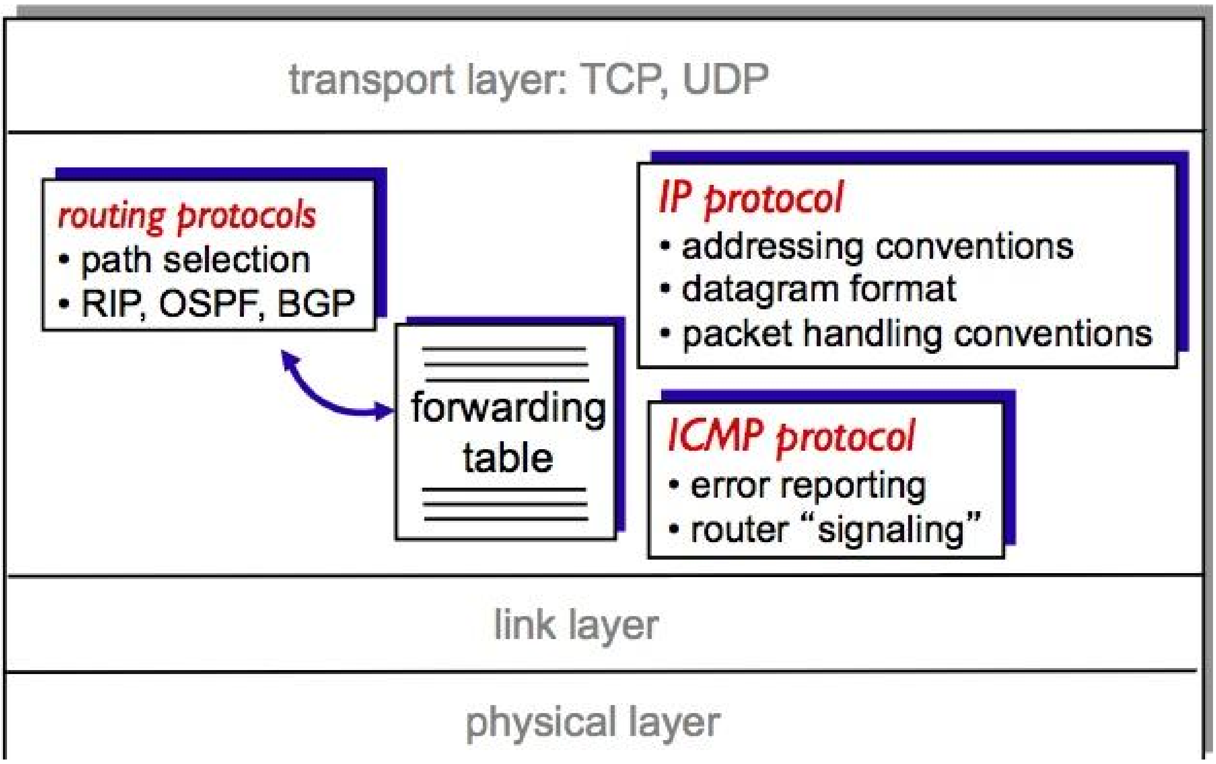
Review is pretty much complete! Please help formatting review (indenting). - Jorden

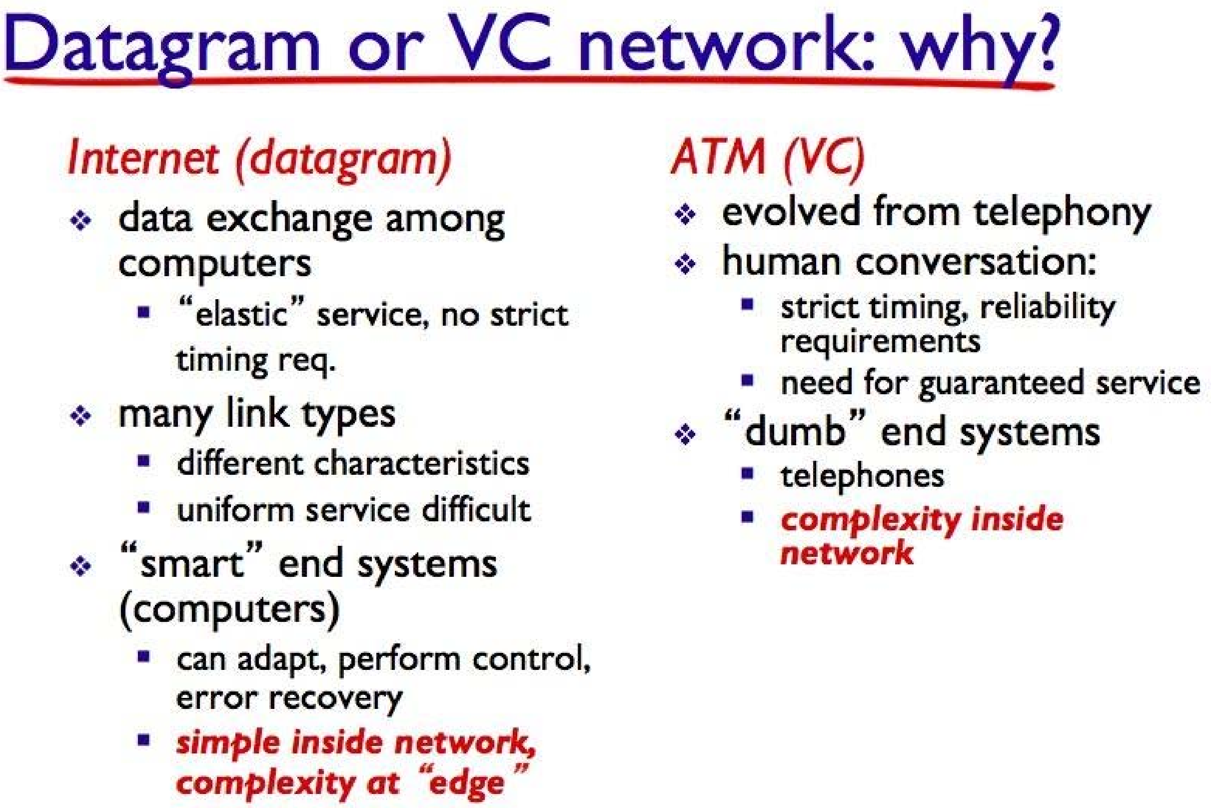
(Ch 4) Network Layer



- **Describe the purpose of the Network layer (service Model) as compared to the Transport layer**
  - **Transport layer: ensures that the protocols operated at this layer provide reliable end-to-end flow and error control (TCP, UDP). ** Between two processes ****
  - **Network layer: controls routing of data from source to destination as well as assembling and dismantling of data packets. ** Between two hosts ****
    - **transport segment from sending to receiving host.**
    - **on sending side: encapsulates segments into datagrams.**
    - **on receiving side: delivers segments to transport layer.**
- **Difference between Forwarding and Routing**
  - **Forwarding: moves packets from routers input to appropriate router output.**
  - **Routing: determine route taken by packets from source to destination.**
- **Virtual Circuit networks (manner of forwarding etc)**
  - **Connection-oriented service**

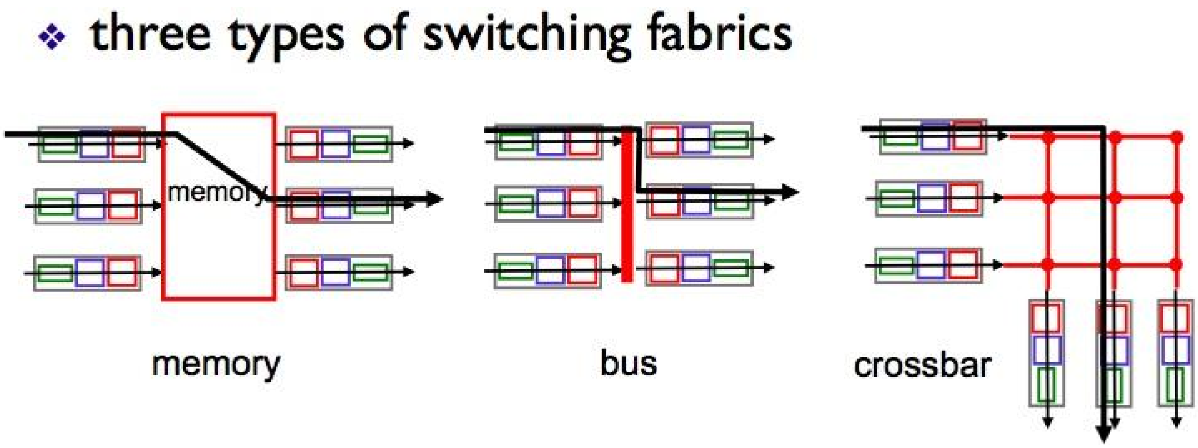
- Each packet carries VC Identifier
- Every router on source-destination path maintains "state" for each passing connection.
- Router resources may be allocated to VC (predictable behavior).
- Contains:
  - path from source-destination.
  - VC numbers, one for each link along path.
  - entries in forwarding tables in routers along path.

- <u>Datagram networks (manner of forwarding etc)</u>
  - Connectionless-oriented service
  - Packets forwarded using destination host address
    - Forwarding table "Destination Address" divided into "ranges"
      - Longest Prefix Matching: use longest address prefix that matches destination address.

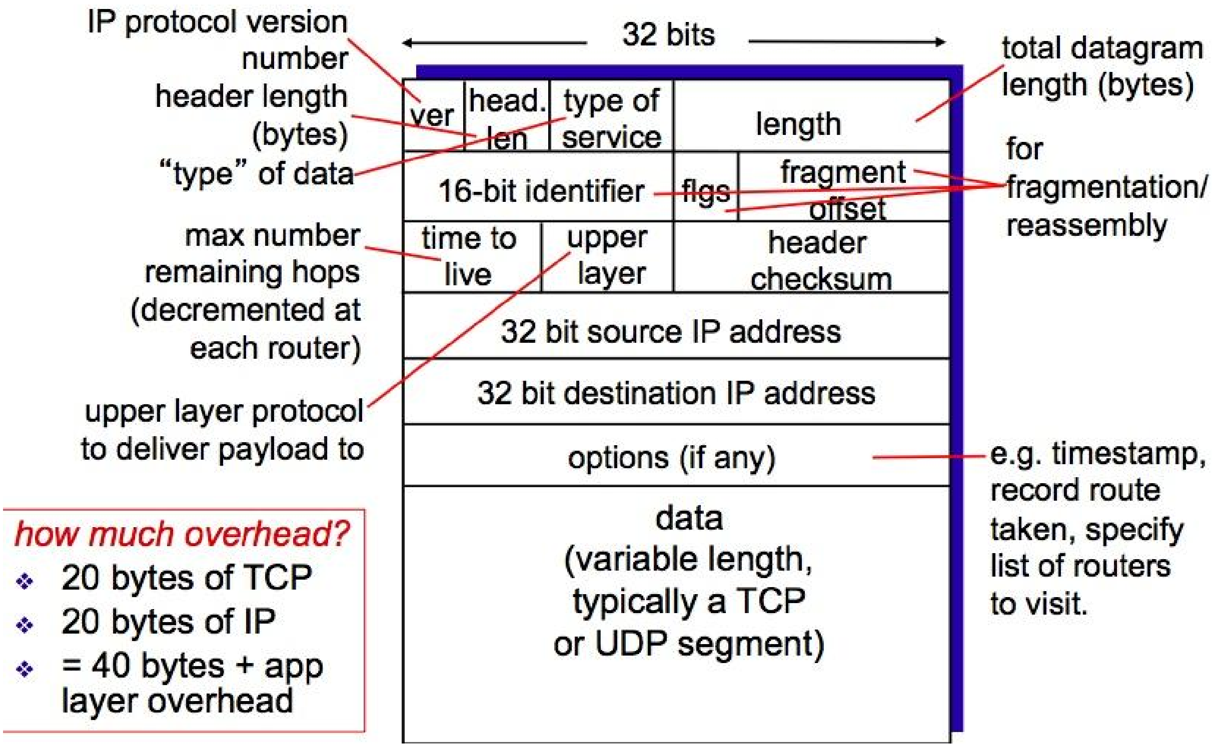


- <u>What's in a Router?</u>
  - run routing algorithms/protocols (RIP, OSPF, BGP).

    - forwarding datagrams from incoming to outgoing links.
    - <u>Input ports</u>
        - given datagram destination lookup output port using forwarding table in input port memory “match + action”.
    - <u>output ports</u>
        -
    - <u>switching fabric</u>: transfer packet from input buffer to appropriate output buffer.



        -
        - Memory: packets copied to systems memory (first generation routers), speed limited to memory bandwidth.
        - Bus: shared bus, speed limited by bus bandwidth.
        - Crossbar: fragmenting datagram into fixed length cells, switch cells through fabric.
    - <u>queuing</u>: if datagrams arrive faster than forwarding rate into switch fabric, queueing occurs.
        - Queuing delay & loss for both input & output buffer overflow.
        - Head of the Line Blocking: queued datagram at front of queue prevents others from moving forward.
    - <u>buffer-size</u>: [(RTT * Link Capacity)/sqrt(N)] where N is the number of flows.
- <u>Datagram format</u>



- **<u>Datagram fragmentation</u>**
    - **network links have MTU (maximum transfer size) thus large datagrams must be "fragmented" in order to be sent and are then reassembled at final location.**
        - **IP Header bits used to determine order of fragments.**
- **<u>Addressing in IPv4</u>**
    - **IP addresses associated with each interface, normally either ethernet or wireless.**
    - **Subnets: device interfaces with same subnet portion of IP address**
        - **devices in same subnet can physically reach each other without intervening router.**
- **<u>Dotted decimal notation</u>**
    - **"xxxx.xxxx.xxxx.xxxx" dots separate the octets of an IP address. IPv4 are 32 bit addresses split up into 4-octetes.**
- **<u>Classless InterDomain Routing (CIDR)</u>**
    - **subnet portion of IP address of arbitrary length.**

- address format: ***a.b.c.d/x*** **where x is the number of bits in the subnet portion of address.**

- **<u>Hierarchical addressing:</u> A common form of location identification that is made up of several levels.**
    - **allows efficient advertisement of routing information.**
    - **Classes *A, B, C, D***
    - **ICANN Internet Corporation for Assigned Names & Numbers responsible for allocating addresses, managing DNS, assigns domain names, and resolves disputes.**
- **<u>Dynamic Host Configuration Protocol (DHCP)</u>**
    - **allow host to dynamically obtain IP address from Network Server when connection to network is made.**
    - **Host broadcasts "DHCP Discover" message**
    -
    - **DHCP Server responds with "DHCP Offer" message**
    - **Hosts request IP address "DHCP request" message**
    - **DHCP Server sends address: 'DHCP ACK" message**
    - **Can return more than just allocated IP address of subnet (address of first-hop router, name/ip of DNS, network mask)**
- **<u>Network Address Translatoin (NAT)</u>:**
    - **local network uses just one IP address as far as outside world is concerned.**
    - **devices inside local network are not explicitly addressable, visible by outside world.**

implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

    - **16-bit port-number field, thus supports up to 60k connections with single LAN-side address.**
    - **NAT is CONTROVERSIAL!**
        - **routers should only support up to layer-3, thus violates end-to-end argument.**
    - **Address shortage should be resolved by IPv6 implementation.**
    - **Traversal Problem: client wants to connect to local NAT address.**
        - **Solution #1: statically configure NAT to forward incoming connection requests at given port to server.**
        - **Solution #2: Universal Plug-n-Play (UPnP) Internet Gateway Device (IGD) Protocol, allows NAT host to:**
            - **learn public IP address.**
            - **add/remove port mappings**
        - **Solution #3: Relaying**
- **<u>Internet Control Message Protocol (ICMP)</u>**
    - **used by hosts & routers to communicate network-level information.**

            - error reporting: unreachable host, network port, protocol.
            - echo request/reply
    - ICMP message = type, code, followed by first 8 bits of datagram causing error.
- <u>IPv6</u>
    - 128-bit address
    - solution to IPv4 addresses being completely allocated.
    - fixed-length 40-byte header
    - NO fragmentation allowed
    - checksum removed to reduce processing time at each hop.
    - Transition from IPv4 to IPv6 accomplished by
        - Tunneling: IPv6 datagram carried as payload in IPv4 datagram among routers.
- <u>Routing Algorithms</u>
    - <u>Link State (be able to do this one)</u>
        - Global information, all routers have complete topology (know location/costs of entire topology).
            - Dijkstra's Algorithm
        - accomplished via link state broadcast.
        - O(n^2)
    - <u>Distance Vector</u>
        - Decentralized information, router knows physically connected neighbors, link costs to neighbors.
        - "Bellman-Ford"

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$

        -
    - <u>Which is better?</u>

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
  - error propagate thru network

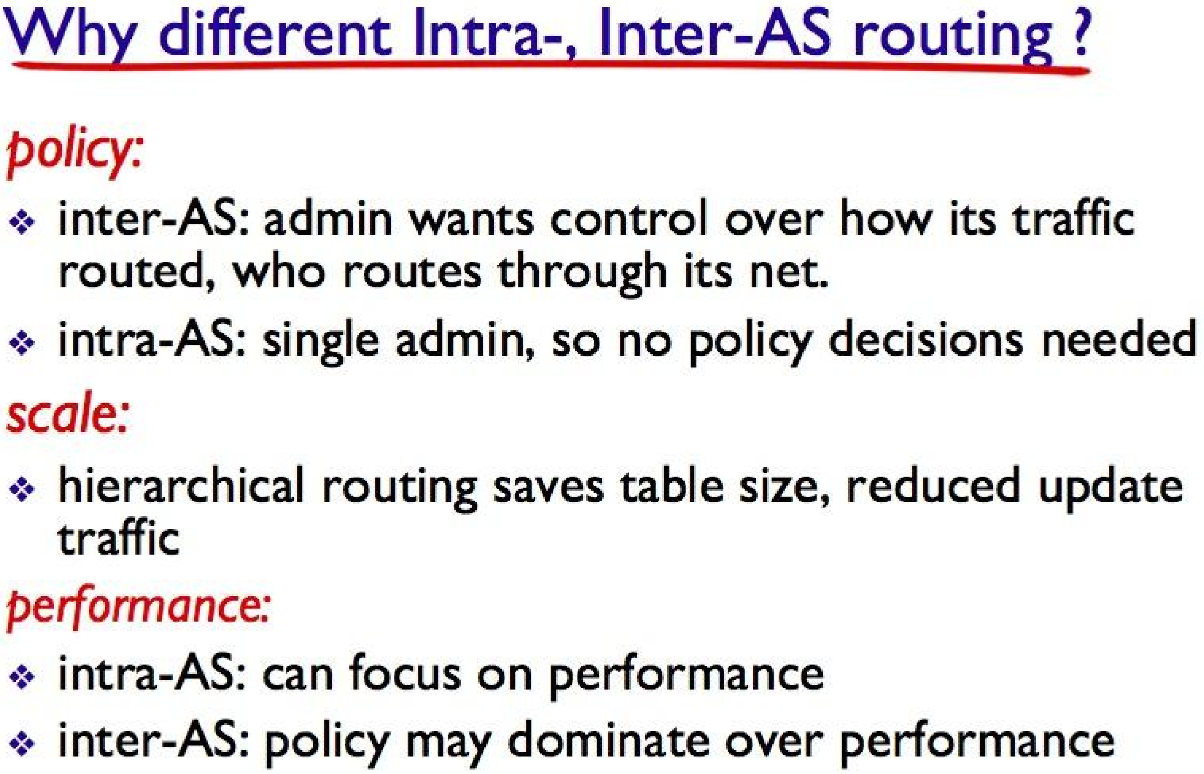
■

- **Hierarchical routing**
  - **can't store all destinations in routing tables!**
  - **routing table exchange would swamp links.**
  - ***administrative autonomy***
    - **each network admin may control routing in their network.**
  - **aggregate routers into regions, "autonomous systems"**
- **Autonomous Systems**
  - **Routers in same AS run same routing protocol.**
  -
    - **"intra-AS routing," sets entries for internal destinations.**
      - **Also known as *Interior Gateway Protocols (IGP).***
    - **"Inter-AS routing," sets entries for external destinations.**

            - Also known as ***Border Gateway Protocols (BGP).***
    - **“Hot Potato Routing” sends packet to closest of two routers.**
    - **<u>Routing Information Protocol (RIP)</u>**
        - **utilizes Distance Vector algorithm.**
        - **Link Failure/Recovery: if no advertisement (response) heard after 180sec. neighbor/link declared dead.**
            - **route invalidated**
            - **neighbors are notified**
            - **neighbors in turn send advertisements if tables have changed.**
            - ***poison reverse* used to prevent ping-pong loops (infinite distance = 16 hops).**
            - **RIP routing tables managed by application-level process called route-d (daemons).**
            -
            -
    - **<u>Open Shortest Path First (OSPF)</u>**
        - **utilizes Link State Algorithm**
        - **route computation using Dijkstra’s algorithm**
        - **advertisement carries one entry per neighbor.**
            - **advertisements flooded to entire AS.**
                - **carried in OSPF messages directly over IP rather than TCP or UDP.**
        - **all OSPF messages authenticated.**
        - **multiple same-cost paths allowed whereas RIP contains a single path.**
        - **Integrated uni- and multicast support.**
        - **Hierarchical OSPF**
            - **Two-Level Hierarchy: local area, backbone**
                - **advertisements only in area.**

            - each node has detailed area topology
        - Area Border Routers: “summarize” distances to nets in own area, advertise to other Area Border Routers.
        - Backbone Routers: run OSPF routing limited to backbone.
        - Boundary Routers: connect to other AS’s.
- <u>Border Gateway Protocol (BGP)</u>
    - the de-facto, inter-domain routing protocol.
    - “glue that holds the internet together”
    - Provides each AS means to:
        - eBGP: obtain subnet reachability information from neighboring AS’s.
        - iBGP: propagate reachability information to all AS-internal routers.
    - ALLOWS SUBNET TO ADVERTISE ITS EXISTENCE TO REST OF INTERNET!
    - BGP Session: two BGP routers exchange BGP messages.
    - Route = prefix + attributes
    - AS-PATH: contains AS’s through which prefix advertisement has passed
    - NEXT-HOP: indicates specific internal-AS router to next-hop AS
    - Gateway router receiving advertisement uses import policy to accept/reject ad.
        - policy-based routing
    - Route selection based on:
        - policy decision
        - shortest AS-PATH
        - closest NEXT-HOP router: hot-potato routing
        - additional criteria
    - BGP messages exchanged between peers over TCP
        - OPEN: opens TCP connection to peer and authenticates sender

- UPDATE: advertises new path.
- KEEPALIVE: keeps connection alive in absence of UPDATE; also ACK’s open request.
- NOTIFICATION: reports errors in previous connection; also closes connection.



- **<u>Broadcast routing</u>: deliver packet from source to all other nodes.**
    - **source duplication is inefficient.**
    - **Flooding: when node receives packet, broadcast packet, send copy to all neighbors.**
    - **Controlled Flooding: node only broadcasts packet if it hasn’t broadcasted before.**
    - **Spanning Tree: no redundant packets received by any node.**
- **<u>Multicast routing</u>**
    - **goal: find a tree (or trees) connecting routers having multicast group members.**
    - **tree: not all paths between routers used.**
    - **shared-tree: same tree used by all group members.**

            - **minimal spanning (Steiner): minimum cost tree connecting all routers with attached group members.**
                - **not used in practice, computationally complex.**
            - **center-based trees: single delivery tree shared by all.**
                - **one router defined as “center"**
        - **source-based: different tree from each sender to receivers.**
            - **shortest path tree**
                - **Dijkstra’s Alg.**
            - **reverse path forwarding**
                - **bad choice with asymmetric links.**
        - **Tunneling: mcast datagram encapsulated within “normal” datagram, similar to IPv6 within IPv4.**
- **<u>(Ch 5) Link Layer</u>**
    - **has responsibility of transferring datagram from one node to physically adjacent nodes via a “link"**
    - **<u>Link Layer services:</u>**
        - **<u>Framing (link access):</u>**
            - **encapsulates datagram into frame, adding header and trailer.**
            - **“MAC” address used in frame headers to identify source/destination.**
        - **<u>Link access MAC protocol</u>: encompassed in “framing"**
        - **<u>Reliable delivery</u>**
            - **seldom used on low bit-error link.**
            - **wireless links have high error rate.**
        - **<u>Flow control</u>: pacing between adjacent sending and receiving nodes.**
        - **<u>Error detection</u>**
            - **errors caused by attenuation, noise.**
            - **receiver detects presence of errors.**
                - **signals sender for retransmission or drops frame.**

        - **<u>Error Correction</u>: receiver identifies and corrects bit-errors without resorting to retransmission.**
        - **<u>Half/Full Duplex</u>**
            - **<u>Half</u>: nodes at both ends of link can transmit but not concurrently.**
            - **<u>Full</u>: nodes at both ends of link can transmit concurrently.**
    - **<u>Link layer implementation</u>**
        - **in every hosts "adaptor" (Network Interface Card) or on a chip.**
            - **Ethernet Card, 802.11 card, Ethernet Chipset**
        - **combination of hardware/software/firmware.**
    - **<u>Checksum</u>: detect errors in transmitted packet, TRANSPORT LAYER ONLY!**

goal: detect "errors" (e.g., flipped bits) in transmitted packet (note: used at transport layer *only*)

sender:

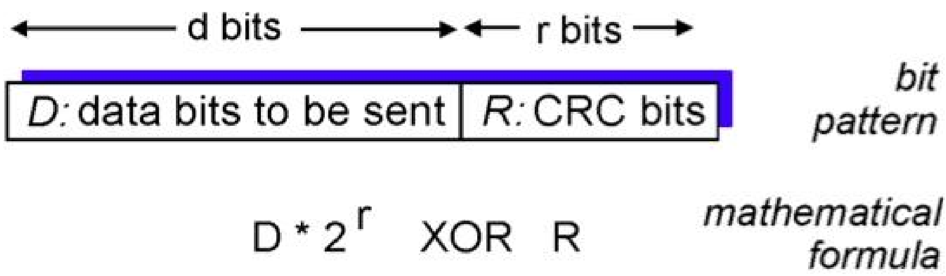
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

receiver:

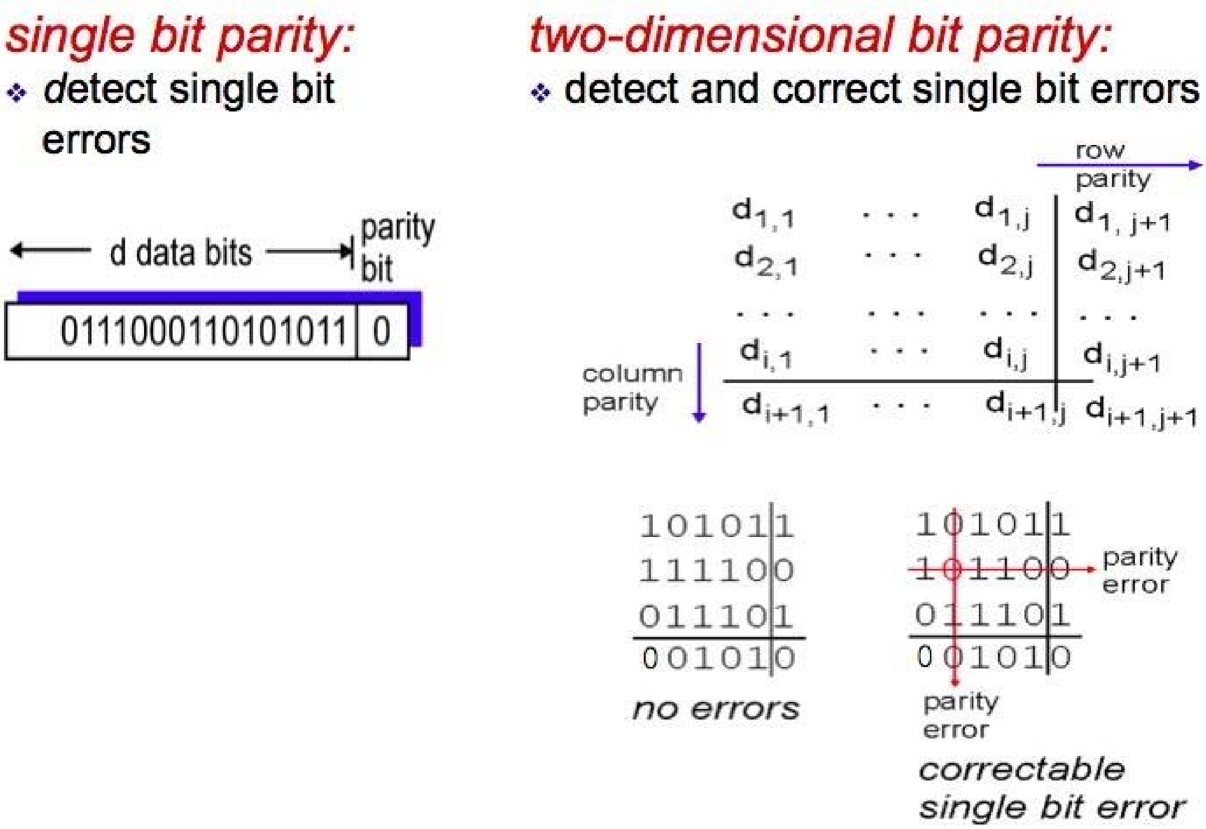
- compute checksum of received segment
- check if computed checksum equals checksum field value:
    - NO - error detected
    - YES - no error detected. *But maybe errors nonetheless?*

        -
    - **<u>Cyclic Redundancy Check (CRC)</u>:**

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)



- **<u>Parity</u>: the use of parity bits to check that data has been transmitted accurately.**



- <u>**Multiple Access Links**</u>
    - **Two types**
        - **Point-to-Poing (PPP for dial-up or point-to-point between Ethernet switch & host).**
        - **Broadcast (shared wire or medium)**
            - **old-fashioned ethernet**
            - **802.11 wireless LAN**
            - **Interference: 2+ simultaneous transmissions by node**
                - **Collision if node receives 2+ signals at the same time.**
- <u>**Multiple Access Protocols:**</u>
    - **distributed algorithm that determines how nodes share a channel (i.e. decide when node can transmit).**
    - **communication about channel sharing must use channel itself.**

- **<u>Channel partitioning</u>: divide channel into 'pieces' allocating 'pieces' to respective nodes for exclusive use.**
-
    - **TDMA: Time Division Multiple Access**
        - **Each station gets fixed length slot in each round.**
    - **FDMA: Frequency Division Multiple Access**
        - **channel spectrum divided into frequency bands, each station assigned a specific frequency band.**
- **<u>Random access</u>: channel not divided, allows collisions and recovers from them.**
-
    - **when node has packet to transfer it transmits at full channel data rate.**
    - **Random Access MAC protocol determines how to handle collision.**
- **<u>Taking turns</u>: nodes take turns but larger datagrams take longer turns.**
    - **Taking Turn Protocols:**
        - **Polling: "master" node invites "slave" nodes to transmit in turns.**
            - **Concerns: polling overhead, latency, single point (master) of failure.**
        - **Token Passing: control "token" passed from one node to another sequentially.**
            - **Concerns: token overhead, latency, single point (token) of failure.**
- **<u>Random Access Protocols</u>**
    - **<u>(Slotted) ALOHA</u>**

assumptions:

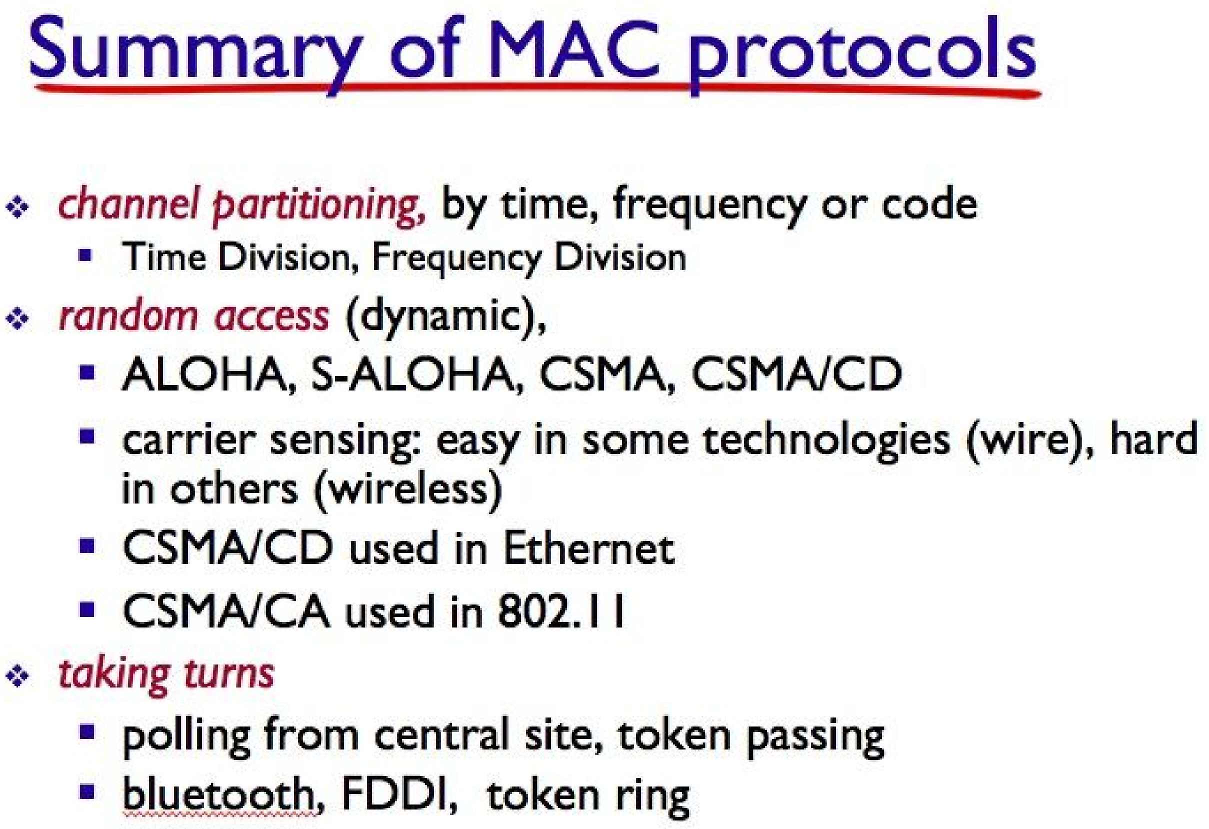
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

operation:

- when node obtains fresh frame, transmits in next slot
  - *if no collision:* node can send new frame in next slot
  - *if collision:* node retransmits frame in each subsequent slot with prob. p until success

  -
  - **At Best: channel used for useful transmissions %37 of the time.**
- **<u>Carrier Sense Multiple Access (CSMA)</u>: "listen before idle"**
  - **if channel sensed is idle, transmit entire frame, otherwise defer frame.**
  - **Human analogy: "don't interrupt others!"**
- **<u>Carrier Sense Multiple Access with Collision Detection (CSMA/CD)</u>**
  - **colliding transmissions aborted reducing channel wastage.**
  - **easy in wired LANs: measure signal strengths, compare transmitted & received signals.**
  - **Human analogy: "the polite conversationlist"**
  - **Algorithm**
    - **Receives datagram from network layer, creates frame.**
    - **If NIC senses channel idle, sends frame otherwise it waits until channel is idle.**

- If NIC transmits entire frame without error then its done.
- If NIC senses another transmission while transmitting, aborts and sends jam signal.
- NIC then enters binary backoff. Longer back off interval with more collisions!



-
- **Link layer addressing:** (MAC or LAN or Physical or Ethernet address)
    - **MAC addresses**
        - used locally to get frame from one interface to another physically connected interface.
        - 48-bit address burned on NIC ROM.
        - Unique for every interface, similar to social security numbers.

        - **<u>Address Resolution Protocol (ARP)</u>: determines interfaces MAC address with known IP address.**
            - **ARP Table: each IP node has table containing: IP/MAC address mappings && TTL.**
            - **If A wants to send B a datagram and B is not in A's ARP table then A broadcasts an ARP query packet containing B's IP address.**
    - **<u>Ethernet:</u> dominant wired LAN technology.**
        - **Bus: popular through 90's, all nodes in same collision domain.**
        - **Star: prevails today, active switch in center; each spoke runs separate Ethernet protocol, no collision.**
        - **<u>Frame</u>: sending adapter encapsulates IP datagram in Ethernet Frame.**

        -
        - **Preamble: 7 bytes with pattern "10101010" followed by one byte with pattern "10101011" used to synchronize sender/receiver clock rates.**
        - **Address: 6-byte source/destination MAC addresses**
            - **if adapter receives frame with matching destination MAC address or broadcast address it passes data in frame to network layer protocol (ARP).**
        - **Type: indicates higher layer protocol**
        - **Cyclic redundancy check (CRC): at receiver; error detected? Then frame is dropped.**
        - **CONNECTIONLESS**
        - **UNRELIABLE: No ACKS or NACKS between NICs.**
    - **<u>Switches</u>:**
        - **store/forwards ethernet frames.**

- examine incoming frames MAC address and selectively forward frame to one or more outgoing links.
- Transparent: hosts are unaware of presence of switches.
- Plug-n-Play: switches do not need to be configured.
- Hosts have dedicated-direct connection to switch.
- switches buffer packets.
- Every switch has a switch-table containing routing table.
- <u>Forwarding & Filtering:</u>

Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   then {
   if destination on segment from which frame arrived
      then drop frame
      else forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving interface */

  -

- <u>Self-Learning</u>: switches update switch table with sender/location every incoming frame.
- <u>PPP</u>: type of Point-to-Point access link
  - for dial-up access.

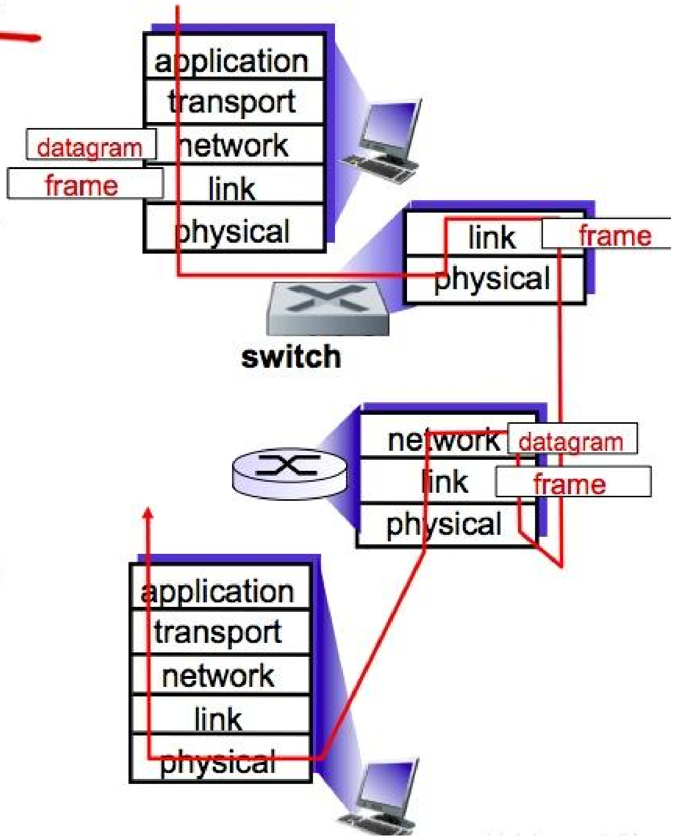
Switches vs. routers

both are store-and-forward:

- *routers:* network-layer devices (examine network-layer headers)
- *switches*: link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses



○