Chapter 3. Transport layer

3.1 Transport layer services

- TCP: reliable, in-order delivery
    - congestion control
    - flow control
    - connection needs to be set up
- UDP: unreliable, out of order
    - no-frills extension of best effort IP

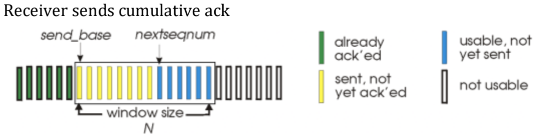
3.2 Multiplexing and demultiplexing

- Multiplexing:
    - Sender handling data from multiple sockets
    - Demultiplexing at receiver: use header info to deliver received segments to correct socket
- Demultiplexing
    - host receives IP datagrams: source and destination IP
    - datagram: IP addresses and port numbers to direct segment to socket
- Connectionless demultiplexing
    - Datagram sending to UDP socket must have destination IP addr and port#
    - From different source with same destination will be sent to the same socket
- Connection-oriented demultiplexing
    - 4-tuple: src+dst IP+port
    - demux: receiver uses all
    - simultaneous TCP sockets
    - web servers have different sockets for each connecting client

3.3 Connectionless transport: UDP

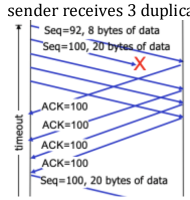
- No handshaking, each segment handled independently
- No congestion controls
- use streaming multimedia, DNS, SNMP
- Header
    - source port number, dest port number, length of segment, checksum (32bits)
    - payload

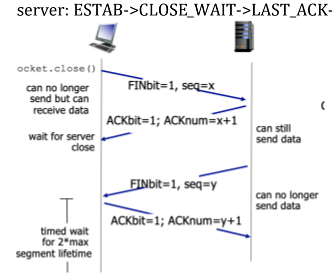
3.4 Principles of reliable data transfer

- rdt1.0: reliable channel
    - underlying channel is perfect, no bit errors, no packet loss
    - separate FSM for sender and receiver
- rdt2.0: channel with bit errors
    - checksum to detect bit errors
    - error recovery
        - ACK: receiver tells sender that pkt received OK
        - NAK: receiver tells sender that pkt had errors, sender retransmits
    - Corrupted ACK/NAK
        - duplicates
        - Sender adds a sequence number to each packet, receiver discards duplicate packet
        - Stop and wait: sender sends one packet and waits for receiver response before sending the next
- rdt2.1: sender, handles garbled ACK/NAKs
- rdt2.2: a NAK-free protocol
    - receiver must include sequence number of pkt being ACKed
    - Sender/receiver fragment
- rdt3.0: channels with errors and loss
    - sender wait until timeout and then retransmit
    - if no loss just delays -> duplicate ACKs, handled by sequence number because receiver needs to specify seq# of pkt being ACKed.
    - Performance calculation of rdt3.0
        - 1Gbps link, 15ms propagation delay, 8000bit packet
            - D_trans =8000/10^9=8 microsecs
            - Fraction of time sender busy sender:
                - U_sender = L/R/(RTT+L/R) = 8us/(30ms+8us) = 0.00027
- Pipelined protocols
    - Sender can have up to N unacked packets in pipeline
    - Go-back-N (GBN)
        - Receiver sends cumulative ack
        - 

        - Timer for the oldest unacked packet, retransmit
        - ACK only: send ACK for correctly received pkt with highest sequence number
        - No receiver buffering, discard out of order packet, need to resend all arriving packets after first lost packets
    - Selective repeat (SR)
        - Receiver:
            - sends individual ack for each packet
            - [base, base+N-1] Buffer out of order, deliver in order
            - [base-N, base-1] ACK
        - Timer for each unacked packet, retransmit only that unacked packet
        - Sender only resends unreceived packets

3.5 Connection-oriented transport: TCP

- Overview
    - One sender, one receiver.
    - RDT
    - Pipelined, TCP congestion and flow control sets window size
    - Bi-directional data flow in same connection
    - Connection oriented
- TCP segment
    - source and destination port #
    - Seq num: byte stream number of first byte in segment's data
    - ACK num: sequence number of next byte expected from the other side
    - ...
    - checksum: pseudo-header from IP header + TCP header + TCP payload
    - TCP round trip time (RTT), timeout
        - estimatedRTT = (1-0.125) *estimatedRTT + 0.125*SampleRTT
        - DevRTT = (1-0.25) *DevRTT+0.25*|SampleRTT-estimatedRTT|
        - TimeoutInterval or RTT = estimatedRTT + 4*DevRTT
- Reliable data transfer
        - Pipelined segments, cumulative acks, single retransmission timer
        - Retransmit upon timeout or duplicate acks
    - TCP Sender Events
        - create segment with seq# from receiver (rcvr's ack)
        - start timer if not ready done so
        - timeout + retransmit + restart timer
        - ack rcvd
    - TCP ACK generation
        - in-order, all data up to seq# acked => delayed ack, wait up to 500ms for next segment, if no, send ack
        - in-order, one ack pending => send single cumulative ack, acking both in-order segments
        - out of order, higher than expected seq#, gap! => send duplicate ack, indicating expected byte
        - filling gap => send ack
    - TCP fast retransmission
        - sender receives 3 duplicate ACKs -> resend unacked segment with smallest seq#
        - 

- Flow control
    - receiver-side buffering
    - rwnd: free space in the receiver buffer
- Connection management
    - Three-way handshake: LISTEN->SYN SENT+RCVD->ESTAB
        - client request connection: SYN on, SEQ c
        - server accept connection: SYN on, SEQ s, ACK on, ACK = c+1
        - client send data: ACK on, ACK = s+1
    - Closing a connection
        - client: ESTAB->WAIT_FIN1->WAIT_FIN2->TIMED_WAIT->CLOSED
        - server: ESTAB->CLOSE_WAIT->LAST_ACK->CLOSED
        - 

        - client wants to close connection: FIN on
        - server respond to FIN: ACK on
        - server wants to close connection: FIN on
        - client respond to FIN: ACK on

3.6 Principles of congestion control

3.7 TCP congestion control

- Overview
    - last_byte_sent – last_byte_acked <= cwnd
    - rate = cwnd/RTT bytes/sec
    - loss event = timeout or 3 dup
    - TCP reduces cwnd after loss event
- AIMD rule: additive increase, multiplicative decrease
    - cwnd += 1MSS before loss
    - cwnd /= 2 after loss
- Slow start
    - when connection begins, cwnd = 1mss
    - increase rate exponentially until threshold
- Congestion avoidance
    - cwnd > ssthresh, cwnd+(MSS*MSS)/cwnd upon every incoming nonduplicate ACK
    - 3 duplicate ACK: enter fast retransmission
    - Retransmission timeout: reset everything
- Fast retransmission/fast recovery
    - Fast retransmission
        - 3 duplicate ACK to indicate packet loss
        - reduce threshold: ssthresh = max(cwnd/2, 2*MSS)
        - cwnd = ssthresh + 3MSS
        - retransmit lost packet
    - Fast recovery
        - increase cwnd by 1 MSS upon every additional duplicate ACK
        - transmit new packets if allowed by the updated cwnd
        - upon aa new ack, cwnd = ssthresh
- Retransmission upon timeout
    - Reduce threshold: ssthresh = max(cwnd/2, 2*MSS)
    - Reset cwnd = 1MSS
    -
- TCP throughput
    - avg TCP throughput = ¾*W/RTT bytes/sec, W is window size
    - L: segment loss probability
        - throughput = 1.22*MSS/(RTT*sqrtL)

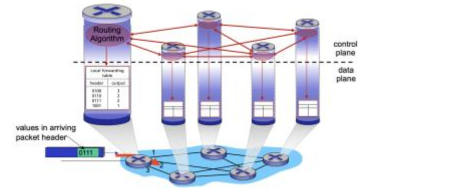
Study guide questions:

- Can you enumerate all the basic mechanisms needed to ensure reliable data transfer?
- How to handle the following scenarios (if any exists) using Stop-and-Wait, Go-back-N, or selective repeat?
    - Packet loss
    - Packet corruption
    - Corrupted ACK
    - Lost ACK
    - duplicate packets
    - Out-of-order packet delivery
- TCP round-trip estimation and timeout
    - Is the SampleRTT computed for a segment that has been retransmitted? Why?
- What is the negative effect if the timeout value is set too small, or too big?
- Why does sampleRTT fluctuate?
- how does TCP readjust its timer? (see lecture slides)
    - When receiving a new ACK
    - When receiving a duplicate ACK?
    - When the current timer expires for N times?
- What is 3-way handshake?
    - How are the initial seq, ACK #, etc. decided?
- Are the TCP connection setup and teardown identical?
    - Why are they different?
    - Why do you need so many states in the FSM model for TCP connection?

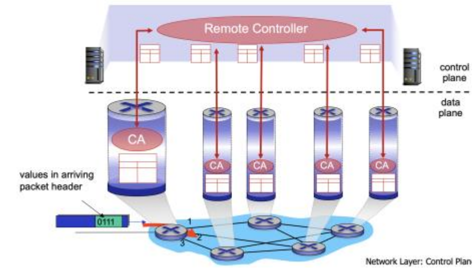
Chapter 4: Network layer, Data plane

4.1 Overview of network layer

- What does network layer do?
    - transport segment from sending to receiving host through all the network layers
    - Sender: wrap segments into datagrams
    - Receiver: deliver segments to transport layer
    - There is a network layer in every host and router along the path
    - Router examines header fields in all IP datagrams passing through it
- Network-layer functions
    - forwarding: move packets from router's input to appropriate router output
        - getting through a single interchange
    - routing: determine which route the packet goes from source to destination
        - planning trip
    - Data plane
        - local, per-router function
        - determine how datagram arriving on router input port is forwarded to router output port
        - forwarding function
    - Control plane
        - netword-wide logic
        - determine how datagram is routed among routers along end-end path from source to destination (routing function)
        - Two control-plane approaches:
            - Traditional routing algorithm: implemented in routers



            - Software-defined networking (SDN): implemented in remote servers



- Network service model
    - What service model is for channel transporting datagrams from sender to receiver?
        - example services for individual datagrams:
            - guaranteed delivery with less than 40ms delay
        - example services for a flow of datagrams:
            - in-order datagram delivery
            - guaranteed minimum bandwidth to flow
            - restrictions on changes in inter-packet spacing

4.2 What's insider a router?

- Router architecture
    - Routing control plane (Software)
        - Routing processor
    - Forwarding data plane (Hardware)
        - Router input ports -> high-speed switching fabric (controlled by routing processor) -> router output ports
        - Input port functions
            - Line termination [physical layer: bit-level reception]
            - -> link layer protocol(receive) [data link layer: ethernet]
            - -> lookup, forwarding, queuing
                - Decentralized switching:
                    - Header filed values: lookup output port using forwarding table in input port memory
                    - Goal: complete input port processing at line speed
                    - Queueing if datagrams arrive faster than forwarding rate
                    - Destination-based forwarding: based on destination IP address (traditional)
                    - Generalized forwarding: based on any set of header field values
            - -> switch fabric
            - Destination-based forwarding
                - What happens if ranges don't divide in a forwarding table? -->
                - Longest prefix matching: when looking for forwarding table entry for given destination address, use longest address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

ples:

DA: 11001000 00010111 00010110 10100001 which interface? 0

DA: 11001000 00010111 00011000 10101010 which interface? 1 rather than 2

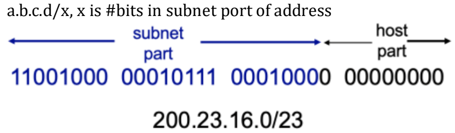
                    -
                    - Performed using ternary content addressable memories, retrieve address in one clock cycle regardless of table size
                    - Why?
            - Input port queueing

- queueing delay and loss due to input buffer overflow
- head of the line(HOL) blocking: line front blocks others
    - Switching fabric
        - Transfer packet from input buffer to appropriate output buffer
        - Switching rate: measured as multiple of input/output line rate (N inputs, N times line rate desirable)
        - Three types of switching fabrics:
            - Memory
                - packets are copied to system's memory
                - limited by memory bandwidth
            - Bus
                - datagram from input port memory to output port memory via a shared bus, one datagram at a time
                - limited by bus bandwidth
            - Crossbar
                - overcome bandwidth, fragment datagram into fixed length cells
    - Output port
        - Output port queueing
            - buffering required, otherwise packet loss, possible overflow too
            - scheduling datagrams
        - How much buffering is need?
            - arrival rate > output line speed
            - RTT_C/sqrt(N), C = link capacity(bps), N: #of flows
- Scheduling: FIFO, Priority, Round Robin: multiple classes, send one complete packet from each class

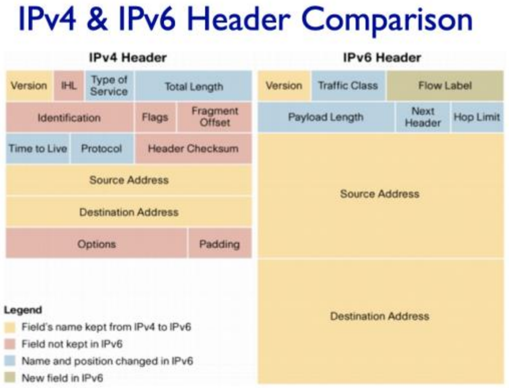
4.3 IP: Internet protocol

- Network layer: routing protocols(path selection) <-> forwarding table, IP protocol, ICMP protocol(error reporting, router signaling)
- Datagram format
    - 20 bytes of IP, total = 20 + tcp 20b header + app layer overhead
    - version, length, flags, time, source IP address, destination IP address, data
- packet handling conventions
    - IP fragmentation, reassembly
        - one datagram becomes several smaller datagrams
        - reassembled at destination
        - IP header bits to identify and reorder fragments

- IP internet protocol
    - IPv4 addressing conventions
        - 32-bit identifier for host, router, and interface(connection between host/router and physical link)
        - **Subnet** part (higher order bits)
            - device interfaces with same subnet part of IP address
            - can physically reach each other without intervening router
            - recipe:
        - Host part (lower order bits)
        - **CIDR** (classless interdomain routing)
            - a.b.c.d/x, x is #bits in subnet port of address



        - How to get an IP address
            - hard-coded
            - **DHCP: dynamic host configuration protocol**
                - DHCP discover: arriving client says is there a DHCP server?
                - DHCP offer: D server offers an IP addr
                - DHCP request: client takes the IP addr
                - DHCP ack: server says you got the IP addr
                - can return more allocated IP on subnet + first-hop router address + name and IP addr of DNS server + network mask
            - Example:
                1) DHCP gives connecting client its IP, first-hop router, DNS server addr
                2) DHCP request is wrapped in UDP, in IP, and in 802.11 ethernet
                3) DHCP server receives Ethernet frame broadcast on LAN
                4) Demux: ethernet demux to IP to UDP to DHCP (reverse (2))
                5) DHCP server formulate DHCP ACK containing (1)
                6) DHCP server wraps them in UDP, IP, Ethernet and send to client, client demux
            - How does network get subnet part of IP address?
                - gets allocated portion of its provider ISP(internet service provider)'s address space
- **Network address translation (NAT)**
    - Motivation
        - all datagrams leaving LAN have same source NAT IP address with different source port numbers
        - can change addresses of provider without notifying outside world
        - devices inside the local net are not addressable from outside (security)
    - Implementation
        - Outgoing datagrams: replace source IP with NAT IP and new port#
        - Remember in NAT table: pair
        - incoming datagrams: replace NAT with corresponding IP stored in NAT table
- **IPv6**
    - 40 byte header, no fragmentation
    - **datagram format:** include priority of datagram in a flow, flow label, next header, ICMPv6; remove checksum, options outside of header



    - **Tunneling**: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers

4.4 Generalized Forward and SDN

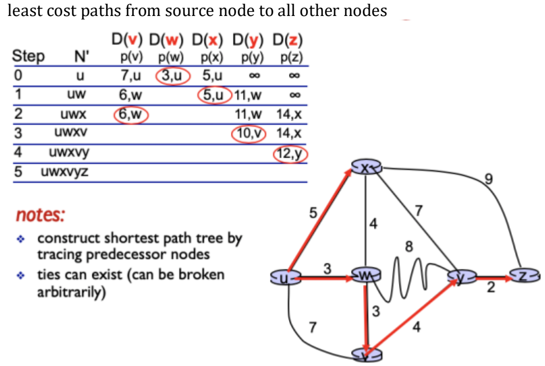
Chapter 5 Network layer: control plane

5.1 Introduction

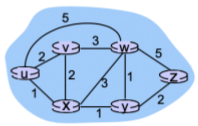
- Two approaches to structure network control plane:
    - per-route control (traditional)
        - individual routing algorithm components in each and every router, interact with each other in control plane, to compute forward tables
    - logically centralized control (software defined networking) SDN
        - A distinct remote controller interacts with local control agents (CA) in routers to compute forward tables

5.2 Routing protocols

- Link state algorithm
    - Global information, all routers have complete topology, link cost info
    - Static: routes change slowly/ Dynamic: routes change quickly, periodically, or in response to link cost change
    - Dijkstra's algorithm
        - least cost paths from source node to all other nodes



        - O(n^2)
- Distance vector algorithm
    - Decentralized information, router knows physically-connected neighbors only, iterative process of computation, exchange info of neighbors
    - Bellman-ford
        - D(x,y) = min{c(x,v)+D(v,y)} for all v.



clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\qquad c(u,x) + d_x(z), \\ &\qquad c(u,w) + d_w(z) \} \\ &= \min \{2 + 5, \\ &\qquad 1 + 3, \\ &\qquad 5 + 3\} = 4 \end{aligned}$$

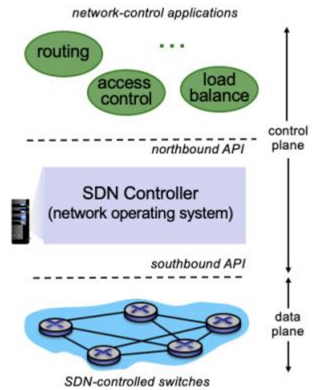
- **node achieving minimum is next hop in shortest path, used in forwarding table**
- Each node sends its own distance vector estimate to neighbors, when x receives its neighbors' DV, updates its own DV using the equation above
- Converge to actual least cost D(x,y)

- **LS vs. DV**
  - Message complexity: DV > LS=O(|nodes|*|links|)
  - Speed of convergence: DV varies, LS O(n^2)
  - Robustness: LS (each node computes its own table) > DV (might have incorrect path cost, others use your table, error propagates)

5.3 Intra-AS routing in the internet: OSPF
- AS: autonomous systems (domains)
- Intra-AS routing:
  - routing among hosts, routes in same AS
  - all routes run in same intra-domain protocol
  - gateway router: AS has links to routers in other ASes, perform intra/inter-domain routing
  - Common protocols:
    - RIP: Routing information protocol
    - OSPF: Open shortest path first
      - Link-state algorithm, topology map at each node, compute route via Dijkstra
      - Security: authenticated messages
      - **Multiple same-cost paths allowed (one for RIP)**
      - Multiple cost metrics for each link
      - Uni-/Multi-cast support
      - Hierarchical OSPF
        - local area each running its own OSPF link-state routing algorithm
        - area border routers: routing packets to outside areas
        - backbone area: route traffic between other areas in AS, contain all border routers and others.
        - link-state info only in area, each node has area topology and knows direction to nets in other areas
    - IGRP: Interior gateway routing protocol
- Inter-AS routing:
  - routing among AS'es
  - AS2 <-> AS1 <-> AS3
  - AS1 needs to:
    - Learn which destinations are reachable through AS2, which through AS3
    - Propagate reachability info to all routers in AS3

5.4 Routing among the ISPs: BGP (border gateway protocol)
- Inter-domain routing protocol, provides a means to:
  - eBGP session: obtain subnet reachability info from neighboring AS. (inter)
    - Destinations are not hosts but instead are CIDRized prefixes
    - eBGP sends prefixes
  - iBGP session: propagate reachability info to all AS internal routers. (intra)
    - distribute the prefixes to other routers in the AS
  - Allow subnet to advertise its existence to rest of internet
- BGP session
  - two BGP routers exchange BGP messages over TCP connection
  - AS 3 BGP advertisement to AS2: AS3, X; AS3 promises AS2 to forward data to X
- BGP path attributes
  - route = prefix + attributes, used to choose best routes
  - AS_PATH: list of ASes through which prefix ad has passed
  - NEXT_HOP: internal-AS router to next-hop AS, IP address of next-hop
  - policy-based routing: determine whether to accept a path or advertise a path
- Forwarding table entries
- BGP route selection
  - select local preference value attribute (also shortest)
  - shortest AS-path
  - closest NEXT-HOP router: hot potato routing
    - choose local gateway that has least intra-domain cost, get rid of traffic
    - don't worry about inter-domain cost
- Why different intra/inter AS routing?
  - Only inter needs policy. Intra can focus on performance.
  - Hierarchical routing saves table size, reduces update traffic

5.5 The SDN control plane
- Why?
  - Easier network management: avoid router misconfigurations, greater flexibility of traffic flows, difficult traditional routing when defining a custom route from u to z.
  - Table-based forwarding allows programming routers, centralized programming easier
  - Open implementation of control plane



- Data plane switches
  - switch flow table computed and installed by SDN controller
  - OpenFlow: API for switch control (define what is controllable), protocol for communicating with controller
    - Use TCP to exchange messages: controller->switch, asynchronous switch->controller, symmetric, OpenFlow tables
- SDN controller
  - Maintain network state info, interacts with network control app above, implemented as distributed system for performance, scalability, fault-tolerance, and robustness
  - Components
    - Interface layer to apps
    - network-wide state management layer (distributed database)
    - communication layer to switches
- Control applications
  - implement control functions with API provided by SND controller
  - can be provided by 3rd parties

Study guide questions:
- What is the Internet service model?
  - Channels transport datagrams from sender to receiver
  - guaranteed deliver for individual datagrams
  - in-order datagram flow delivery with minimum bandwidth to flow
- Comparing VC (virtual circuits) and datagram networks
  - Virtual circuits
    - Need connections at network layer
    - Reservation for resources like bandwidth, cpu,
    - Router needs to maintain connection state info
    - VC setup -> data transfer -> VC close
    - Signaling protocol
  - Datagram networks
    - Internet
    - Connectionless
    - stamps the packet with the destination address and deliver into the network
    - Pass though routers and link interfaces(prefix matching)
- How does a router decide which next hop to forward when a packet arrives?
  - forwarding table , longest prefix matching
- What is the rationale for each field in the IP packet header?
  - version number, header length, type of service, length, 16-bit identifier, flags, fragment offset, time to live, upper layer, header checksum, source IP, dest IP, option
- IP fragmentation & reassembly
  - One datagram becomes several smaller datagrams
  - reassembled at destination
  - IP header bits to identify and reorder fragments
- What is subset? What is CIDR?
  - subnet
    - devices in a subnet have same IP address's higher order
    - multiple host interface, one routing interface
    - reach each other without intervening router
  - CIDR
    - internet's address assignment strategy, classless interdomain routing

        - a.b.c.d/x. x = #bits in subnet port of address
- How does NAT work? What about DHCP?
- What fields exist in IPv4 but not in IPv6? What exist in IPv6 but not in IPv4? What exist in both?
- How does the tunneling technique work?
    - When you plan to deploy a new network technology on the global Internet, how do you address the issue of incremental deployment? Tunneling
- Compare link state routing and distance vector routing
- Given a network topology, apply link-state routing or distance vector routing algorithm to compute the minimum-cost path (exercise)
- What kind of info is propagated/collected in link state routing or distance vector routing? How many messages are propagated in each?
    - Link state packet: identities and costs of its neighboring links
    - Distance vector packet: receive info from neighbors, and send result of calculation back
- What is a potential problem with distance vector routing? How to address it?
    - An incorrect node calculation can be diffused through the entire network
    - Routing loops, count-to-infinity: when a route fails, spread bad news by poisoning the route
    - slower convergence
- Why does RIP limit the maximum hop count as 16? Can it fully address the count-to-inf problem?
    - Limit the use of RIP to autonomous systems that are fewer than 15 hops in diameter (#subnets traversed along the shortest path from source to destination router), to avoid count-to-inf problem,
    - No
- Can OSPF compute multiple same-cost paths? Yes
- Why intra-AS and inter-AS routing protocols are different?
    - Can BGP always compute the shortest path route? No, policy-based
    - Does the path vector in BGP include any router's IP address? Why? Yes, NEXT-HOP includes the IP address of the next router.
- What is the difference between hierarchical OSPF and BGP inter-domain routing?
    - Hier OSPF: intra-domain, can always divide large domain to smaller ones, focus more on performance
    - BGP: inter-domain, policy is more important in BGP (carried in path attributes), focuses more on scalability
- What is longest prefix matching rule?
    - The first x bits are used to determine subnets because these devices share the common prefix.
    - due to CIDR, subnet addressing
- Compare SDN routing and the current Internet routing
- Compare SDN and router-based data forwarding
-
- How do iBGP and eBGP work?
- How is the path vector computed?
- Given a topology, how does the BGP advertise the path vector?
    - Look at the example in the lecture notes
- Can BGP lead to routing loop? Why?
- How does BGP work with intra-AS routing?
    - How is the BGP reachability info propagated within an AS and across Ases?
- What is hot potato routing? How does it play in the Internet routing in reality?

Chapter 6 Link layer and LANs

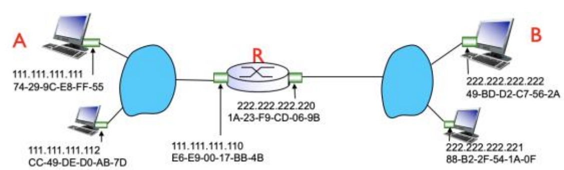
6.1 Introduction, services
- Definition
    - Node: hosts and routers
    - Links: connect adjacent nodes
        - wired, wireless, LAN
    - Frame: encapsulates datagram, layer-2 packet
    - Data-link layer: transfer datagram from one node to physically adjacent node over a link
- Link layer services
    - framing, link access
        - encapsulate datagram into frame, adding header, trailer, channel access
        - MAC addresses used in frame headers to in frame headers to identify source, destination (not IP as in datagram)
    - reliable delivery between adjacent nodes (rdt in chapter 3)
    - flow control: pacing between sending and receiving nodes
    - error detection/corrections (correct bit errors w/o retransmission)
    - half-duplex (nodes at both ends can transmit, but not at the same time) and full-duplex
- Where is the link layer implemented?
    - In every host's adaptor (network interface card, NIC) or on a chip
        - Adaptor sending side: encapsulate datagram, add error checking, rdt, flow control...
        - Adaptor receiving side: look for error, rdt, flow control..., extract and pass datagram
    - attaches to host's system buses

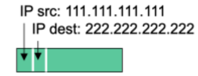
6.2 Error detection, correction
- EDC: error detection and correction bits
    - not 100% reliable, the larger the better
- Parity checking
    - single bit parity
    - 2D bit parity: row parity and column parity, can correct single bit error
- Internet checksum
    - Sender: addition of segment contents as 16-bit int, put checksum into UDP checksum field
    - Receiver: compute checksum of received segment, check checksum
- Cyclic redundancy check
    - D*2^r XOR R

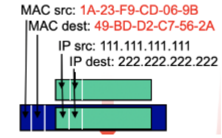
6.3 Multiple access protocols
- Point-to-point link: between ethernet switch, host
- Broadcast link: old-fashioned ethernet
- Collision if node receives two or more signals at the same time, so we need:
- MAC protocols
    - channel partitioning
        - divide channel into fixed-size smaller time slots, frequency, code; allocate piece to node for exclusive use
        - access to channel in rounds
        - TDMA: time division multiple access (length = packet transmission time)
        - FDMA: frequency
    - random access
        - when a node has a packet to send, send in full channel data rate at R, but channel is not divided -> possible collisions with multiple nodes, but can recover from them
        - Slotted ALOHA
            - Assume all frame and time slots same size, nodes are synchronized, if 2 or more nodes transmit in a slot, all nodes detect collision
            - When a node obtains fresh frame, transmit in next slot
                - No collision: send new frame in next slot
                - Collision: retransmits frame with prob. b
            - Pros: single node transmits at full rate of a channel, decentralized (only slots in nodes need to be in sync), simple
            - Cons: collisions, wasting slots, clock synchronization, detect collision in less than time to transmit packet
            - Efficiency calculation
                - Np*(1-p)^(N-1), N=#Nodes, p=each node's probability to transmit in the current slot
                - As N -> ∞, 1/e = 0.37 -> channel useful transmission 37% of time
        - ALOHA
            - No synchronization, transmit immediately when frame first arrives, more collisions due to overlapping
            - Efficiency calculation
                - p(1-p)^2(N-1), no other transmission before and after this node transmission
                - 1/2e = 0.18
        - CSMA (carrier sense multiple access)
            - Carrier sense: monitor when the channel is busy. If a channel is busying, some other node is transmitting., defer your transmission if channel is busy via carrier sense. (Don't interrupt)
            - Collision can still occur
                - Propagation delay means two nodes may not hear each other's transmission
                - waste entire packet transmission time
                - physical distance limit
        - CSMA/CD:
            - collision detection is short, colliding transmission abort -> not so much waste
            - collision detection
                - Same as ALOHA
                - easy in wired, hard in wireless
            - local area network is limited by the physical distance between nodes
            - Ethernet CSMA/CD algorithm
                - Is a frame ready? (API to upper layer)
                - (carrier sense) Is the channel idle? If not, wait until the channel is idle.
                - Start transmission and monitor the channel.
                - If a collision occurs, go to collision resolution
                    - insert transmission with a jam signal so that all receivers can detect collisions.
                    - increment retransmitted counter. If reach max value and a collision is still perceived, declare failure.
                    - Was the max number and transmission attempts reached? If so, abort transmission. (for fast loss recovery; Collision induces loss, not for reliable data transfer).
                - Otherwise, reset retransmitted counter and complete the frame transmission
                - After aborting, enter binary backoff
                    - BEB: calculate backoff and wait for the random backoff period.
                    - go to step 1
                    - After mth collision, NIC chooses K at random from {0, 1, 2, ..., 2^m-1}. The node with smallest K value will usually succeed (due to carrier sense of other nodes)
                    - Longer backoff interval with more collisions

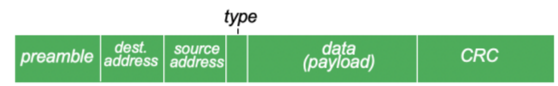
- CSMA/CD efficiency
  - efficiency = 1/(1+5t_prop/t_trans)
  - efficiency goes to 1 as t_prop goes to 0 or as t_trans goes to infinity (infinitely big frame size)
- Taking turns
  - Noes take turns, but nodes with more to send take longer
  - polling
    - Master node invites slave nodes to transmit in turn
    - Typically used with dumb slave devices
    - Concerns: polling overhead, latency, single point of failure (master)
  - Token passing
    - control taken passed from one node to next sequentially
    - Token message
    - Concerns: token overhead, latency, single point of failure
    - Total design fails if a token can't circulate
- Summary of MAC protocols
  - channel partitioning
    - By time, frequency, or code
    - efficiently and fairly at high load but inefficient at low load
  - random access
    - ALOHA, S-ALOHA, CSMA, CSMA/CD
    - carrier sensing: easy in wire, hard in wireless
    - CSMA/CD in Ethernet
    - CSMA/CA in 802.11
    - efficient in low load where single node can fully utilize channel
  - taking turns
    - polling from central site, token passing
    - Bluetooth, FDDI, token ring

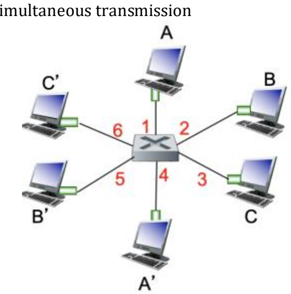
6.4 LANs

- MAC addressing (like SSN), ARP
  - MAC address
    - used locally to get frame from one interface to another physically-connected interface (same network, in IP addressing sense)
    - 48-bit, hexadecimal notation
  - ARP: address resolution protocol
    - ARP table: determine interface's MAC address, knowing its IP address.
      - ARP table records the mapping
        - <IP addr   MAC address   Timer>
        - Timer: Time to live, TTL
          - Time after which address mapping will be forgotten (20min)
    - ARP protocol: must be in the same LAN
      - A wants to send datagram to B, but B's MAC address is not in A's ARP table.
      - A broadcasts ARP query packet, containing B's IP address. Heard by all nodes on LAN.
      - B receives the ARP packet, replies A with B's MAC address. Frame sent to A's MAC address (unicast)
      - A caches(saves) IP-to-MAC address pair in its ARP table until information times out
      - ARP is plug-and-play: nodes create their ARP tables without intervention from net administrator
    - Addressing: routing to another LAN
      - 

      - Send datagram from A to B via R
        - A knows B's IP address (DNS lookup), R's IP address (DHCP), R's MAC address (DHCP+ARP)
        - Steps:
          - 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, received at R, datagram removed, passed up to IP
          - Drop if CRC has a problem, otherwise continue checking IP header.
          - 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 packet:
          - 

          - IP src: A's IP
          - IP dest: B's IP
        - Actual datagram
          - Frame header:
            - 

R-B packet
            - MAC src: A's MAC address
            - MAC dest: R's MAC address
            - IP packet
            - Error checking CRC
- Ethernet
  - Dominant wires LAN technology
  - Simple and cheap, 10Mbps-10Gbps
  - Physical topology
    - Bus: all nodes in same collision domain (can collide with each other)
    - Star: active switch in center, each spoke runs a separate Ethernet protocol (nodes do not collide with each other)
  - Frame structure
    - 

    - Preamble
      - used to synchronize receiver, sender clock rates
      - 7 bytes with pattern 10101010 followed by 10101011
    - Address
      - 6 bytes source, destination MAC addresses
      - If adapter receives frame with matching destination address or broadcast address (ARP packet), it passes data in frame to network layer protocol. Otherwise adapter discards frame.
    - Type: higher layer protocol (IP, AppleTalk)
    - CRC
  - Unreliable and connectionless
    - Connectionless: no handshaking between sending and receiving NICs
    - Unreliable: receiving NIC doesn't send acks or nacks to sending NIC, data in dropped frame are recovered only if initial sender uses higher layer RDT, otherwise drop lost data
    - MAC protocol: unslotted CSMA/CD with binary backoff
- Ethernet switch
  - Link-layer device
    - store, forward ethernet frames
    - examine incoming frame's MAC address, selectively forward. use CDMA/CD to access segment
  - Transparent: hosts are unaware of presence of switches
  - Plug and play, self-learning
    - Learns which hosts can be reached through which interfaces
    - When frame received, switch learns location of sender via LAN segment
    - records sender/location pair in switch table
    - require no intervention from a network administrator or user.
  - Multiple simultaneous transmission
    - 

    - A to A', B to B' can transmit simultaneously, without collisions. Destinations are different.
  - Switch forwarding table
    - <host MAC address, interface to reach host, time stamp> (like a routing table)

- self-learning, works also for interconnecting switches
  - Initially empty table
  - For each incoming frame received on an interface, store in table
  - Deletes an address if no frames are received with that address as the source after some period of time
- Switch vs. routers
  - Both are store and forward
    - routers are network-layer devices
    - switches are in link-layer devices
  - Both have forwarding tables
    - routers compute tables using routing algorithms, IP addr
    - switches: self-learning, flooding, MAC addr

6.7 A day in the life of a web request

- Scenario: student attaches his device to campus network, requests/receives a web page
  (1) network layer: DHCP gives client IP address, name and addr of DNS server, IP address of its first hop router
    a. DHCP request for <client's IP addr, name and addr of DNS, IP of first hop router> encapsulated in UDP, IP, 802.3 Ethernet
    b. 802.3 Ethernet frame broadcasts on LAN, received at DHCP server
    c. Demux at DHCP server: Ethernet -> IP -> UDP -> DHCP
    d. DHCP server formulates DHCP ACK containing <client's IP addr, name and addr of DNS, IP of first hop router>
    e. Demux at client: frame forwarded through LAN (switch learning) to client
    f. DHCP client receives DHCP ACK reply
  (2) link layer: ARP and DNS give client the web page's IP address
    a. DNS query is created, encapsulated in UDP, IP, Ethernet.
    b. ARP query broadcasts, received by router, which replies with ARP reply containing MAC address of router interface.
    c. Client sends the DNS query frame via LAN switch to first hop router.
    d. DNS query IP datagram forwarded from campus network into comcast network, routed to DNS server via routing table created by routing protocols (OSPF, BGP)
    e. DNS server: demux and reply to client with IP address of the web page.
  (3) transport layer: TCP connection carrying HTTP
    a. Client opens TCP socket to web server
    b. Client TCP three-way handshake with web server
      i. Routed SYN segment via inter-domain routing
      ii. Server responds with TCP SYN ACK
      iii. Connection established
  (4) application layer: HTTP request/request
    a. Client sends HTTP request into TCP socket
    b. IP datagram containing HTTP request routed to the web page
    c. Web server responds with HTTP reply containing web page
    d. IP datagram containing HTTP reply routed back to client

Study guide questions

- How does the binary exponential backoff work?
- How does ARP work? Is it using soft-state (i.e., maintaining timers for its state information)? Yes
- Compare the efficiency of CSMA/CD, ALOHA and slotted ALOHA?
  – Where does the saving come from in CSMA/CD?
- Is DHCP a soft-state protocol? Why? Yes, it will eventually forget the past information. Associate TTL with the info.
- Can ARP work in point-to-point link, rather than broadcast medium?
- What is the difference between a switch and a router?
- Which device can isolate collision domains? Switch, in a LAN built from switches, there is no wasted bandwidth due to collision.
- Given a scenario, use the appropriate devices (hub, switch, and router) to interconnect hosts to form a large network. (Data center networking)
  – Interconnecting LANs with routers by using IP addresses instead of physical address like MAC.
- How does the self-learning algorithm work?
- What protocols are used in web browsing, file transfer or email checking?
  – Which service is accessed first, DNS or DHCP?
  – How do you find out the DNS server via DHCP?
  – For the UDP/TCP segments, can arbitrary source/destination ports be selected? Host specified.
  – How many times is ARP used? Can ARP messages propagate to different subnets across routers? No.

Chapter 7

7.2 Wireless links, characteristics

- Elements of a wireless network
  - host: laptop, smartphone
  - base station: relay (sending data between wired network and wireless hosts)
  - wireless link
  - Infrastructure mode: base station connects mobiles into wired network, handoff reverses this process
    - mesh net
  - Ad hoc: no base stations, nodes transmit to each other link coverage, nodes organize themselves into network
- wireless link characteristics
  - compared with wired link: decreased signal length, interference from other sources, multipath propagation
  - SNR (signal to noise) vs BER (bit to error rate) tradeoffs
    - increase power -> increase SNR
- Code division multiple access (CDMA)
  - Unique code assigned to each user.
- IEEE 802.11 Wireless LAN
  - 802.11b 2.4-5 GHz, 11 Mbps, 11 channels at different frequencies
    - host must associate with an AP
  - 802.11a 5.6 GHz, 54 Mbps
  - 802.11g 2.4-5 GHz, 54 Mbps
  - 802.11n 2.4-5 GHz, 200 Mbps
  - LAN architecture
    - wireless host communicates with base station through access point (AP)
    - basic service set (BSS), aka cell: wireless host + access point + ad hoc hosts
  - Passive/active scanning
    - passive scanning
      - beacon frames sent from APs
      - association request frame sent: host to AP
      - association response frame sent: AP to host
    - active scanning
      - probe request frame broadcast
      - probe response frames sent from AP
      - association request frame sent: host to AP
      - association response frame sent: AP to host
  - MAC protocol: CSMA/CA (collision avoidance)
    - Sender
      - channel idles for DIFS -> transmit entire frame (no CD)
      - channel busy -> start random *backoff time*, timer counts down while channel idea, transmit when timer expires, if no ACK increase random backoff interval
    - Receiver
      - return ACK after SIFS (ACK needed due to hidden terminal problem)
    - Avoiding collisions
      - reserve channel rather than random access, using small reservation packets (RTS) via CSMA
      - Receiver send **CTS in response to RTS** to clear channel, heard by all nodes, sender can then transmit data frame
      - 802.11 frame 4 addresses: MAC of receiver host/AP, MAC of transmitter host/AP, MAC of router, (ad hoc)
  - Mobility within same subnet
    - Hosts remaining in same IP subnet have same IP address

7.4 Cellular Internet Access

- Components: cell -> mobile switching center -> public telephone network(wired)
  - cell [base station, mobile users, air-interface]
    - combined FDMA/TDMA: divide spectrum in frequency channels, divide channel into time slots
    - CDMA
  - Mobile switching center [connects cells to wired tel.net, manages call setup, handles mobility]
- 3G (voice + data)
  - operate in parallel with existing voice network (vn unchanged), data in parallel
  - radio interface + radio access network -> core network -> public internet
- 4G LTE
  - radio access network(E-UTRAN) -> evolved packet core -> public internet
    - UTRAN: eNodeB (connection mobility, radio admission control) + RRC + PDCP + ...
    - EPC:
      - MME (sets up eNodeB-PGW channel, UE transition)
      - S-GW (mobile anchoring): hold idle UE info, QoS enforcement
      - P-GW (allocate UP IP address, filter packets)
    - UE (user element) -> eNodeB -> PGW
      - IP packet from UE encapsulated in GPRS tunneling protocol (GTP) message at eNodeB
      - GTP encapsulated in UDP, and then in IP, large IP packet addressed to SGW
  - All IP core: IP packets tunneled from base station to gateway
  - No separation between voice and data, all traffic carried over IP core to gateway

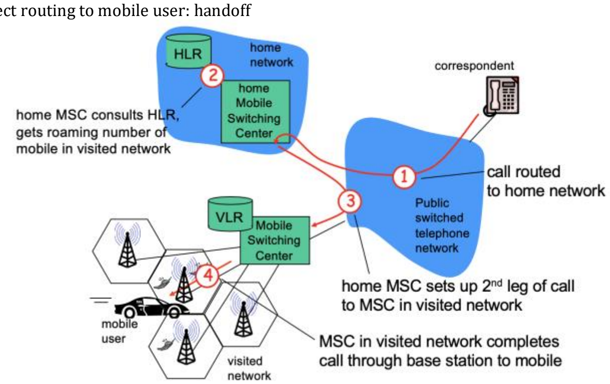
7.5 Principles: addressing and routing to mobile users

- Definition
  - home network: permanent home of mobile, permanent address can always be used to reach mobile
  - home agent: perform mobility functions on behalf of mobile, when it's remote
  - visited network: network in which mobile currently resides
  - care-of-address: address in visited network
  - correspondent: wants to communicate with mobile
  - foreign agent: entity is visited network performing mobility functions on behalf of mobile
- Find changing address
  - Routing: (not scalable)
    - routing table: routers advertise permanent address of mobile-nodes-in-residence
    - no changes to end-systems
  - End-systems:

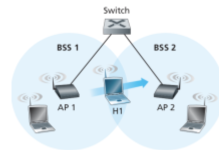
- **indirect routing**: correspondent -> home agent -> mobile (permanent addr used by corresp, careofaddr used by home agent), inefficient when correspondent and mobile are in the same network (or triangle routing problem)
  - Steps
    (1) Correspondent addresses packets using home address of mobile
    (2) In home network, HA intercepts packets, forward to FA
    (3) FA received the packets, forward to mobile in visited network
    (4) mobile replies directly to correspondent
  - What if mobile moves to another network?
    - registers with new foreign agent
    - new foreign agent registers with home agent
    - home agent updates care-of-addr
    - continuing data with new coa
  - Triangle routing problem: datagrams addressed to mobile node must be routed first to the home agent then to the foreign network
- **direct routing**: correspondent gets foreign address of mobile -> send
  - Steps
    (1) Correspondent requests and receives foreign address of mobile via home network
    (2) Correspondent forwards to FA knowing its address
    (3) FA receives the packet and forwards to mobile in visited network
    (4) mobile replies directly to correspondent
  - **Overcome triangle routing problem**
  - Non-transparent to correspondent: must get coa from home agent
  - What if mobile moves to another network?
    - Data routed first to anchor FA (the FA in first visited network)
    - new FA arranges to have data forward from old FA
- Registration
  - visited network: mobile contacts FA -> wide area network -> home network: foreign agent contacts home agent "this mobile is in my network"
  - Now, FA knows about the mobile, HA knows the location of the mobile

7.6 Mobile IP
- Agent discovery: home/foreign agent advertises its services to mobile nodes
- Registration with the home agent: mobile nodes register/deregister COAs with agents
- Indirect routing of datagrams: datagrams are forwarded to mobile nodes by a home agent

7.7 Handling mobility in cellular network
- Home network (network carrier) Home location register: database storing phone#, location
- Visitor location register: entry for each user
- GSM: indirect routing to mobile user: handoff



Study guide questions:
- Which category of MAC does CDMA belong to? Random access
- The detailed operations of CSMA/CA.
  - What components are the same, or different between CSMA/CA and CSMA/CD?
    - Both listen before speaking and stops talking if someone else begins at the same time.
    - CA transmits a frame entirely.
    - CD terminates the current transmission as soon as a collision is detected.
- Why does not 802.11 MAC implement collision detection but uses collision avoidance?
  - The ability to detect collision requires the ability to send and receive at the same time, but the strength of received is signal is small compares to transmitted, Costly to build.
  - Hidden terminal problem – adapters still can't detect all collisions.
- What is the purpose to use link-layer acknowledgment in 802.11 MAC? lower bit error rates than wireless channels
  - Can TCP ACK replace it? Can MAC ACK replace TCP ACK?
- What is the mechanism to handle hidden terminals? RTC and CTS
- How to handle mobility in the same IP subnet?



  - When H1 moves from BSS1 to BSS2, it may keep its IP address and all of its ongoing TCP connections.
  - As H1 wanders aways from AP1, H1 detects a weakening signal from AP1 and starts to a scan the stronger signal of AP2.
  - Switch can self-learn the moves.
- How to do routing to a mobile host? Indirect routing and Direct routing
- How is mobility supported across different subnets?
  - Operations of home agent, foreign agent,
- How to avoid triangle routing (i.e., indirect routing where packets are forwarded to the home network, then the visited network of the mobile host) in mobility support?
- How can you know a mobile host's current location? Registration
- How does a mobile host update its location?

Chapter 8.

8.1 What is network security?
- Confidentiality: encryption
- Authentication: identity of sender and receiver
- Message integrity: message not changed
- Access and availability: for users
- Problems - Eavesdrop: intercept message, insert message, impersonation (fake source address), hijacking (removing sender/receiver), denial of service (overload resource)

8.2 Principles of cryptography
- Symmetric key cryptography (how to agree on keys?)
  - plaintext -> Key -> ciphertext -> Key -> plaintext
    - Key is substitution cipher
  - DES (data encryption standard)
    - 56-bit symmetric key, 64-bit plaintext input, encrypt 3 times with 3 different keys
  - AES advanced
    - Process data in 128-bit blocks, 128-bit or higher keys
- Public key cryptography
  - public key known to all; private decryption key known to receiver
  - plaintext -> K_pub -> ciphertext -> K_prv -> plaintext=K_prv(K_pub(msg))

8.3 Message integrity, authentication
- Authentication
  - I am Alice -> nonce (sender sends nonce, receiver returns K-(R)) + send me Alice's public key K+ -> public key cryptography K+(K-(R))
- Message integrity
  - Digital signature
    - sender signs: encrypt with his private key, Kb-(msg)
  - Message digest
    - produce fixed-size mag
    - digital signature = signed message digest
      - m -> hash(m) -> Kb-(hash(m)) -> Kb+(Kb-(hash(m))) -> hash(m) -> m
      - bob sends digitally signed message
      - Alice verifies signature and integrity of digitally signed message
  - Certification authorities
    - binds public key to particular entity, E (person, router)
    - certificate containing E's public key signed by CA: K_ca-(K_b-)

8.8 Operational security
- Firewall: isolates organization's internal net from larger internet. allowing some packets to pass, blocking others
  - prevent denial of service, illegal access of internal data, authorized access
  - stateless packet filters: filter packet by packet
  - stateful packet filters: track status of every TCP connection, filter packets that make no sense, timeout inactive connections
  - application gateways: require authorized users to telnet through gateway, filter connections not from the gateway