Store and forward: entire packet must arrive at router before it can be transmitted on next link

ISP: Internet Service Providers | IXP: Internet Exchange Point

Four sources of packet delay
- d-proc: nodal processing delay, checking bit errors and deciding the output link
- d-queue: queueing delay, waiting to transmit at output link
- d-trans: transmission delay, time to transmit on the output link
- d-prop: propagation delay, time to propagate through a physical link

$$RTT = 2 \cdot d_{prop}$$

Throughput: rate at which bits transferred between sender and receiver

Bottelneck link: link on end-end path that constrants end-end throughput

The reason for layering:
- Decrease the complexity: break a complex task into smaller ones
- Modularization eases maintenance and updating: Change of the layer's service transparent to rest of the system

TCP/IP layer model:
Application — Layer 7 (packet)
Transport — Layer 4 (segment)
Network — Layer 3 (datagram)
Link — Layer 2 (frame)
Physical — Layer 1 (bit)

Encapsulation: At each lower layer a new header is added incorporating the headers underneath

Example:
b) Give the name (not just number) of the layers of the network stack described below:

i. This layer exists at end hosts and takes care of reliability (if any) by ensuring lost data is retransmitted and all data arrives reliably in order.
ii. This layer would be used by programmers to open and close sockets and to send data from one machine to another.
iii. This layer contains the routing protocols OSPF and RIP.
iv. This layer would move data from somewhere in a subnetwork to another place in the same subnetwork.
v. This layer would move data across the network from the initial source machine to the final destination machine.

Answer:
(1) Transport layer — Layer 4
(2) Transport layer — Layer 4
(3) Network layer — Layer 3
(4) Network layer — Layer 3
(5) Network layer — Layer 3

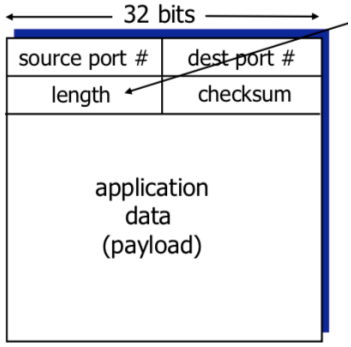
Transport Layer
Transport Layer: provide logical communication between app processes running on different hosts

Multiplexing and demultiplexing
multiplexing: combing several streams of data into a single stream
demultiplexing: a stream of data is seperated into different streams

Demultiplexing (UDP)
2-tuple (Dest IP address, Dest port number)

Demultiplexing (TCP)
4-tuple (Source IP address, Dest IP address, Source port #, Dest port #)

UDP segment header:



UDP segment format

UDP checksum: detect errors in transmitted segment. (e.g. flipped bits)
Obtain the checksum value by adding up the segment contents from sender, put this checksum value in the segment header, after transmitting to the receiver, the receiver adding up the segment content, if it is not equal to the checksum, error is detected.

Source port: 16-bit
UDP header: 64-bit = 8 bytes

Advantage of UDP over TCP:
- Simple to implement
- Connectionless (No need for handshaking)
- Efficient: Small in header size
- In some cases when you do not need retransmission, and want to transmit as fast as possible, UDP is better

Advantage of TCP over UDP:
Reliable, can deal with packet loss and corruption

RDT (Reliable data transfer):
Rdt 1.0: Assume no packet loss and no packet corruption

Rdt 2.0: Assume packet corruption (bit error)
checksum: detect bit errors
ACK: the packet has no error
NAK: the packet has error

Flaw of rdt 2.0
ACK and NAK may be corrupted —> sender retransmits current pkt which may cause duplicate packets.
When sender send a correct segment, the receiver received and send back an ACK, however the ACK corrupted and changes to NAK. The sender received this NAK and retransmit the segment, which cause duplicate packets

Stop and wait: sender sends one packets, then waits for receiver response

Rdt 2.1: Assume packet corruption and ACK/NAK corruption
Adding sequence number (0,1) —> stop and wait

When sender send a correct packet 0, the receiver received and send back an ACK, move to "Wait for ACK/NAK 1", corrupt change to NAK, the sender resend packet 0, however receiver is at "Wait for ACK/NAK 1", drop the packet, stay at the same state, send back an ACK, not corrupt, sender receive an ACK, go into state "Wait for call 1", send packet 1. —> circulate

Rdt 2.2: Assume packet corruption and ACK/NAK corruption
Replacing NAK with ACK for last pkt received
1. packet corrupted (loss bit)
When sender send a packet0, receiver receive, packet corrupted, send back ACK1 (NAK), still "wait for packet0", sender receive ACK1, retransmit, still "wait for ACK0", receiver receive packet0, correct, send back ACK0, move to"wait for packet 1", sender receive ACK0, move to "wait for call 1", send packet1.
2. ACK0 corrupted
When sender send correct packet0, receiver receive, send back ACK0, corrupted to ACK1, move to "wait for packet1", sender receive ACK1, retransmit, still "wait for ACK0", receiver receive packet0, drop, send back ACK0, sender receive ACK0, move to "wait for call from 1", send packet1.

Rdt 3.0: packet corruption, channel with errors and loss (packet loss)
requires countdown timer: retransmits if no ACK received in this time
Solution for packet corrupted: Uses ACK with the "wrong number"
Solution for packet lost: count down the time and when the timout occurs

Sequence number: measures which packet number we are currently sending
ACK-numbered: a specific packet has definitely been arrived
Timeout: the time we wait before we think a packets is lost

Utilisation: $\frac{\frac{L}{R}}{RTT+\frac{L}{R}}$, stop and wait

U-sender = $\frac{m\cdot\frac{L}{R}}{RTT+\frac{L}{R}}$, pipelining

utilisation: the proportion of time that the link is used for sending data

pipelining: allows multiple, "in-flight", yet-to-be-acknowledged packets

Two pipe-lined protocols: GBN (go-back-N), SR (selective repeat)

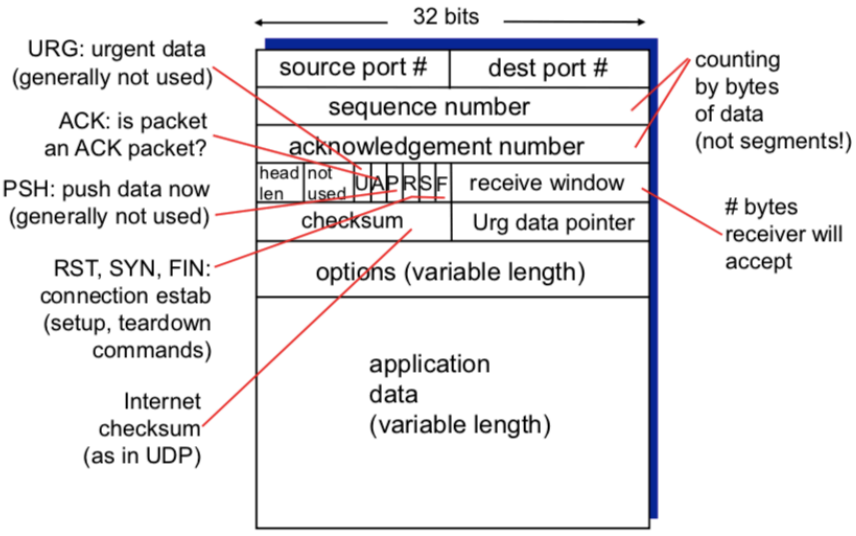
Difference between GBN and SR
Sender side:
1. GBN has timer for the oldest unACKed packets, SR has time for each unACKed packet
2. GBN resends all the packets with higher sequence number in the windows, SR resends only the packets that timed out

Receiver side:
1. GBN will only send acumulative ACK, if there is a gap, ACK won't be sent. SR will send ACK for all correctly received packets
2. GBN won't buffer packets, SR will buffer packets as needed, if it is out of order

Advantage of GBN and SR
GBN:
- Simple to implement
- Only a single timer
- One ACK can be acknowledge many packets

SR:
- Minimise resending of packets
- Individual ACKs give more information to sender

TCP header



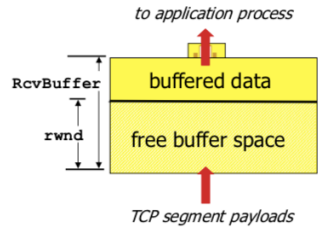
Sequence number: byte stream "number" of the first byte in segment's data
ACK: sequence number of next byte expected from other side (cumulative ACK)
(sender1 send seq#:100, sender2 send ACK#:101)

Set timeout value: longer than RTT
Too short, early timeout, unecessary retransmission
Too long,, slow reaction to packet loss

EstimatedRTT = $(1-a)\cdot EstimatedRTT + a\cdot SampleRTT$
DevRTT = $(1-b)\cdot DevRTT + b\cdot |SampleRTT - EstimatedRTT|$
Time interval: EstimatedRTT plus "safety margin"
Timeout interval = $EstimatedRTT + 4\cdot DevRTT$ —> to determine the timeout

TCP fast retransmits:
motivation: Time-out period is often long
If sender receives 3 ACKs for same data, resend unacked segment with smaller sequence number

Flow control: receiver control sender, so sender will ot overflow free buffer space by transmitting too much, too fast

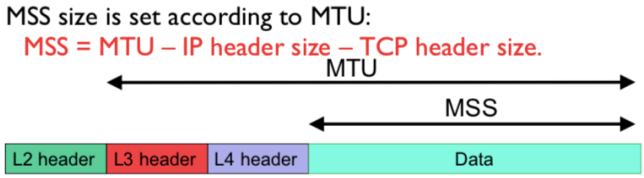


receiver-side buffering

rwnd = receive window

rwnd = receive window (free buffer space), how much buffer space receiver has
rwnd: control the sender, limits the amount of unACKed data to the rwnd value

MSS: largest amount of data in a single datagram
MTU: largest amount of data that a system can pass onwards



Nagle's Algorithm
解决问题：repeatedly emit data in small packets (1 byte data, 40 bytes header)
Ideal: Combining small number of outdoing messages and sending them all at once
终止条件：1. When receiver sends an ACK | 2. Fill a MSS

Silly Window Syndrome

Forced to send very small packets (两种情况):

1. Sender produces data very slowly:
2. Receiver processes data very slowly: (very small rwnd) — every time the receiver only process small number of packets and informed sender the small rwnd, sender sends small number of packets as well

Cure for 1: Nagle's algorithm

Cure for 2: receiver does not advertise rwnd that cause sender to send small packet

TCP 3-way handshaking: SYN, SYN-ACK, ACK

Client state: Listen —> SYN SENT —> ESTAB

Server state: Listen —> SYN RCVD —> ESTAB

TCP closing a connection: FIN, FIN-ACK, ACK

ESTAB —> FIN_WAIT_1 —> FIN_WAIT_2 —> TIMED_WAIT —> CLOSED

ESTAB —> CLOSE_WAIT —> LAST_ACK —> CLOSED

- Timed wait for 2*max segment lifetime

Why FIN-ACK be split into two packets: The closing server may have data left to send

Congestion control: to control or prevent the network from being overloaded and getting too much traffic caused by sources sending data too much or too fast

Congestion window (cwnd): limits the amount of unACKed data send into network (dynamic)

TCP congestion control approach:

Sender increases transmission rate (window size), probing for a usable bandwidth until loss occurs

AIMD:

1. Set cwnd — congestion window to initial value
2. Additive increase: increase cwnd by 1MSS every RTT until loss detected
3. Multiplicative decrease: cut cwnd in half when loss occur

Sender limit transmission

$$LastByteSent - LastByteAcked \leq \min\{rnwd, cwnd\}$$

$$\text{rate} \approx \frac{\text{cwnd}}{\text{RTT}} \text{ bytes/sec}$$

Send cwnd bytes, wait RTT for ACKs, then send more bytes

TCP Slow Start (initial rate is slow)
purpose: when new TCP connection is built, sending large packet will lead to congestion. Slow start prevents a network from becoming congested by starting with small cwnd, and than increment exponentially.
When connection begins, increase rate exponentially until first loss event (ssh):
Intially set cwnd = 1 MSS, Double cwnd every RTT

TCP RENO: initial cwnd = 1 MSS, double cwnd every RTT until reach a threshold, then increment linearly, when loss occur, cut cwnd to half, then grows linearyly

TCP Tahoe: initial cwnd = 1 MSS, double cwnd every RTT until reach a threshold, then increment linearly, when loss ovvur, set cwnd to 1, then repeat

TCP average thourhgput: $\frac{3}{4} \cdot \frac{W}{RTT}$, average windows size: $\frac{3}{4}W$

TCP fairness: if K TCP sessions share same bottleneck link of bandwidth R, each should have average rate of R/K

Network layer
Network layer: to move packets the whole journey from the source to destination to provide internetworking

fowarding: move packet's from router's input to appropriate router's output
routing: determine the routes taken by packet from the source to destination

Data plane
- local, pre-router
- Move datagram from router's input to certain router's output according to the forwarding table
- Forwarding function

Control plane
- Network-wide logic
- Determine the routes taken by packets from source to destination according to the routing algorithm
- Two control-plane approaches:
  - Traditional routing algorithm — implemented in routers
  - Software-defined networking (SDN) — implemented in remote controller

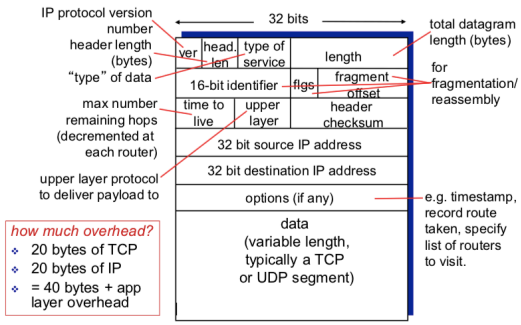
Lookup, fowarding (both use header field values)
Destination-based forwarding: destination IP address
Generalized fowarding: any set of header field values

Longest prefix matching: looking for forwarding table entry for given destination address, using longest prefix to find the matching address
1. forwarding table entry转化成二进制，标出netid
2. 将destination address转化成二进制，与table匹配，判断出口

Three types of switching fabric: memory | bus | crossbar

Head-of-Line blocking: According to ouput port contention, only one of the datagrams with the same destination can be sent, lower on will be blocked. This queued datagram at the front of queue will block others in queue from sending.

IP datagram format



IP address: 32-bit

IP fragmentation, reassembly: "reassembled" only at final destination
MTU include 20 bytes for IP header, minus the 20 bytes we will get the bytes in data field

Example: A datagram has 3000 bytes. It is sent over a network with MTU 1500 bytes. How many fragments must it be split into?
Answer:
The datagram has 2980 bytes data
Each MTU has 1480 bytes for data, the datagram should be divided into 3 segment

Length	ID	Fragflag	Offset
1500	x	1	0
1500	x	1	185
40	x	0	370

Example: A datagram has 2600 bytes. It is sent over a network with MTU 1300 bytes. How many fragments must it be split into?
Answer:
The datgram include 2580 bytes data. Each MTU inclue 1280 bytes data, 1280*2 = 2560, 2560+20 = 2580.

Length	ID	Fragflag	Offset
1300	x	1	0
1300	x	1	160
40	x	0	320

IPv4 addressing
Classful IP addressing: A, B, C, D, E

subnet: device interfaces with same subnet part of IP address

按子网个数分：延长netid，$n_{sub} = n + \log_2 s$
Subnetting reasons:
A large block of IP addresses —> smaller blocks of addresses (individual networks)

Classless addressing
Network address: the address with all hosts bits set to zero
Broadcast address: the address with al hosts bits set to one

VLSM (variable length subnet mask) — only support the largest subnet
要点：从最大的开始，先确定hostid —> subnet mask —> network address + broadcast address —> broadcast address 进一位变成下个subnet的network address —> 继续

Example:

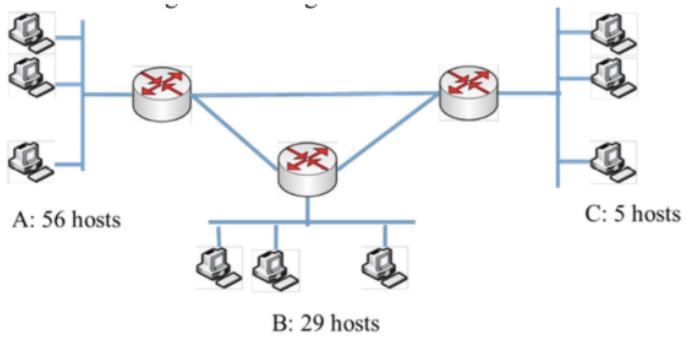


Figure 1

You have the network 144.132.26.0/24. Efficiently assign addresses to the subnets A, B, C in that order as early as you can in the address space (assuming the networks will not get any more hosts).

Answer:
First, for 56 hosts, assign 6 hostid bits, /26
Second, for 29 hosts, assign 5 hostid bits, /27
Third, for 5 hosts, assign 3 hostid bits, /29
Forth, fifth, sixth, assign 2 hostid bits for each, /30

Subnet #1: 144.132.26.0/26 — 10010000.10000100.00011010.00|000000/26
Broadcast address: 10010000.10000100.00011010.00|111111/26,
Network address: 144.132.26.0/26, Broadcast address: 144.132.26.63/26

Subnet #2: 144.132.26.64/27 — 10010000.10000100.00011010.010|00000/27
Broadcast address: 10010000.10000100.00011010.010|11111/27,
Network address: 144.132.26.64/27, Broadcast address: 144.132.26.95/27

Subnet #3: 144.132.26.96/29 — 10010000.10000100.00011010.01100|000/29
Broadcast address: 10010000.10000100.00011010.01100|111/29,
Network address: 144.132.26.96/29, Broadcast address: 144.132.26.103/29

Subnet #4: 144.132.26.104/30 — 10010000.10000100.00011010.011010|00/30
Broadcast address: 10010000.10000100.00011010.011010|11/30,
Network address: 144.132.26.104/30, Broadcast address: 144.132.26.107/30

Subnet #5: 144.132.26.108/30 — 10010000.10000100.00011010.011011|00/30
Broadcast address: 10010000.10000100.00011010.011011|11/30,

Network address: 144.132.26.108/30, Broadcast address: 144.132.26.111/30

Subnet #6: 144.132.26.112/30 — 10010000.10000100.00011010.011100|00/30
Broadcast address: 10010000.10000100.00011010.011100|11/30,
Network address: 144.132.26.112/30, Broadcast address: 144.132.26.115/30

DHCP (dynamic host configuration protocol)
function: dynamically obtain its IP address from network server when it joins network
DHCP discover —> DHCP offer —> DHCP request —> DHCP ACK (broadcast all the way), broadcast: client还没有得到IP address, 所以只能broadcast

DHCP can get: IP address, first host router IP address, name and IP address of DNS server

Example: A network given an Ethernet (MAC) address when it is manufactured. Why not also give it an IP address?
Answer: The host that always has the same IP address can only connect to one network

IPv4 problem facing: running out address, allocated unfairly

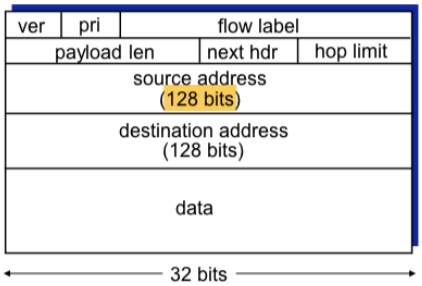
NAT: Network address translation
All datagrams leaving local network have same single source NAT IP address

Implementation (NAT router):
outgoing: (source IP, source port #) —> (NAT IP, new port #)
remember: NAT translation table, store the pair
incoming: search the NAT table, replace (NAT IP, new port #) in destination field —> (source IP, port #)

NAT advantage:
- Just on IP address for all devices
- Change inside don't change outside, Change outside without change inside

NAT disadvantage:
- The address is still running out (address shortage not solved)
- The complexity is in the middle, should be at "ends" (end-to-end)

IPv6 datagram format (fixed-length 40 byte header)



flow label: identify datagram in the same flow
IP address: 128-bit

IPv4 datagram format and IPv4 datagram format
IPv4:
- fragmentation reassembly: 16-bit identifier, flgs, fragmentation offset
- checksum
- options
- header length

IPv6: flow label
both: version, priority, length = payload len, upper layer = next hdr, TTL = hop limit

Example: List some differences between IPv4 and IPv6
Answer:
IPv4 provide 32-bit IP address, IPv6 provide 128-bit IP address
IPv6 do not provide header fields such as checksum and fragmentation in order to speed processing and forwarding

Transition from IPv4 to IPv6: tunneling
Tunneling: IPv6 datagram carried as a payload in the IPv4 datagram when entering into the IPv4 routers

SDN
Data plane:
- Data plane is generalised forwarding using a flow table computed and distributed by a centralised routing controller
- A set of "match-action" rules send by a controller can do many things to packet
- Much more flexible: can use any part of the header field for forwarding

Control plane:
- A single centralised control point | logically centralised routing controller
- programmable, not fixed

Example: Concerning the routing method, what are the advantage using SDN compared with traditional routing methods
Answer:
Traditional routing method:
1. Data plane use longest prefix match forwarding by looking at the forwarding table
2. The routing algorithm is calculated and implemented in each router
3. Only use IP address for forwarding

SDN routing method:
1. Data plane use generalised forwarding by looking at the flow table
2. The routing algorithm is calculated in a remote controller
3. The routing algorithm is programmable
4. Can use any part in header for forwarding

OpenFlow data plane abstraction (specfic SDN protocol)
Generalised forwarding: simple packet-handling rules
- Pattern: match values in packet header fields
- Action: for matched packet, it will drop, forward, … (actions)

match+action: different kinds of devices become one
router | switch | NAT | firewall

global or decentralized routing
global
- All routers have complete topology, link cost information
- "link state" algroithm — Dijsktra's algorithm

decentralized
- Router knows physically-connected neighbors, link cost information to neighbors
- Exchange of information with neighbors
- "distant vector" algorithm

U'	D(A),P(A)	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)
S	4,S	7,S	-	-	-
SA		6,A	8,A	-	-
SAB			7,B	10,B	-
SABC				9,C	11,C
SABCD					10,D
SABCDE					

Distant vector algorithm
Bellman-Ford equation: $dx(y) = \min\{c(x,v) + dv(y)\}$
when to apply Bellman-Ford: 1. Msg from neighbor, 2. Link-cost change

Link cost changes (two situations):
1. "good news travels fast"
2. "bad news travels slow" — count to infinity, <— solution: poisoned reverse

poisoned reverse: for x, y, z, y tells z its link cost to x is infinity, so that z won't route to x via y

AS: autonomous systems

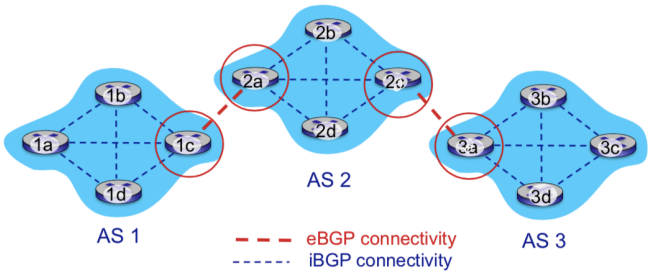
Intra-AS routing: IGP (Interior gateway protocol): OSPF (open shortest path first)
Inter-AS routing: BGP (Border gateway protocol)
gateway router

Interconnected ASes
Forwarding table — both intra- and inter-AS routing algorithm
intra-AS —> entry for destinations within AS
inter-AS and intra-AS —> entry for destination for external destinations

Intra-AS routing : OSPF
- Open
- Link state algorithm (Dijkstra)
- Allow multiple same-cost paths
- Security

Inter-AS routing (BGP)
iBGP: obtain reachability information from neighboring ASes
eBGP: propagate reachability information to all the routers in the same AS
- gateway routers run both eBGP and iBGP protocols

OSPF: link-state algorithm | BGP: distant vector protocol (TCP connection: exchange information)



AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:
AS2 router 2c receives the advertisement AS3,X (eBGP) from AS3 router 2a
Based on AS2 policy, AS2 router propagate the advertisment AS3,X to all the routers in AS2 (iBGP)
Based on AS2 policy, AS2 router 2a send advertisement AS2,AS3,X to router 1c in AS1 (eBGP)

BGP attributes — AS-PATH + NEXT-HOP

SDN — control plane
Question: Logically centralised control plane
- Easier network management: avoid misconfiguration, greater flexibility of traffic flow
- Routing algorithm is programmable:
  - Centralised controller is easier: compute tables centrally
  - Distributed programming is difficult: compute tables and implementd per-router
- Open implementation

ICMP (Internet Control Message Protocol)

Relationship between IP and ICMP:
Two protocols work together, IP: not reliable, no error detection or error recovery, ICMP: error dectection and error recovery.

Type code: 11 0, TTL expired
Type code: 3 3, unreachable destination port

Tracerouter
A series of UDP segments with unreachable destination port number, each has an increasing TTL number starting from 1, i.e. the first segment has TTL = 1, the second has TTL = 2, …….
When nth set arrives at the nth router, since the TTL is expired, the router will return an ICMP message with type 11, code 0, which also include the name and IP address of the router. When ICMP message arrives, source records RTT from the timer.
When the UDP segment finally arrives at the destination, since the port numbers are all unreachable, the destination will send back an ICMP with type 3, code 3, when source received this message, the source stops.

Question: difference between the routing using match-action in Open Flow and the traditional routing using a link-state protocol
Answer:
Traditional routing:
- Using forwaring table on individual routers
- Dijkstra algorithm is implemented on each router

Openflow routing:
- Using match-action table
- OpenFlow is centralised in controller, operates between controller, switch
- Match-action table comupation is done in SDN controller, after computation, the OpenFLow will install the tables

Link layer: responsible for transferring datagram from one node to physically adjacent node over a link

Link layer sercives:
- Framing, link access
- Reliable delivery
- Flow control
- Error detection | correction
- Half-duplex and full-duplex

Question: Why both link-level and end-end reliability
In network, not all hosts will go to the transport layer (such as router), when come to the network layer. If the error happen on the very beginning, it can not be solved.

Parity checking
Cyclic Redundancy Check (CRC)

Three types of Multiple access protocols:
- Partition channel
- Random access
- Take turns

Random access protocols
Types of random access protocols:
- ALOHA: slotted ALOHA | Pure ALOHA
- CSMA: CSMA/CD | CSMA/CA

Slotted ALOHA
Pros:
- Single node — transmit at full rate of channel
- Highly decentralized — only slots in node need to be in sync (有node的slot才用同步)

Cons:
Wasting slots (2 reasons): 1. collision, wasting slots 2. idle slots
Clock synchronization

Difference slotted ALOHA and pure ALOHA
1. Different operation mechanism: For slotted ALOHA, when frame arrives, frames only transmit in the next time slot. For pure ALOHA, when new frame arrives, it will transmit immediately
2. Different operation efficiency: The efficiency of pure ALOHA is only half of that for slotted ALOHA
3. Different operation complexity: Pure ALOHA is simple and no synchronization, while slotted ALOHA needs synchronization

Question: CSMA operation mechanism
If channel sensed idle: transmit entire frame
If channel sensed busy: defer transmission

CSMA collisions
Collissions: caused by propagation delay

CSMA/CD (collision detection)
- In wireless LANs: difficult — received small signal strength (channel fading), overwhelmed by local transmission strength

Ethernet CSMA/CD algorithm
1. NIC receives datagram from network layer, creates frame
2. Sense idle, starts transmission, sense busy, waits until channel idle, than transmit
3. NIC transmits without detecting another transmission, success
4. Detects another transmission, aborts and sends jam signal
5. After m th collision, NIC chooses K at random from $\{0,1,2,\ldots,2^m - 1\}$. NIC waits $K \cdot 512$ bit times, return to 2 (binary backoff)

CSMA/CD requires a frame to have minimum size
If two frames are transmitted within certain times, collision will happen. Both senders will detect collision. The senders can only detect collision during transmission. If the transmission is long enough, senders can dectect the collision. For a given channel rate, the corresponding minimum frame size will be obtained.

The minimum packet should transmit at least twice the propagation delay
Consider two nodes: node A and node B are on two distant sides. The one-way trip time is denoted as T. At a certain time T_0, node A is sending frames. Node B sends frame just before A's grame arrives at B. Then, we can approximately assume that node B transmits its frame at $T + T_0$. After $T + T_0$, node B detects collision due to the frame A's arrival. After another one-way trip time T, node B arrives at node A, node A detects collision due to frame from node B at $T + 2T_0$. From T_0 to $T + 2T_0$, node A should keep transmitting, since collision can be detected only during transmission.

CSMA/CA (collision Avoidence)
The hidden station problem
A and C are blocked by obstacles, A can't hear from C. When C is transmitting to B, since node A cannot detect the existence of C, it will also transmit to B. Collision occurs.

The exposed station problem
B wants to send to C, but B is prevented from transmitting to C as it concluds after carrier sense that it will interference with the transmission to A, however B can still transmit to C since C is out of range of A.

RTS: request-to-send is used by a sender to ask for permission to send a message and reserve space on the channel
CTS: clear-to-send indicates that the sender is allowed to send its message, others delay
ACK: acknowledgement is used to indicate that a message has been successfully received

Operation:
1.Sender first transmits small RTS packets to base station (BS) using CSMA to reserve the channel. 2. If this is received, BS broadcasts CTS in respond to RTS which is heard by all nodes. If the sender does not hear any CTS, then it will "backoff" and send another RTS. 3. If the sender gets a CTS, it will transmit the data frame. 4. After transmission, BS send an ACK to all nodes to declare that it has received the data and transmission is over.

IEEE 802.11 MAC Protocol: CSMA/CA
1. If sense idle for DIFS transmit entire frame
2. If sense channel busy, start random backoff time, timer counts down while channel idle, transmit when timer expires, if no ACK, increase random backoff interval, repeat 2

Difference between MAC address and IP address
IP address: 32-bit, used for layer 3 forwarding, not fixed, hierarchical address not portable
MAC address: 48-bit, used for layer 2, fixed, portable (can move LAN card from one to another)

ARP (address resolution protocol)
ARP: determine interface's MAC address when knowing its IP address

DHCP (dynamic host configuration protocol)
DHCP: allows machines with a layer 2 (MAC) address to obtain a layer 3 (IP) address

ARP protocol: same LAN
Operation
A sends datagram to B. First, A broadcasts ARP query, containing B's IP address, the detination MAC address is : FF-FF-FF-FF-FF-FF, all nodes in the LAN received this ARP query (broadcast). Then, B with matching IP address receives ARP query and replies to A with B's MAC address, frame sent to A's MAC address directly (unicast). Finally, A caches the IP-to-MAC address pair in its ARP table until information time out

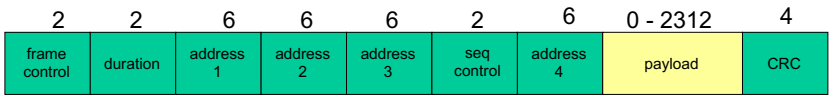
Addressing: routing to another LAN
Operation
1.A creates IP datagram (src IP: A, dest IP: B)
2.A creates frame (src MAC: A, Dest MAC: R (left-side))
3.R received frame from A. Datagram removed (demux), passed up to IP
4.R forwards datagram (src IP: A, dest IP: B), R creates frame (src MAC: R (right-side), dest MAC: B) (encapsulate)
(note: frame encapsulate datagram)

Ethernet frame structure
preamble: 7 bytes with pattern 10101010 followed by one byte with pattern 10101011, used synchronize receiver, sender clock rates
Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

Switch self-learning
Switch forwarding table: used to determine which switch interface each node is reachable (MAC address | Switch interface pair)

Two modes for wireless network: infrastructure mode, ad hoc mode
Infrastructure mode
Basic Service Set (BSS) contains
- Wirless hosts
- Access points (AP): Base station

802.11 frame: addressing



Address 1: MAC address of wireless host or AP to receive this frame
Address 2: MAC address of wireless host or AP transmitting this frame
Address 3: MAC address of router interface to which AP is attached
Address 4: used only in ad hoc mode

VLAN作用: VLAN is used to group devices logically for easy management

VLANs (Virtual Local Area Network): switch supporting VLAN capabilities can be configured to define multiple virtual LANS over single physical LAN infrastructure

MPLS (Multiprotocol label switching)
目的: high-speed IP forwarding using fixed length label (instead of IP address)

IP routing: path to destination determined by destination address alone
MPLS routing: path to destination can be based on source and destination address
- Fast reroute: precompute backup routes in case of link failure

Firewalls
Firewall: isolates oeganization's internal net from larger Internet, allowing some packets to pass, block others

incoming: come into internal network
outgoing: do out from an internal network

Three types of firewalls
- Stateless packet filters
- Stateful packet filters
- Application gateways

Reasons for having firewalls
- Prevent denial of service attack
  - SYN flooding: attacker establishes many bogus TCP connections, no resources for "real" connections
- Prevent illegal modification/access of internal data
- Allow only authorized access to inside network

Stateless packet filtering: Router filters packet-by-packet, decision to forward/drop packet

Example: Assume that the filter blocks inbound(进入的) TCP segments with ACK=0. What is the result.
result: prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

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

1.source: internal network, dest: large internet. Allow all the internal network client with TCP segment going out to the outside internet with port 80. All the internal client can make TCP connection with the outside network.
2.source: outside network, dest: internal network. Allow all the outside network to send TCP segment with port 80 and flag bit ACK into the internal network. Since when TCP connection is setting up, first ACK is always 0, so it prevents external clients from making TCP connections with internal clients.
3.4. allow DNS segment go inside or outside the internal network

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
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	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	---	X
deny	all	all	all	all	all	all	

Stateful packet filtering
Track status of every TCP connection
- timeout inactive connections at firewall: no longer admit packet
- need to check connection state table before admitting packet

If there is an attacker sending a TCP source 80 and ACK flag bit into the internal network. The connection state table need to be checked before go into the network. If this segment is not the part of TCP connection, then filter out.

Difference between stateful and stateless packet filtering
Stateless: each packet is filtered independetnely
Stateful: track the status of every TCP connection

Application gateways (网关)
Filter packets not only on IP/TCP/UDP fields, but also the application data
1. Require all telnet users to telnet through gateway
2. For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections.
3. Router filter blocks all telnet connections not originating from application gateway

Infrastructure for network management
Definitions: managed devices contain managed objects whose data is gathered into a Mangement Information Base (MIB)

SNMP: a protocol that facilitates the exchange of management information, between network devices
作用： to control and monitor status of network devices

SNMP Basic Components
- Network Management System (NMS): Executes applications that monitor and control managed devices
- Agent: a network-management software module that resides in a managed device
- Management Information Base: Used by both agent and manager to exchange and store management information