

## → Network Topology

L2

- \* ~~Wired~~ Fully mesh: each device has a dedicated point-point link to every other device.



- \*  $n$  computers, each should connect to  $n-1$  computers.  
 total links:  $n(n-1)$ , ~~with  $\frac{n(n-1)}{2}$~~  bidirectional :  $\frac{n(n-1)}{2}$   
Bad scalability since no. of links increases quadratically  
 w.r.t. no. of comps.

\* Bus

- \* runs a Central Cable as a backbone and all computers are connected to this backbone.
  - \* only a single device on the network can transmit at any point in time. Requires negotiation mechanism to resolve conflicts.
  - \* pros: easy to remove/add devices
  - \* cons: single point of failure (cable)



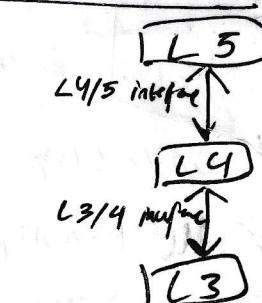
\$ Star

- \* All devices attached to a central device (hub)
  - \* hub manages the return
  - \* Data from clients goes to hub first, then it goes to dest.



\* Ring

- \* Ring
    - + Each device passes on data to be ~~sent to~~ next device until it reaches dest.
    - + Transmission dir. set clockwise or anti-clockwise
    - \* requires access control to resolve prop. queing
    - \* bottleneck : device is slower and process info slower relative to others
  - Network Software : protocol Hierarchies (1)
  - \* Network : stack of layers
  - + each layer offers services to layers above via Interfaces
  - \* protocol : agreement b/w communicating parties on how 'communication is to proceed.'



- \* protocol : set of rules that define details on how to provide service
- \* how to add and where's inside header
- \* services are abstract, ~~not~~ implemented using primitives
- \* Service : set of primitives that a layer provides to a layer above it through interfaces b/w layers

\* analogy : services = interfaces, protocols = class implementation of interfaces

→ OSI (Open Systems Interconnection) (ref. model of a network)

\* 7 layers, each performing a defined fn. chosen to minimize info flow across interfaces

\* physical layer (L1) (bit)

\* sends raw data (0s and 1s) to the other side

\* Both sides agree to a transmission rate

\* transmission medium

\* Data Link Layer (L2) (frame)

\* error detection/correction and flow control

\* provide reliable and efficient service for upper layer

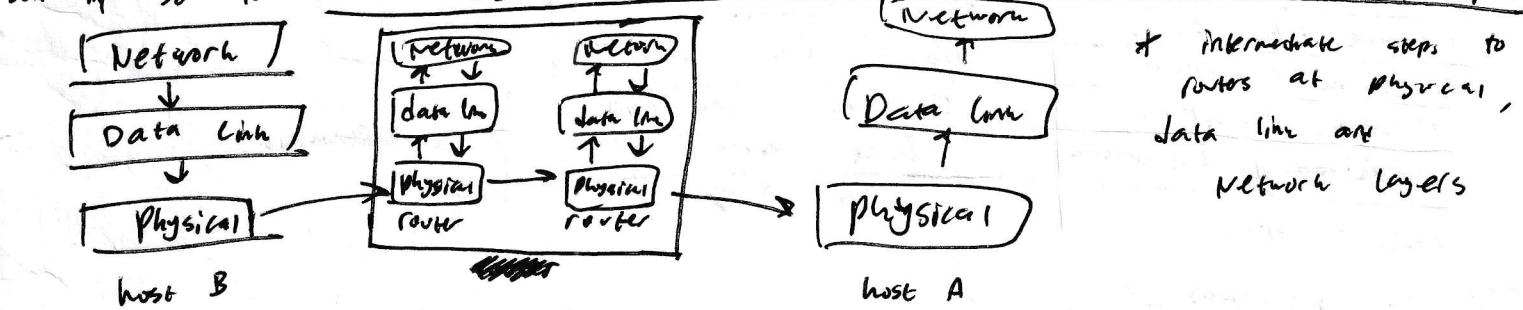
\* can ask for retransmissions

\* Network layer (L3) (packet)

\* flow packets are sent from source to dst. across different networks

\* handle intermediate steps b/w sender and receiver

\* info go to network layer and check IP and determines best router on path



+ intermediate steps to routers at physical, data link and network layers

\* Transport layer (L4) (segment)

\* doesn't care about intermediate steps

\* how to send messages from source to dest.

\* conduct segmentation and re-assembly them

\* Transport and Network could split messages depending on protocol.

\* Session layer (L5) (SPDU)

\* control session, flash management (restart from certain point), e.g. for video and audio of data

\* Presentation layer (L6)

\* data representation and compression fn. (ensure data from app can be accepted)

\* Application layer (L7)

\* web browser, email, etc. + users can be humans or other software

→ TCP / IP (Transmission Control Protocol / Internet Protocol)

\* 4 layers

Application	HTTP, SMTP, P2P, DNS
Transport	TCP, UDP
Network/Internet	IP, ICMP
Link	DSL, SONET, 802.11, Ethernet

## → Architecture of Internet

- \* Users connect to local area network or ISP network
- \* Service networks are connected to exchange points
- \* Backbone network supports high speed up to low distance transmission

## → Physical layer (data transmission over physical medium)

- \* Core abt bit level transmission
- \* 3 aspects: Mechanical, Electrical and Timing
- \* Both sides must agree to data rate and be in sync. → timing
- \* Mechanical: material, cable length (attenuation)
- \* Electrical: voltage levels, signal strength

### \* Bandwidth: rate of transmission in bits/second

(how much data can be sent/received per sec)

- \* Delay / Latency: time required for 1st bit to travel from Comp A to Comp B.
- \* Message latency: associated w/ sending a message over a link
- \* Transmission delay:  $T\text{-delay} = \frac{\text{Message length in bits}}{\text{Rate of transmission (bps)}}$
- \* Time needed to put message on the link
- \* Propagation delay:  $P\text{-delay} = \frac{\text{length of channel}}{\text{speed of signals } (\frac{2}{3}c \text{ for wire})}$
- \* Time needed to send message

$$\boxed{\text{Latency} = T\text{-delay} + P\text{-delay}}$$

$$\boxed{\text{end-end transit time} = \frac{2 \times \text{distance}}{c}}$$

## → Transmission media

- \* Wired: twisted pair, coaxial, fiber optics
- \* Wireless: EM waves, air, satellites
- \* Performance for physical media affected by physical properties
- \* Signal attenuation: loss in amplitude of a signal as it passes through a medium
- \* Impacts how far and how much data a medium can carry.

\* Twisted pair: 2 insulated copper wires twisted in helical form.

\* twisting reduces interference; cancelling out EM interference from external sources

\* used by telephone line

\* Bandwidth depends on distance, wire quality, density

Cat 8 > Cat 5 > Cat 3

## \* Coaxial Cable

\* Copper wire w/ insulation, mesh, sheath

\* better shielding w/ twisted pair = higher speeds over greater distances

\* Bandwidth approaches 1 GHz, used for TV/internet

## \* Fibre optics

\* High bandwidth and tiny signal loss

\* fibre of glass

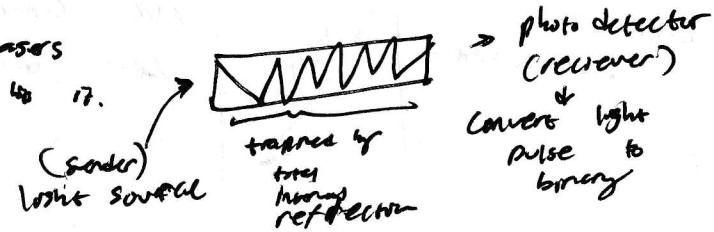
\* high rates and long distances; backbone lines b/w ISP facilities

\* light source, transmission medium, detector

\* emitting using LEDs or semiconductor lasers

\* receiving using electronic pulses w/ light

\* detector generates electrical pulses w/ light



\* Modulation : Process of sending 0s and 1s (bits) using signals

• Baseband Transmission set  
using signals from 0 to max freq. f

• Passband Transmission  
signals are shifted to occupy a wider range of freq.

• NRZ Signal of bits } Baseband  
Amplitude shift keying } Passband } Modulation methods  
Freq. ————  
Phase ————

• Sampling Theorem: If the signal has highest freq at  $f_H$ , then lowest sampling rate should be  $2f_H$ .

• Modulation is changing amplitude, freq, phase to represent  $\log_2 2 = 1$  bit  
symbol 2 or 0s.  
• How many bits can I send based on N types / freq. symbols  $\log_2 4 = 2$  bits  
 $\{ 0 \text{ and } 1 \text{ or } 00, 01, 10, 11 \} = \log_2 N$

• Symbol Rate

Symbol = ~~one~~ piece of signal or signal element (represent combination of bits)

↑ symbol can represent multiple bits (~~multiple~~) (data elements)

Data Rate =  $\log_2 N \times \text{Symbol rate}$  → amount of data transmitted per unit of time across a medium!!!

Symbol rate = No. Signal elements we can send per second (Band rate)

• Max Data Rate of a channel

Nyquist Theorem: Max data rate =  $2B \log_2 V$  bits/sec

Max data rate for noiseless channel,  $B$  = bandwidth,  $V$  = no. of signal levels (cliques, tones, etc.)

Shannon Theorem: Max data rate =  $B \log_2 (1 + S/N)$  bits/sec

↓  
How fast signal can change      ↓  
How many levels can be seen  
Max data rate for noisy channel,  $B$  = bandwidth,  $S$  = signal strength,  $N$  = noise

• High SNR: signal is understandable, low SNR = not understandable.  
use both theorems to find the lower value which is the bottleneck (max channel cap.)

• if Nyquist is bottleneck, increase signal levels

• Channel sharing

\* Users can transmit at the same time (upload as download)

\* Full duplex: transmission in both directions

\* Half duplex: both directions, but not at the same time

\* Simplex: Only one fixed direction at all times

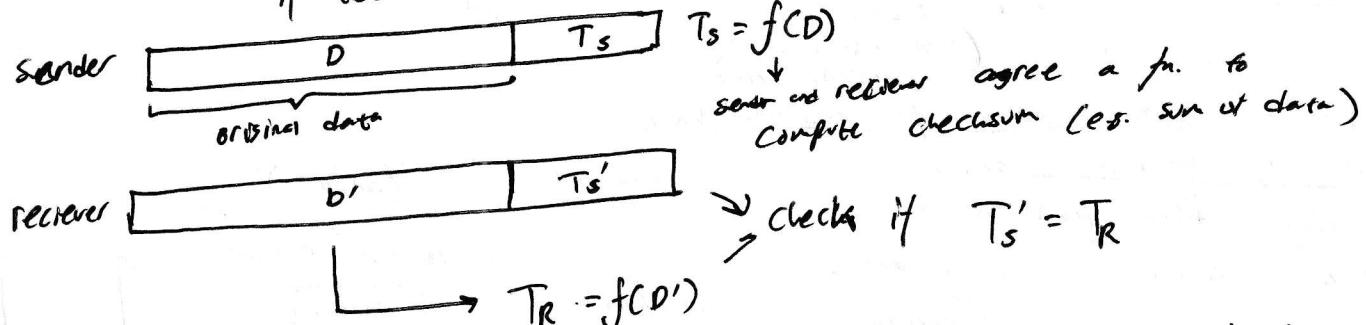
\* Multiplexing (to increase cap. of channel)

• When multiple users want to access the medium

• Time division: Users can use according to a fixed schedule (slotted access to be full speed of channel)

• Freq. division: Users can only use specific freq. to send their data (continuous access of lower speed)

- \* Acknowledged Connection-oriented
  - \* Source transmits independent frames to recipient host.
  - \* Frames are numbered, counted, acknowledged w/ logical order enforced
  - \* Applications: unreliable links such as satellite channel
- Framing
- \* Frame groups raw bits into discrete units (group of bits)
  - \* Framing: provide reliability over unreliable physical layer
  - \* e.g. checksums: extra bits added to the end of data to help the other side if there are errors

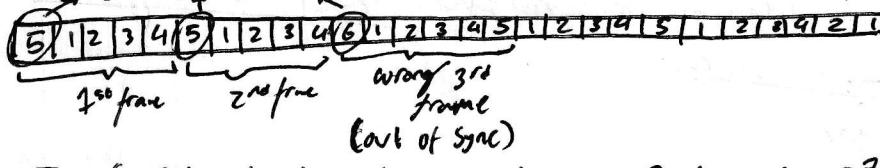


This is part of the trailer.

- \* Let the other side know the start and end of a frame for error handling:
  - \* Char. (Byte) Count
  - \* Flag bytes w/ byte stuffing
  - \* Start and end flags w/ bit stuffing
- } helping receiving side to start and end of a frame

#### \* Character Count

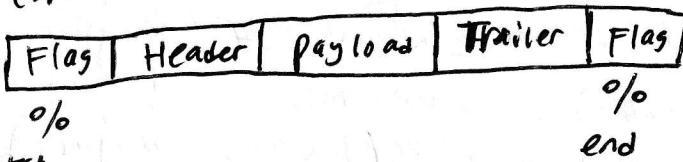
- \* Uses a field in the frame header to specify the length (no. of char) in the whole frame (header incl.)
- ↳ character count → tells you where the start of frame is.



- \* To convert to binary: 1 byte = 8 bits:  $\begin{array}{|c|c|c|c|c|c|c|c|} \hline 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}$  for 5
- \* use over method to find ending of frame

#### \* Flag Bytes w/ Byte Stuffing

- \* each frame starts and ends w/ a special byte - "flag byte"
- (special character: "\0") e.g.



what if \0 is in payload?  
define "\\" char after \0  
to avoid this: "\ \" \0" → ~~byte~~ stuffing

- \* for each ESC "\\" char, it may encloses the next byte. we need to use esc or esc symbols in payload as well.

} make sure to avoid byte stuffing to avoid wasting bandwidth



## → Parity Bit (Detection)

- add 1 bit redundancy ~~to make sure total no. of 1s is even or odd~~  
e.g. given 1000 1110, count no. of 1s  
Sender: add parity bit → 1000 1110 0 (for Even parity) → already even no. of 1s  
~~000~~ 1000 1110 1 (for odd parity) → need to add 1 since here are even 1s and need odd

receiver: Count no. of ~~1s~~ 1s

- Hamming dist = 2, detection =  $2-1 = 1$  bit error

$$\text{Correction} = \frac{2-1}{2} = 0.5 \text{ bit error}$$

Cannot locate any error, so can't correct errors.

## → Internet Checksum (16-bit word) (Detection)

- sum modulo  $2^{16}$  and add any overflow of high order bits have to low-order bits  
e.g. 7, 11, 12, 0, 6, -36 ← make the sum negative so the receiver can just add everything and see if it adds up to 0.

- Add all of the binary code words

Check if there is bit overflow → resolve by putting the 1 to the end  
↓  
find one's-complement (flipping 0 to 1 and 1 to 0)

- Receiver repeats this process w/ the checksum to see if it results in all 1s or all 0s. If yes, no errors.

Hamming dist = 2

- for internet, each word is 16-bits.

- ~~fail~~ fail if bits are flipped w/o changing the sum.

## → Cyclic Redundancy Check (Detection)

- Based on division. If there are any random errors, remainder should be different.

- long division based on generator polynomial  $G(x)$

1.) get divisor from  $G(x)$ . Divisor is the coeff of the polynomial.

(e.g.  $x^4 + x + 1 = G(x)$ , divisor = 10011) (use all terms  $x^4, x^3, x^2, x^1, x^0$ )

2.) get dividend from augmented data which is data to be sent plus 0s at the end, and the no. of 0s is equal to ~~the~~ the highest power of  $G(x)$ .

Exclusive OR (XOR) | 3.) Do long division using XOR, any 0s at the front can be crossed out and replaced w/ the same no. at the next ~~the~~ bits in the augmented data.

4.) After everything, get final 4 bits in result and subtract it from augmented data (use XOR) and that is the data ~~sent~~ sent to the receiver after CRC.

5.) Receiver does long division on the sent data w/ the same divisor  $G(x)$  and should set all 0s since remainder = 0.

32 bits

Hamming distance = 4

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

## Noisy Channel Protocol

\* frames can be lost. What if ACK is lost? Sender will be blocked.

- \* Timeout fn. to determine arrival or non-arrival of complete frames.  
To avoid being blocked by missing ACK.

\* Reject duplicate frame? Requires seq. no. to avoid accepting the frame again

\* Stop and Wait protocol

- ABR (Automatic Repeat Request)

ACK, timeout, and seq. no.

- and of Sec. no. 13 0 and 1,

- $\text{frame } 0 = \text{prev frame}, \text{ frame } 1 = \text{curr. frame}$  (if buffer = 1)

- expect var to see if it is duplicate or not.

- timeout for B set by us. offshoot fn. used to wait ~~the~~ until RESEND.

\* Link Utilisation (LU) in Stop and Wait protocols

- measures efficiency of comm.

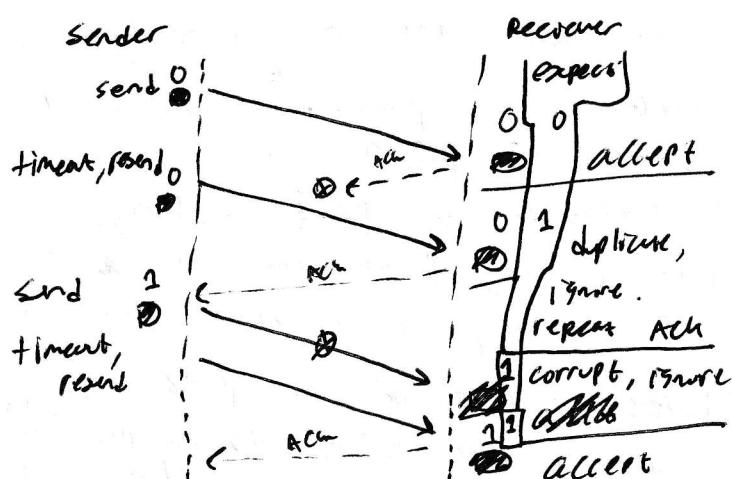
$T_f$  = transmission delay, time needed to transmit frame of length L

$T_p$  = propagation delay, waste time

$T_a = 0$  (time for transmitting Ack)

$$T_t = T_f + Z T_p$$

$$U = \frac{T_f}{T_b}$$



$$U = \frac{L}{L + 2T_p B}, \quad L = \text{frame length}, \quad B = \text{Band width}$$

## \* Sliding Window Protocols

- \* Sliding Window Protocols
  - increase  $CW$  to decrease waiting time (variance)

- How buffer is both sides :

- Have buffer in both sides.
- Sliding window: Server maintains a set of sequence no. corresponding to frames it would like to send

- Recovery window: Receiver  $R_n$  to frames allowed to accept

- to frames allowed to wait to keep scaling and increase proportion of time used for transmission, improve LUB

- Tradeoff: buffer frames before processing

## → Data Link Protocols

\* Point-to-Point Protocol (PPP)

\* Delivery packets across links

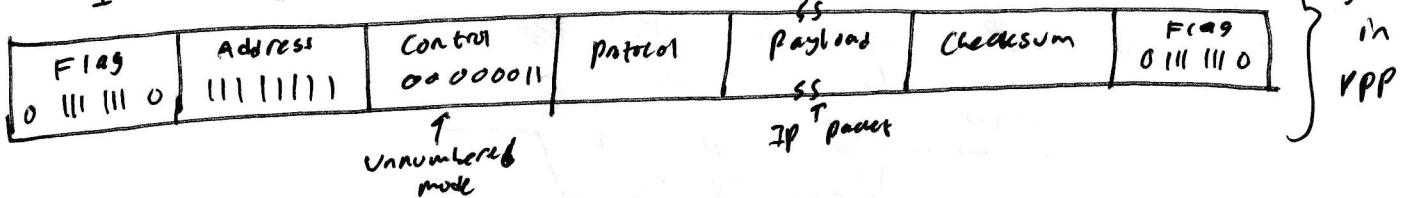
\* Framing: uses Flag byte w/ byte stuffing (0 111 110)

\* Payload contains IP packet from ~~higher~~ network layer.

\* Default in unnumbered mode: connectionless and unacknowledged service.

\* Error detection using Checksum.

Bytes      1      1      1      4 or 2 var      2 or 4      1

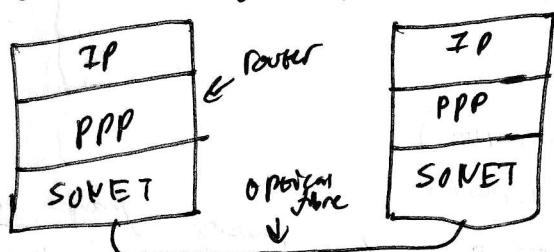


\* Packet over SONET (Synchronous Optical Network)

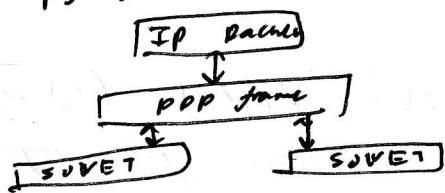
\* Carry IP packets over SONET optical fibre links

\* Use PPP for framing

\* SONET is fibre optic protocol in Physical layer, IP is protocol in Network layer



\* SONET may split PPP frame into fixed length payload that is defined



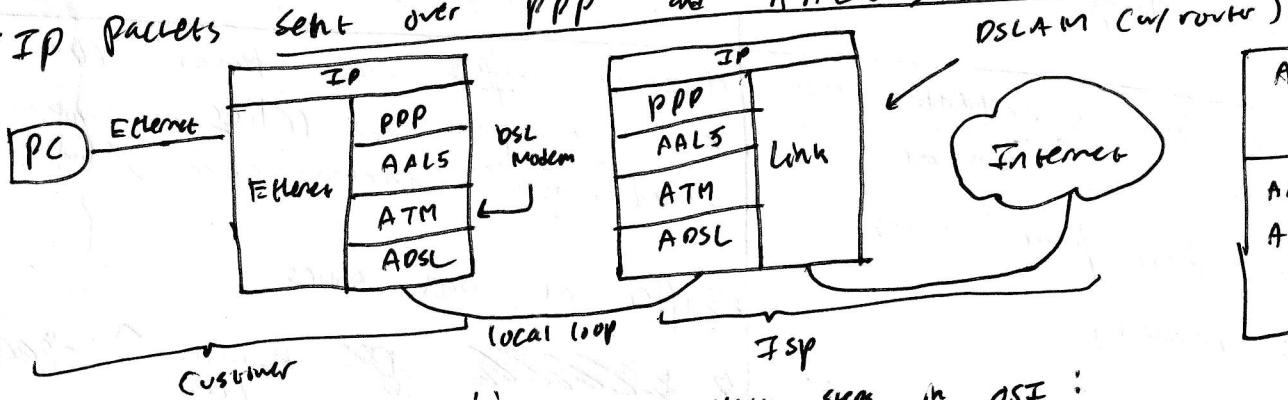
\* frames are split and sent to the other site in regular intervals.

\* Packet over ADSL

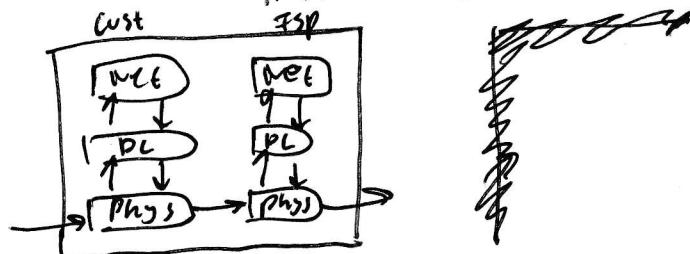
\* Used for broadband Internet over telephone lines

\* Runs from modem (Customer) to DSLAM (ISP)

\* IP packets sent over PPP and AAL5, ATM



Arrives to intermediate steps in OSI:



1.) Channel is ~~not~~ Shared; users compete for resources

2.) ~~stations~~ generate new frames at a certain rate independent of others.

3.) Simultaneous transmission results in damaged frames (collisions)

4.) Time

- Continuous transmission can begin at any time
- Slotted transmission, where timeline is divided into discrete slots. Stations can only send at slots.

5.) Carrier Sense

- Carrier sense: Detection of channel use prior to transmission (to check for potential collisions)
- No carrier sense: No detection

→ MAC Protocols

- ★ Contention: ALOHA, Carrier sense MA (Compete ~~for~~ for channel) (Random access: contention)
- ★ Collision Free
- ★ Limited Contention
- ★ MACA / MACAW (for wireless LANs)

→ ALOHA (Contention)

- ★ User transmits frames whenever they have ~~free~~ (no carrier sense)
- ★ If there are collisions, retry after a random time (or no ACK arrived)
- ★ ~~No central control mechanism required~~
- ★ Efficient under low load, high load is inefficient due to high traffic  $\rightarrow$  high collisions

→ Slotted ALOHA (Contention)

(frames wait for the next slot to send)

- ★ Allow users to start sending only at the start of defined slot.
- ★ Increase efficiency  $\rightarrow$  reducing ~~collisions~~ collisions
- ★ Limitation: have a synchronized clock that every frame must follow!!!

→ Carrier Sense Multiple Access (CSMA) (Contention)

- ★ Before transmission, receive all senders to check channel to detect active transmission. Postpone transmission if it is used.

★ Determine transmission rights dynamically (persistent: Continue to try listening)

- ★ Protocols:
  - persistent and non-persistent CSMA
  - CSMA w/ collision detection

→ Persistent and Non-Persistent CSMA

- ★ 1-persistent CSMA: continuous carrier: Transmit 1 frame and check collisions. If yes, wait a random time and repeat.

- ★ Non-persistent CSMA: If channel busy, wait random period and check again; if idle, start transmitting. If collision, wait random period and repeat.

- ★ p-persistent CSMA: If channel idle, transmit w/ prob.  $p$  or defer to next slot w/ prob.  $(1-p)$  and check again. If collision, wait for random time and repeat.

→ Contention vs. Collision Free

\* low loads : Collision free is less attractive due to

Overhead

\* higher loads ; Contention method is less attractive due to higher no. of collisions.

→ Contended Connection Protocols

\* Contention + Collision Free

\* divide stations into groups, within which only a very small no. of stations are likely to transmit data

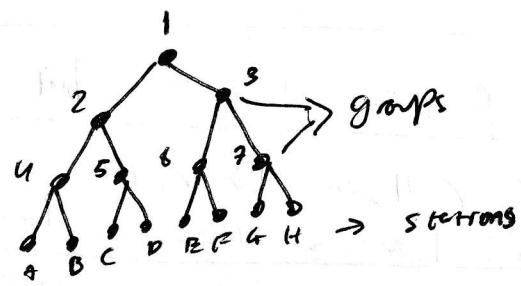
\* Avoid waste due to idle periods and collisions

\* Adaptive Tree Walk Protocol (LCP)

\* organize stations in a tree and use tree to divide stations into groups.

\* All stations can transmit.

If collision occurs, binary division used to resolve contention.



+ DFS under rules w/ poll levels if > 1 station can transmit.

+ start search at lower levels if only 1 station in a group was

+ go down the tree and if only 2 stations in a group were found. (left to right)

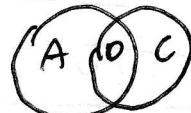
+ go down the tree and if any 2 stations, use contention to send, then station will send first.

+ If it goes to final group w/ only 1 station, use collision free alg.

\* Dynamic

\* → Wireless LAN Protocols

\* Stations have coverage regions in wireless.



(can only hear or transmit in a certain range) of multple stations.

\* Consider wireless coverage range of multiple stations.

Interference affects signal reception.

\* Stations must detect transmission around receiver, not just carrier sensing.

## → MAC Address

- \* physical address for hardware (Physical Interface)
- \* 48-bit no. encoded in frame, written in hexadecimal notation  
 ↓  
 16 symbols  
 $(1 \rightarrow 9)$   
 $(A \rightarrow F)$   
 (6 bytes)

## → Network Layer

- \* Connecting different networks (interNetworking), routing
- \* Internet is a network of networks inter connected by IP
- \* IP provides best-effort service to route datagrams from source host to destination host
- \* Host may be on different networks
- \* Routy : Take the shortest path w/ shortest hops b/w source as dest.

## → Store-and-Forward Packet Switching

- \* Hosts generate packets and inject to network
- \* Router routes packets through network
- \* treat packets as messages, receive/store them as per forward provided to Transport layer
- \* Services independent of Router technologies

## → Service

- \* Network address to send packet (IP address)
- \* Types of services:

- \* Connectionless: Packets are injected into subnet individually and routed independently to dest. (Internet) (this is hard)
- \* Connection-oriented: Packets travelling to dest. follow same route (telecomm.) (this is easy)

- \* Connectionless - post office model (Datagram subnet)
- \* Routing table is dynamic
- \* routed individually based on dest. addresses

## → Forward

- \* Receive - store - process - forward

- \* Connection-oriented - tel. network model (Virtual Circuit Subnet)

- \* define connection id where packets ~~use~~ virtual circuits are based on.

- \* Send all packets using connection id.

- \* To update, have to set up another network.

- \* Virtual circuit + ~~use~~ can be whatever transmission media

## → Internetworking

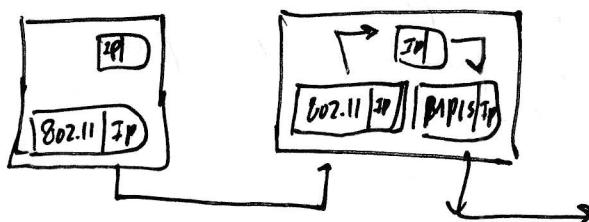
- \* joins multiple, heterogeneous networks into a single larger network

\* Issues when connecting networks:

- 1.) Different network types and protocols
- 2.) Different motivations for network choices
- 3.) Tech at both hardware and software levels

} How to deal w/ different network requirements.

→



Source packets are encapsulated in packets travelling through connecting network.

\* Tunneling: When source and dest.

are on the same network b-t

But there is another network in b/w (e.g. network is IPv6 but a subnet is IPv4)

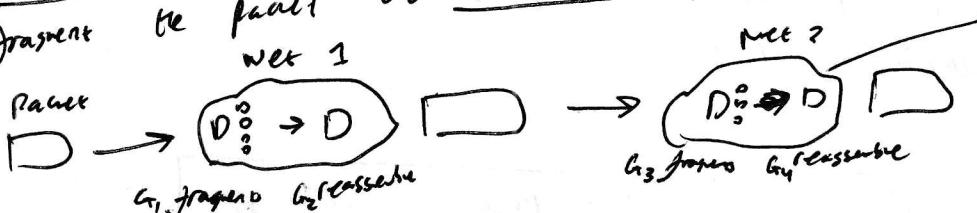


Source and dest. Must be on the same network for tunneling to work!!!. (both must be IPv4/IPv6)

## → Fragmentation

- \* All networks have max size for packets, ~~max~~ transmission unit (MTU) to improve transmission efficiency and confidence (don't want to block channel w/ large packets)
- \* Large packet size decreases overhead since division into smaller sizes would result in many smaller packets having to get headers.
- \* If packet is over limit, fragmentation divides packet into smaller packets and sets individually and independently to the channel.

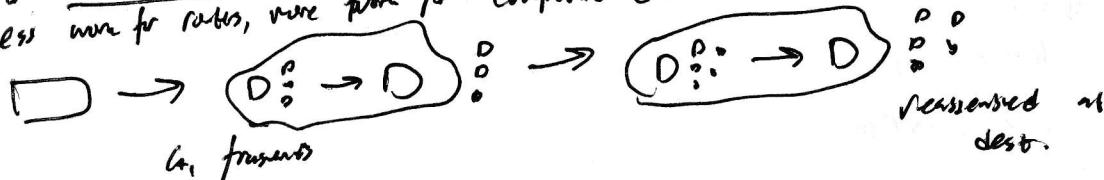
\* Trans parent  
• invisible to the next network  
• fragment the packet and reassemble before exiting network



fragment based -  
network own  
requirements. Net 2  
doesn't know it was  
been split by.

## \* Non Transparent Fragmentation

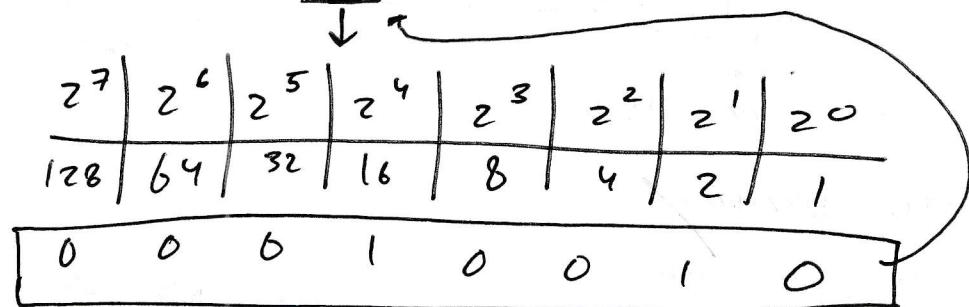
- \* reassembled at destination
- \* less work for routers, more work for computers (used in internet)



→ IPv4 addresses

- IPv4 addresses
  - Subnetting: Divide a block of addresses in IPv4
  - IPv4 is 32-bit long, in dotted decimal notation -
    - 4 segments, each w/ 8 bits.

18. 128. 18. 3. 11



Case: O-255



Address allocation  
an host gets an IP address first gets it. (play)

- \* First come first serve, var and .
  - \* Allocate based on group. Split address based on group.
  - \* And group allocates each address in order to peer users.

first address in a network should have all 0's,  
last ————— 1's )  
128.18.3.255

22. 128. 18. 3. 0

~~100000000~~ 00010010 00000011 00000000  
11111111

es. 120.18.3. 0127

$$\text{host portion: } \begin{array}{c|ccccc} 0 & 0 & 0 & | & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & | & 1 & 1 & 1 & 1 \end{array} \quad \begin{array}{l} \text{first} \\ \text{last} \end{array} \quad \begin{array}{l} 0 \\ 31 \end{array} \quad \begin{array}{l} (2^0 - 1) \\ (2^5 - 1) \end{array}$$

\* Private IP Addresses that cannot appear on Internet  
(reserved for private networks)

\* 10.0.0.0/8 ( $2^{32} - 2^8 = 2^{24}$  hosts)

172.16.0.0/12 ( $2^{20}$  hosts)

192.168.0.0/16 ( $2^{16}$  hosts)

\* Network Address Translation (NAT)

\* Box maps one external IP address to many internal IP addresses. Allows multiple devices on a private local network to share a single public IP address when accessing internet by translating private IP address to public.

→ IPv6

\* Larger address space: 128-bit address

\* Use hexadecimal Colon notation

\* Support more security, encryption and authentication.

\* 40 bytes IPv6 header

→ Internet Control Protocols

\* IP works w/ help from other protocols

\* ICMP (Internet Control message Protocol):  
returns error info (e.g. traceroute, ping)

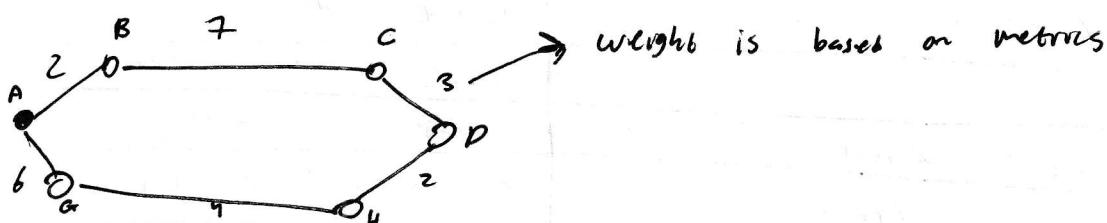
\* ARP (Address Resolution protocol):  
finds MAC address of local IP address  
(host queries on address and owner replies)

\* DHCP (Dynamic Host Control Protocol):  
assigns a local IP address to a host  
(host starts by automatically config it.)  
(host sends req. to server, which grants a lease)

## → Shortest Path Routing (Non-adaptive)

- \* finds shortest path b/w 2 routers on the graph.
- \* Each node represent a router and each arc represent Comm. link.
- \* Metrics: no. of hops, distance, delay, etc.

e.g.



- \* Dijkstra algorithm.
- \* identify shortest-path tree for a dest. root node.
- \* Assume weights are non-negative. Curshes of 1 gives path w/ <sup>fewer</sup> <sub>hops</sub>
- \* shortest path is one of lowest total weight
- \* algorithm:

• Set  $P$  = set of all nodes added to free, initialise empty and root node. Iteratively add 1 node which has shortest distance to root node.

• start w/ table. to lns as all to nodes =  $\infty$ .

• put weights of neighbors for every iteration  
direct

• Select the node with shortest distance (lowest weight)

• put node in  $P$  and clear neighbors of the selected node, repeat.

• Note that this is the distance from curr. node to first node NOT curr. node to neighbor.

## ⊕ Flooding algorithm

- \* Every incoming packet is sent out on every outgoing line except the one on which it arrived.

\* Inefficiencies: generates lots of duplicates; uses lots of bandwidth

\* ~~Selective flooding~~: send packets only in lines where they appear in right direction