
Step 1: A's Initial SYN Packet



Step 2: B's SYN-ACK Packet



Step 3: A's ACK of the SYN-ACK



… upon receiving this packet, B can start sending data

Network-Layer Functions

Forwarding: move packet from router’s input to appropriate router output. Process of planning trip from source to destination.

Routing: determine route taken by packets from source to destination. Process of getting through single interchange

Routing algorithms: RIP, OSPF, BGP.

CIDR: Classless Inter Domain Routing

DHCP: Dynamic Host Configuration Protocol

1. host broadcasts “DHCP discover” msg
2. DHCP server responds with “DHCP offer” msg
3. host requests IP address: “DHCP request” msg
4. DHCP server sends address: “DHCP ack” msg

ICANN: Internet Corporation for Assigned

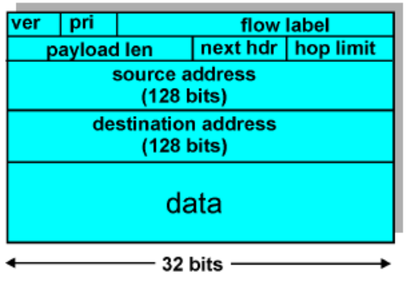
NAT: Network Address Translation

IPv6 Header (Cont)

Priority: identify priority among datagrams in flow

Flow Label: identify datagrams in same “flow.” (concept of “flow” not well defined).

Next header: identify upper layer protocol for data

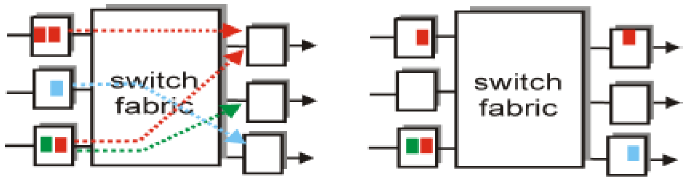


Input Port Queuing

Fabric slower than input ports combined -> queueing may occur at input queues

Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward

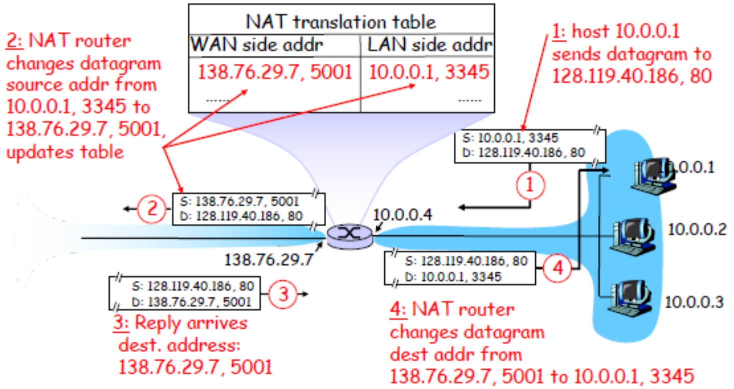
queueing delay and loss due to input buffer overflow!



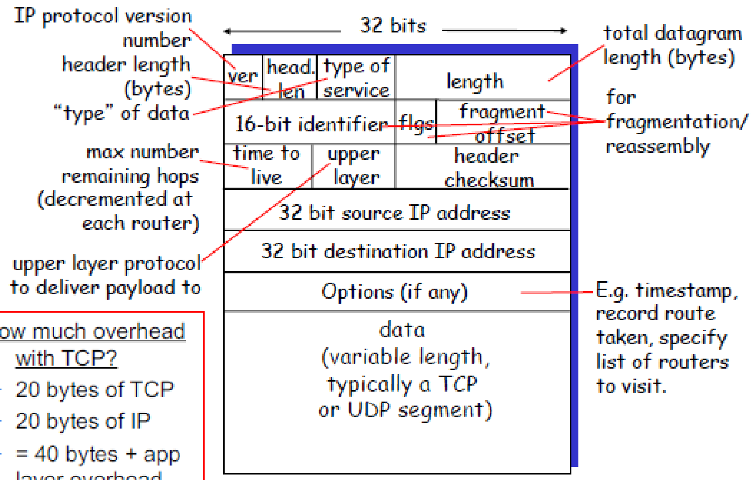
output port contention at time t - only one red packet can be transferred

green packet experiences HOL blocking

NAT: Network Address Translation



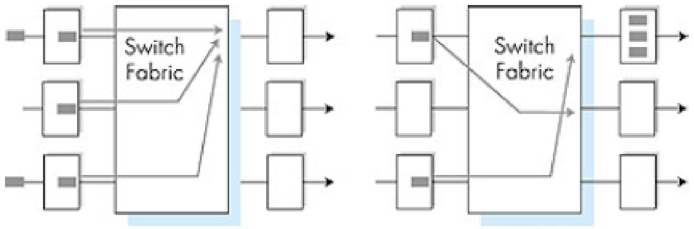
IP datagram format



how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Output Port Queueing



Output Port Contention at Time *t*

One Packet Time Later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*
- Buffer sizing: $B=RTT*C/\sqrt{N}$, C=link capacity, N=# flows
- Packet scheduling: FCFS, WFQ
- Active queue management: RED

A Link-State Routing Algorithm
Dijkstra's algorithm

Distance Vector Algorithm
Bellman-Ford Equation(dynamic programming)

Intra-AS Routing
Also known as Interior Gateway Protocols(IGP)
RIP: Routing Information Protocol (Distance Vector)
OSPF: Open Shortest Path First (Link State)
IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

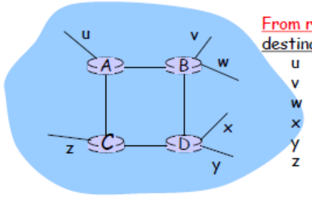
- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - errors propagate thru network

RIP (Routing Information Protocol)

- distance vector algorithm
- included in BSD-UNIX Distribution in 1982
- distance metric: # of hops (max = 15 hops)
- ***distance vectors:*** exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- each advertisement: list of up to 25 destination nets within AS



From router A to subnets:

destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

OSPF (Open Shortest Path First)

- "open": publicly available
- uses Link State algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbor router
- advertisements disseminated to entire AS (via flooding)
  - carried in OSPF messages directly over IP (rather than TCP or UDP

Dijsktra's Algorithm

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
       then D(v) = c(u,v)
    else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
     D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

Classify RIP/OSPF/BGP according to the following metrics: LS or DV, Intra-AS or Inter-AS, Centralized or Distributed.

RIP – DV, Intra-AS, Distributed
OSPF – LS, Intra-AS, Centralized
BGP – DV, Inter-AS, Distributed

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

Internet inter-AS routing: BGP

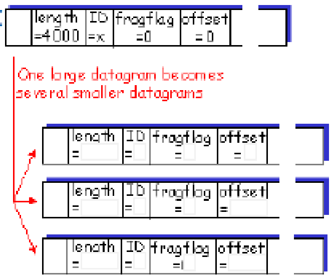
- BGP (Border Gateway Protocol): *the* de facto standard
- BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate reachability information to all AS-internal routers.
  3. Determine "good" routes to subnets based on reachability information and policy.
- allows subnet to advertise its existence to rest of Internet: *"I am here"*

- Consider the CIDR address: 206.68.149.103 / 21
- What is the first address in the range?
  11001110 01000100 10010101 01100111
  206 68 144 0 is the first address
  206 68 144 1 is the first assignable address
- What is the broadcast address?
  11001110 01000100 10010111 11111111
  206 68 151 255
- What is the subnet mask (in binary)?
  11111111 11111111 11111000 00000000
  255 255 248 0

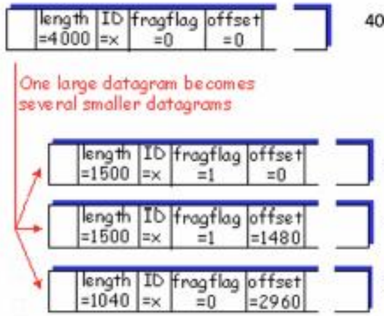
Q17

An IPv4 datagram is fragmented into three smaller datagrams. Which of the following is true?

A: The do not fragment bit is set to 1 for all three datagrams.
B: The identification field is the same for all three datagrams.
C: The more fragment bit is set to 0 for all three datagrams.
D: The offset field is the same for all three datagrams.



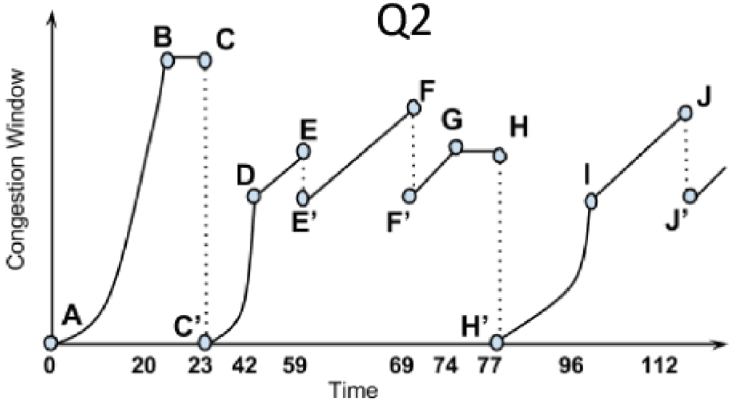
- Suppose an IPv4 datagram is fragmented into three datagrams. The MTU is 1500 bytes. Fill in the information in the diagram below.



4000=20+3980
1500=20+1480
1500=20+1480
1040=20+1020

18

Q2



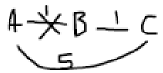
Slow-start: A-B, C'-D, H'-I
Timeout: C, H
Fast-retransmit and Recovery: E, F, J
CWND at A, B, C, C', D, E: 1, 2^20, 2^20, 1, 2^19, 2^19+17
SSThresh at A, B, C, C', D, E: inf, inf, inf, 2^19, 2^19, 2^19

4

Differences between Link State, Distance-Vector and Path-Vector routing protocols

- Which generates more network traffic in a large network? *Link State*
- Which protocol uses the least router memory? Distance Vector
- Which protocol handles link additions and failures?
  Distance Vector
- Which protocol handles routing loops better?
  Path-vector

Path vector



Q25

Q: The fundamental reason for which loops form in the Distance Vector protocol is that a node A decides to use a neighbor B as the next hop for a destination based on routing information that was, at some point, propagated by A itself.

Give an example of this - draw a simple topology, break a link, and show a sequence of updates triggered by the distance vector protocol.

Before break:
At C: d(ca)=2
After break: At B: d(ba)=infinity
At B: d(ca)=2, run BF, d(ba)=2+1=3, inform c
At C: run BF, d(ca)=4, inform b
At B: run BF, d(ba)=5, inform c
At C: run BF, d(ca)=min(5+1,5)=5
At B: run BF, d(ba)=6
Everything stabilizes.

```
A—B—C

If B–C breaks, then B will notice,
recompute its route to C using A,
which will then update its route to C
using B, and so on forever. This is
the count-to-infinity problem in DV
routing.
```

Poisoned reverse partially fixes this: B lies to A.

In Path vector, nodes also exchange path information. This fixes the routing loop problem once and for all.

Q29

W=1, send 1 packet
After 1 RTT → W=2, send 2 packets
After 2 RTT → W=4, send 4 packets
After 3 RTT → W=8, send 3 packets

- Assuming that the available link capacity and the receiver window are infinite how many round-trip times does it take in TCP to send the first 10 packets?
- In general, how many round-trip times does it take to send the first *k* packets?

After 2 RTTs the TCP flow sends 1+2+4=7 packets. After the third RTT, the TCP's cwnd becomes 8. As a result, the TCP will send the remaining 3 packets. Thus TCP needs 3 RTTs to send 10 packets by using slow-start.

1+2^1 + 2^2 + ...2^(m-1) < k <= 1 + 2^1 + 2^2 + ...2^m,
(2^m) - 1 < k <= 2^(m+1) - 1,
m-1 < log2(k+1) -1 <= m,
m = ceiling(log2(k+1)-1) = ceiling(log2(k+1)) - 1
where ceiling(x) denotes the smallest integer that is greater or equal to x.

29

EE4204 Final Examination Cheat-sheet

1. Introduction & Basis

1) *ISO-OSI seven layers architecture:* physical layer, data link layer, IP layer, transport layer, session layer, presentation layer, application layer.
2) *IETF five layers:* (hourglass design) physical layer, data link layer – frame, IP layer – datagram, transport layer – segment, application layer – message.
3) *Layering:* ensure encapsulation and fragmentation, protocols provide service interface and peer-to-peer interface (cross layer design, possible?).
3) *Two kinds of packet switches:* router (IP layer), switch (data link layer).
4) *Network components:* core network (ISP), access network (telephone-based, cable-based, fiber-based, wired, wireless), network edges (hosts + servers).
a. Digital subscriber line (DSL): existing telephone, < 2.5/2.4 Mbps up/down;
b. Hybrid fiber coax (HFC): frequency multiplexing, < 2/30 Mbps up/down;
c. Fiber to the home (FTTH), passive optical network (PON);
d. Wi-Fi 802.11b/g < 11.54 Mbps (local), 3G/4G LTE 1 – 10 Mbps (wide).
5) *Link performance:* bandwidth (Hz), data rate (bps), channel capacity (noise).
6) *In local area networks:* broadcast link, point-to-point link, token ring.
7) *Multiplexing methods:* time division multiplexing (fixed – FTDM, statistical – STDM), frequency division multiplexing.
8) *Switching methods*: circuit switching (fixed TDM), packet switching (store and forward, statistical TDM).
9) *Address translation:* domain name to IP address – DNS (over UDP), IP address to MAC address – ARP (under the same LAN).
10) *Delays:* transmission delay (T_t), propagation delay (T_p), queuing delay (T_q), processing delay, packetization delay, etc.
11) *Transmission speed:* one-way unacknowledged transfer – $T_t + T_p + T_q$, one-way acknowledged transfer – $T_t + 2 \cdot T_p + T_q$.
12) Delay (*D*) and bandwidth (*B*) product = amount of data "in the pipe".
13) Effective throughput: $RTT + message\ size/bandwidth$.

2. Data Link Layer

1) When a packet is transferred around in the network, the source/destination MAC address changes between each two hops, while IP address remains the same (always the initial source or eventual destination address).
2) Link layer ensures channel reliability; transport layer ensures end-to-end reliability.
3) Shannon's capacity theorem: $C = B \cdot \log_2(1 + S/N)$.
4) *Framing approaches:*
a. sentinel-based: delineate with byte 7E, bit staffing in HDLC– insert 0 after five consecutive 1s, byte staffing in PPP – use 7D as escape character;
b. counter-based: count field in header, back-to-back frames could be affected;
c. clock-based: 810 bytes per 125 μs = 51.84 Mbps (STS-n = n * 51.94 Mbps).
5) *Cyclic Redundancy Check (CRC):* represent the message and divisor as polynomial, perform modulo-2 arithmetic (binary addition with no carry).

```
          11010111
      ______________
1101 | 10100110000
       1101
       ----
        1110                 M = 1 0 1 0 0 1 1 0
        1101
        ----                 C = 1 1 0 1
          1111
          1101               T = 1 0 1 0 0 1 1 0 0 0 0
          ----
           1000              R = 0 1 1
           1101
           ----              P = T-R=1 0 1 0 0 1 1 0 0 1 1
            1010
            1101
            ----
             1110
             1101
             ----
              011
```

6) Flow control ensures that the sender does not overwhelm the receiver (stop and wait, sliding window with ACK n or RR n).
7) *Automatic repeat request (ARQ):* introduce NACK, REJ, SREJ.
a. Stop and wait: TIMEOUT mechanism, alternate between ACK0 and ACK1;
b. Go back N: ACK n or RR n, REJ i will trigger sender to go back to i;
c. Selective reject: ACK n or RR n, SREJ i will trigger sender to re-transmit i.
8) *Performance:* let $a = T_p/T_f$ represent the number of frames held in the link.
a. Stop and wait: link utilization $U = (1 - P_f)/(1 + 2a)$;
b. Sliding window (error-free): assume window size is W, $U = W/(1 + 2a)$ if $W < 1 + 2a$ or $U = 1$ if $W \geq 1 + 2a$;
c. Selective reject: $U = (1 - P_f) \cdot W/(1 + 2a)$ if $W < 1 + 2a$ else $U = 1 - P_f$;
d. Go back N: $U = \frac{(1-P_f) \cdot W}{(1-P_f+P_f \cdot W)(1+2a)}$ if $W < 1 + 2a$ else $U = \frac{1-P_f}{1+2a \cdot P_f}$.
9) *Ethernet:* max 2500m by 5 segments (separated by 4 repeaters).
a. Collision detection: carrier sense multiple access (CSMA), use exponential back-off algorithm (randomly wait [0, 2^n-1] slots at n^{th} collision, give up after);
b. Minimum frame size: 64 bytes (512 bits for 10 Mbps link = 51.2 μs RTT);
c. LAN connection: bus (single collision domain), hub (copy frames to all other ports) and switch (store and forward, port to port);

d. LAN extension: bridge (source routing, transparent, spanning tree);
e. Forward table & backward learning: dynamic record down source port;
f. Distributed spanning tree bridge: to avoid loop (assign each bridge a unique ID, use the bridge with smallest ID as root, initially claim itself as root, stop forwarding when a neighbor is nearer to the actual root).
10) *Wireless network:* Bluetooth, Wi-Fi and 3G/4G LTE.
a. Spread spectrum technique: frequency hopping (transmit over a sequence of frequencies, from a pseudo-random generator with pre-agreed seed);
b. Direct sequence technique: n-bit chipping code (XOR with n random bits);
c. 802.11 does not have collision detection (due to hidden & exposed node problem), but has collision avoidance (request to send, clear to send);
d. Scanning (active – Probe, Probe Response, Association Request, Association Response, passive – Beacon, Association Request, and Association Response).

3. IP (network) Layer

1) *Two key functionalities:* forwarding (longest prefix matching), routing.
2) *Datagram network* – "smart" end systems, *virtual circuit (VC) network* – "dumb" end systems, complexity inside network.
3) *Router:* run routing algorithm, forward datagrams from in-port to out-port.
a. Switching fabrics: memory, bus, crossbar (interconnection network);
b. Input port: decapsulation, decentralized switching, queuing (HOL blocking);
c. Output port: buffering (queuing), scheduling discipline;
d. Queuing (delay) and loss leads to input/output buffer overflow.
4) By class-less interdomain routing (CIDR), each isolated network is a subnet.
5) Dynamic Host Configuration Protocol (DHCP) dynamically allocates IP addresses (DHCP discover, DHCP offer, DHCP request, DHCP ack).
6) *Network Address Translation (NAT):* replace all internal IP addresses with one single IP address differentiated by ports. Although NAT solves the address shortage problem, the optimal solution should be IPv6 instead.
7) *NAT traversal problem:* static configuration, Universal Plug and Play (UPnP), relaying (used in Skype).
8) *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers.
9) *Link state routing algorithm:* Dijkstra's algorithm, global algorithm.
a. May not be able to produce correct answer for negative weights;
b. Cannot work when there is negative cycle (since answer is -∞).
10) *Distance vector routing algorithm:* Bellman-Ford algorithm, decentralized.
a. Bellman-Ford equation: $d_x(y) = min_v\{c(x,v) + d_v(y)\}$;
b. Each node waits for any change, recompute the estimates and broadcasts;
c. Could result in "count to infinity" problem if links breaks;
d. Poisoned reverse: Z tells Y $d_z(x) = \infty$ if Z routes to X via Y;
e. BGP-4 solves the "count to infinity" problem ultimately by using AS_PATH attribute (to list the full path and thus it does not include the current AS).
11) We need to aggregate routers into autonomous systems (AS), thus require intra-AS routing protocol and inter-AS routing protocol.
a. Inter-AS and intra-AS routing reflects the hierarchical network structure;
b. Inter-AS protocol propagates reachability information to all internal routers.
12) *Interior Gateway Protocol (IGP)* in the Internet, intra-AS protocols:
a. Routing information protocol (RIP): based on distance vector with poison reverse (infinite distance = 16 hops);
b. Open shortest path first (OSPF): based on link state, flooding via IP;
c. Interior gateway routing protocol (IGRP): Cisco proprietary.
13) *Border Gateway Protocol (BGP)* in the Internet, inter-AS protocol: based on distance vector, exchange routing information over BGP sessions (via TCP).
14) *Broadcast routing:* use in-network duplicate along a spanning tree.
15) *Multicast routing:* use Steiner Tree as the minimum cost tree to connect all routers with attached group members.

4. Transport Layer

1) Most services use TCP, but some (like DHCP, DNS and traceroute) use UDP due to no setup required.
2) *TCP reliable delivery:* checksum, sequence number, re-transmission.
a. Three-way handshake: SYN, SYN ACK, ACK;
b. Tearing down connection: (FIN, FIN ACK) * 2, RST;
c. Stop and wait: keep timeout length as a function of (estimated) RTT;
d. Sliding window: receiver advertises the window size to sender;
e. Fast re-transmission: re-transmit data after receiving 3 duplicate ACKs;
f. Congestion control: actual window size is min of congestion window and flow window, slow start & additive increase & multiplicative decrease;
g. Facing 3 duplicate ACKs, Reno cuts CW by half, Tahoe treats as timeout;
h. Congestion avoidance: implicit – random early dropping (RED), explicit – intermediate router sets the DEC bit in packet header.
3) *TCP throughput:* controls the amount of traffic by adjusting window size.
a. Instantaneous send rate: W/RTT;
b. Instantaneous receive rate: $\leq$ send rate;
c. Average send rate under AIMD: $((W + 0.5W)/2)/RTT = 0.75 \cdot W/RTT$.
4) *Rethinking end-to-end (e2e):* (approximated) flow recognition is the key.