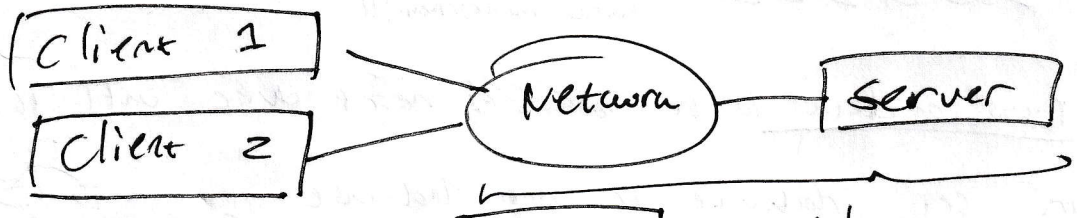
~~Information~~ Internet Tech Exam Revision

→ Network
* intricately connected sys. of things or people
* Computer network: Collection of autonomous computers interconnected by a single tech. Not necessarily a physical connection.
* Network device: eg. PC, Phone, etc. (devices inside network)
* server: provider of a service; accept req. from clients
* Client: A network device connecting to a server and requesting a service
* Packet: Message sent b/w 2 network devices
* Ip address: unique no. identifying a network device
* Internet is a network of networks; connecting all comp. networks!!!
* www is a distributed sys. that runs on top of the internet.
* w/ internet, we can connect to the servers together and connect clients w/ servers and answer req. from others.
* client-server network model



→ Types of Transmission Tech (blackbox)
* Broadcast links: single comm. channel shared by all machines in a network.
* Packets sent by any machine are recieved by everyone and processed by other machines
* Point-Point links: Data from sender is not seen by other machines
* Unicasting: single sender and reciever
* Multicasting: Transmission to a subset of machines. Sender can send to multiple recievers simultaneously.

- Flow of information: travels from upper layer to physical medium to other
- 2 communicating parties talk virtually using protocol.

→ Info flow

- L5 pass message to L4. L4 adds header to complete pack.
  L4 pass to L3. L3 split info to 2 pieces based on protocols.
  Smaller messages to avoid overloading network. send to L2.
  Each piece is treated as individual messages. Depending on protocols,
  add additional headers and trailers, pass to other side.
  Each layer only focuses on info added by its layer part.
  Only 2 HL same L3 treat message from L4 as whole
  message. L3 does not care how big HL is.
- Header = address

→ Services

- Services are provided by lower layer for upper layer and upper layer access them through interfaces (API)
- Connection-oriented: connect, use, disconnect like
- Connectionless: send (order doesn't matter)

service	examples	
Reliable message sys.	seq. of pages	Connection-oriented
Reliable byte sys.	Movie download	
Unreliable connection	Voice over IP	
Unreliable datagram	Electronic junk mail	Connectionless
Acknowledged datagram	Text messages	
req-reply	DB query	

→ Service primitives

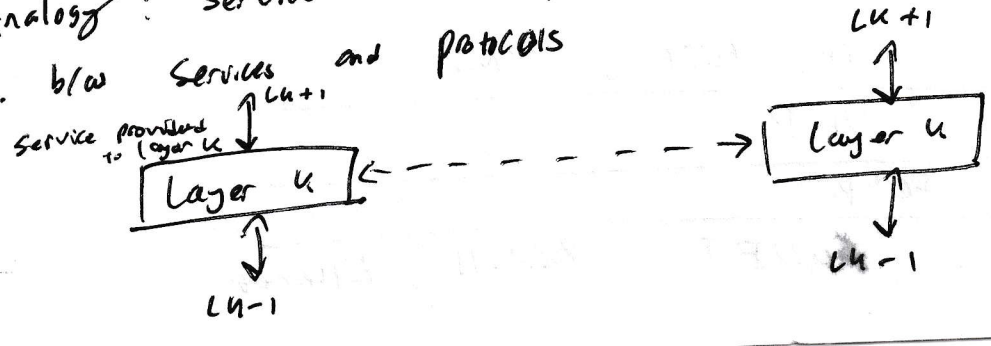
- formal set of op. for services
- put primitives together to provide services

Primitive	Meaning	
LISTEN	Block waiting for an incoming connection	Connect
CONNECT	Establish a connection w/ a waiting peer	
ACCEPT	Accept an incoming connection from a peer	
RECIEVE	Block waiting for an incoming message	Use
SEND	send message to peer	
DISCONNECT	Terminate connection	disconnect

Berkeley sockets

- Analogy: service is a program, primitives are fn.

→ Rel. b/w services and protocols



→ Comparison b/w OSI and TCP/IP

- OSI has 7 layers, TCP/IP has 4
- OSI distinguishes 3 concepts: services, interfaces, protocols
- OSI have 2 layers (presentation, session) empty and 2 crumpled (Network, data link)
- OSI has Bad implementation by companies
- TCP/IP is not a general model, so not adaptable
- TCP/IP does not split data link and physical
- TCP/IP doesn't distinguish b/w service, interface and protocol

→ Hybrid model

5	Application
4	Transport
3	Network
2	Data Link
1	Physical

→ Layering leads to encapsulation

- User data + header = payload
- Data link layer adds header and frame check seq. (FCS) that is to verify that the data hasn't been modified.
- Each layer of the network appends its own header and encapsulates it.

→ On the other side, layers extract original data via decapsulation

→ Addresses

- IP address (Network layer): route info from one network to another. Static or dynamic. Public or private. Used to identify connected device.
- MAC address (Data Link layer): Physical address that exist in network device. Any device that is capable of connecting to a network will have a physical no. hard coded into network device.

→ Transmission Control Protocol (TCP)

- TCP at transport layer help w/ specific services (release and validation)
- Connection-oriented transmission of data b/w 2 devices
- Keeps track of everything transmitted. Ensures reliability through retransmission of lost/corrupted data
- Treats data as segments
- 3 way handshake

→ 3-way handshake

Host A

1.) Send SYN req. (Synchronized req.)

SYN

Host B

2.) SYN recieved, trying to establish connection since TCP is connection-oriented

If it can make connection: send SYN, ACK

SYN-ACK

3.) SYN recovered

Connection established

ACK

Once connection is established, both are free to send data back and forth.

→ UDP (User datagram protocol)

- Connectionless
- Relies on upper layers for error detection and correction
- One way message
- Treat data as segments

* Types based on diameters
1.) single-mode: narrow core, light can bounce; only single ray; long distances
2.) Multi-mode: light can bounce (multiple rays can be transmitted simultaneously)
* Good for bus and ring topologies; scalable in network media
(local area network (LAN) or wide Area Network (WAN))

* optical receiver → signal regenerator (to/from comp) → optical transmitter

Property	wires (twisted pairs or coaxial)	Fibre
Distance	short	long
Bandwidth	Moderate	very high
Security	Easy to tap	Hard to tap
Cost	Inexpensive	More Expensive
convenience	Easy to use	Harder to use

* Wireless Transmission
* Mobility (connect on the go)
* EM wave propagation
* Susceptible to collisions and interference due to broadcasting
* EM waves
- Freq: No. of oscillations per sec (Hz)
- wavelength: distance btw 2 consecutive minima or maxima
- speed: c
- $\lambda f = c \rightarrow m \times \frac{1}{s} = ms^{-1}$

* EM spectrum
- Radio: wide-area broadcast (long distance)
- Microwave: LANs and 3G/4G
- Infrared/Light: line-of-sight
- higher freq → carry more info per unit of time
but doesn't propagate well through walls or higher dist.

* Comm. Satellites
- satellites use microwaves that create artificial channel btw source transmitter and receiver anywhere on earth.
- Effective for anywhere anytime communications
- types: geostationary (35 000 km, 24-hour orbital period (so fixed rel. to us), latency: 270 ms, high coverage), medium-earth orbit (5000 - 15000 km (Navigation), latency: 95-85 ms, med. coverage), low-earth orbits (<5000 km, latency: 1-7 ms, low coverage)

Satellite	Fibre
+ rapidly set up anywhere/anytime comm. + Can broadcast to large regions - limited bandwidth and interference, susceptible to collisions	+ high bandwidth over long distance - installation is difficult

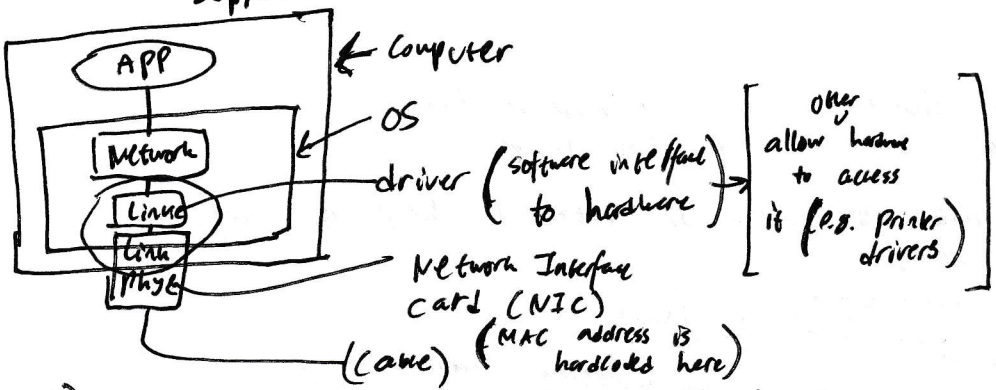
Wireless	Wired
+ supports mobility	+ Easy to maintain a fixed data rate
+ -11- broadcast	+ support point-to-point transmission
+ Easy and inexpensive to deploy	- exp. to deploy over distances
- Transmissions interfere and must be managed	- doesn't support mobility or broadcast
- Signal strengths and data rates vary greatly	

* Data Comm. using signals
* vary physical prop of signals to transmit information
* Sinewave: $f(t) = C \sin(at + b)$
C = amplitude b = phase
$\frac{a}{2\pi}$ = freq

amplitude = highest intensity & energy of signal
Freq = cycles per sec
Phase = pos. of waveform rel. to 0

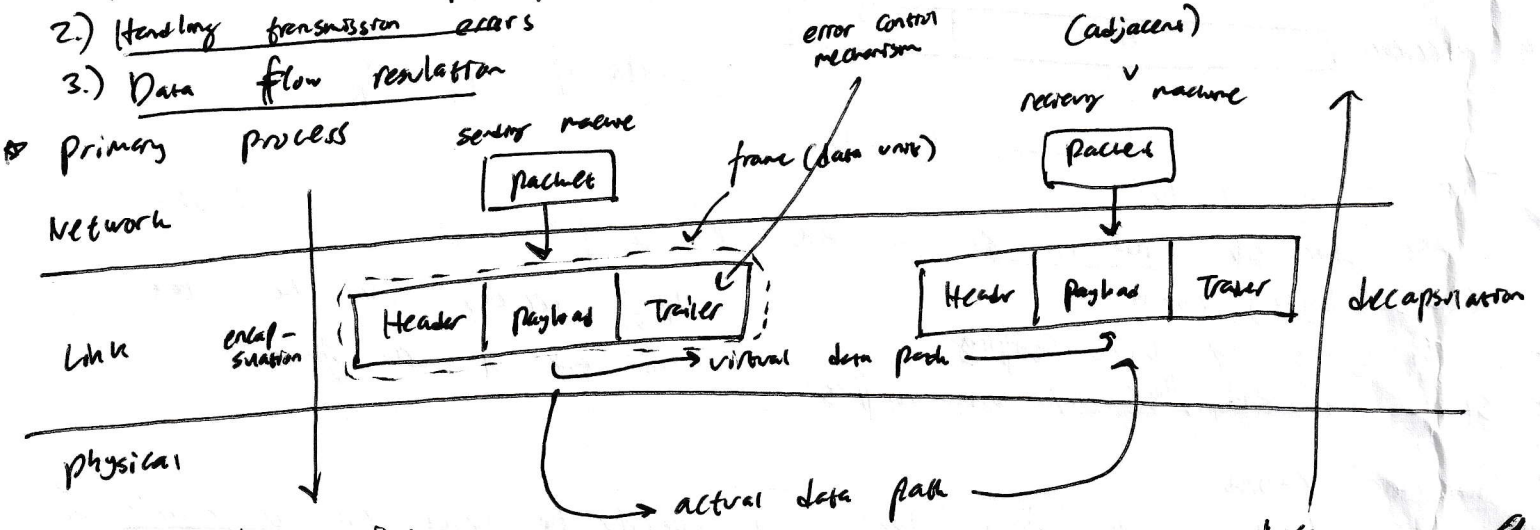
→ Data Link Layer

✿ Support reliable and efficient comm. of "frames" b/w 2 adjacent machines
(not from source to dest., but from source to 1st stop (e.g. router))
supported by error and flow control



✿ 1.) provide service interface for network layer
2.) Handling transmission errors
3.) Data flow regulation

✿ Primary process



✿ Types of services
* Connectionless vs. connection-oriented : whether a connection is setup b4 sending messages
* Acknowledged vs. Unacknowledged : whether the receiver gives the sender an acknowledgement upon receiving the message.

✿ Services provided to Network Layer
* Unacknowledged Connectionless
* acknowledged Connectionless
* acknowledged Connection-Oriented

✿ Unacknowledged Connectionless
* no confirmation from receiver & : unacknowledged
* if frame is lost, DL Layer won't try to recover (left to other layers)
* frames are sent independently (order doesn't matter) : Connectionless
* application: • Ethernet LAN's (short distance, noise is reliable & high quality channel))
Real time traffic (care abt speed, so we don't care abt reliability)

✿ Acknowledged Connectionless
* Each frame is individually acknowledged, and retransmitted if lost
* no logical connection establishment or release
* application: Wireless - IEEE 802.11 Wifi

- Flag stuffing w/ ~~Byte~~ stuffing (still use FLAG)
  - Frames can contain an arbitrary no. of bits (don't need to be multiple of 8 since 1 byte = 8 bits)
  - Each frame begins and ends w/ special bit pattern: 01111110 → flag
  - Stop this pattern appearing in message. Insert one 0 after every 5 1s.

original: 0110 111 11111111111111 0010

sent: 0110 1111 0 11111 0 11111 0 1 0010

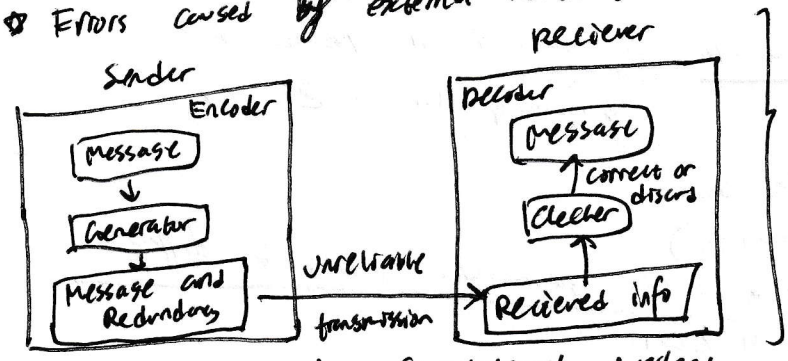
destuffing: 0110 1111 11111111111111 0010 → remove 0 after 5 consecutive 1s

stuffed bits

receiver doesn't know whether a 0 is sent by sender or by DL layer.

→ Error Control

- Adding check bits to ensure that a garbled message by the physical layer is not considered as the original by receiver
- Detecting and retransmitting → boolean (y or n) (don't care abt loc. of errors)
- Correcting → (y or n and correction (flip bits))
- Types of errors
  - Single-bit error
  - burst error: 2 or more bits changed. Easier to detect but harder to resolve
- Errors caused by external noise and length of burst depends on data rate and noise.



Detection: discard and ask for retransmission

Correct: fix and send to upper layer

Redundancy is important for error control

- 1.) Fast and low computational overhead
  2.) min amount of extra bits
  3.) detection of different kinds of errors
- Error detection: Compared recieved data w/ valid data.
  Error correction: use majority voting to find error loc and flip them.
- Error Bounds – Hamming Distance
  - Error control method turns data of n bits into codewords of n+k bits (data + redundancy (k bits))
  - Hamming distance is min. bit flips to turn one valid codeword into any other valid one.
- A Code w/ hamming distance:
  - d+1 → can detect up to d errors (can detect all <d errors)
  - 2d+1 → can correct up to d errors (hamming dist. is capacity, conservative estimate)
- Error Correction: Map to the nearest valid Codeword
  Error detection: Check if they are invalid codeword

→ Hamming Code (Correction)

✡ determine checkbits (redundancy) for n bits of data

at least k check bits $n \leq 2^k - k - 1$

e.g. data 0101 → $4 = (2^3) - 3 - 1$, ∴ $k = 3$ check bits

✡ Checkbits are put in positions P that are power of 2 (1, 2, 4, 8, ...), starting w/ pos. 1

✡ Check bit in pos. P is parity of pos. w/ a P term in their value.

1.) line up every position and set ? for positions that are powers of 2. This is only to be check bits. e.g. P_1 P_2 P_3 P_4 P_5 P_6 P_7
? ? 0 ? 1 0 1

2.) Assume even or odd parity

3.) make a table converting all pos. no. to binary e.g.

	2^2	2^1	2^0
1	0	0	1
2	0	1	0
3	0	1	1

4.) Group pos. by powers of 2 (if P_1 has 1 in 2^0, then it is in 2^0 row)

5.) For each row, find the check bit using parity (e.g. P_1 011, assume even parity, $P_1 = 0$)

6.) Send the data w/ check bits in place to reciever

7.) Reciever groups the received data in terms of $2^0, 2^1, \ldots$

8.) Find if they are all in even or odd parity (as agreed)

9.) any row that is wrong, add the check bit pos. belonging to the wrong rows and the sum pos. is where the error is.

✡ Hamming distance = 3, theoretically could only correct 1 error

→ Error Control Discussion

✡ Error Correction: More efficient in noisy transmission, e.g. wireless

✡ Error Detection: more efficient in reliable w/ lower error rates
e.g. quality wires

✡ Error can be in the checkbits! Correct errors using Hamming Code.

✡ Redundancy is in the code for Hamming unlike others where its at the end

→ Flow Control

✡ Strategies to control when sender can send next frame.

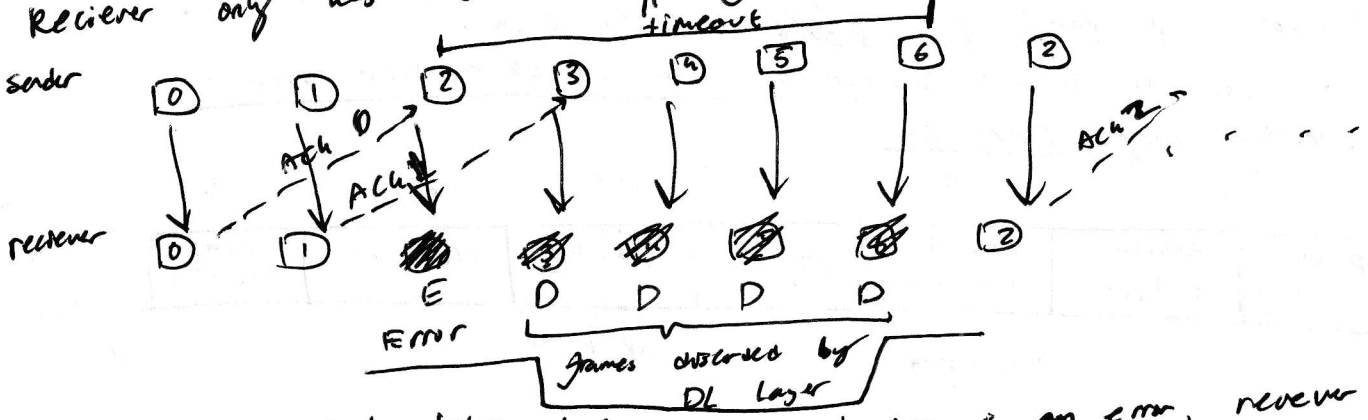
✡ 1.) Feedback based flow control → based on acknowledgement

2.) Rate based flow control → mostly used in transport layer

✡ Sender sends data at its own pace and receiver process them when they get the data. If reciever can't process all, they put them in a buffer, and accept them.

✡ Acknowledged Transmission: if fast sender and slow receiver. To avoid overloading, use acknowledgement (traffic light). (half duplex)

* Go-Back-N (sliding window implementation)
- Sender has window size of N, so they can send up to N frames w/o ACK.
- Reciever only has 1 size buffer (like in stop and wait)



- ACK is sent much later by reciever. If there is an error, reciever starts discarding sent frames during timeout interval until reciever reaches timeout and sent f.2.
- whenever an ACK is recieved, the sliding window moves to the right:

[0, 1, 2, 3, 4, 5] —ACK for 0→ [1, 2, 3, 4, 5, 6] —ACK for 1→ [2, 3, 4, 5, 6, 7] → ...

- Cons: lost a lot of bandwidth.
- If timeout interval is too short, another frame might be sent before an ACK is sent.

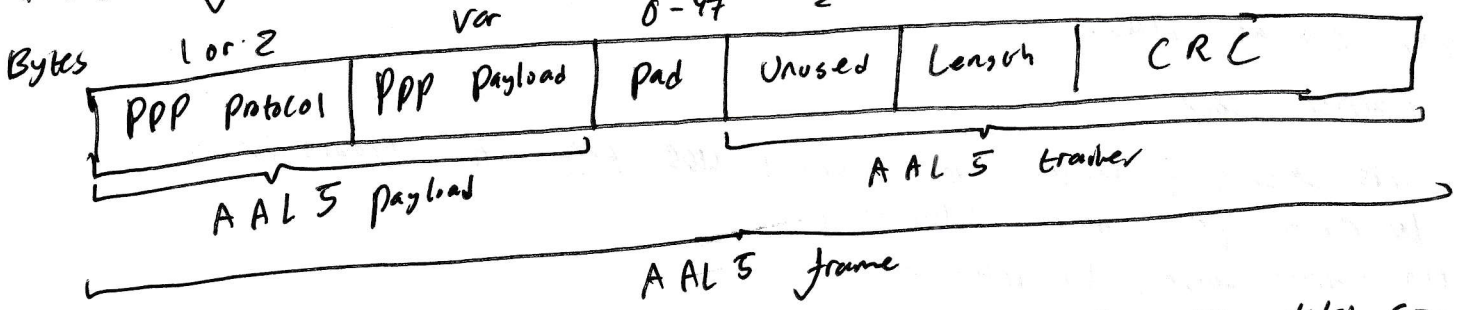
* Selective Repeat (Sliding window implementation)
- Reciever accepts frames anywhere in recieve window
- NAK (negative acknowledgement) triggers retransmission of missing frame before timeout. (eg. If frame 3 is not recieved, but 4 is, it sends NAK for frame 3 and continues to send ACK for others)
- Cumulative ack indicates highest in-order frame recieved. Don't need to send ack for every sent frame.

* Comparison b/w sliding window approaches

Go-back-N	Selective Repeat
•Reciever discards all subsequent frames from error point, sending no ack until recieving next frame in seq. (timeout is reached)	•Reciever buffers good frames after an error point and relies on sender to resend oldest unacknowledged frames.
•Buffer only at Sender Side	•Buffer at both sides
•Bandwidth ↓ space ↑ (CU)	•Bandwidth ↑ space ↓ (CU)

☆ ADSL

* PPP data is sent in ATM cells over ADSL
* ATM defines fixed-size cells (53 bytes); each cell has a virtual circuit identifier.
  - PPP don't have fixed frame. So they need AAL5 frame as adaptation layer to prepare ATM to create fixed-size cells.
  - AAL5 frame is converted to ATM cells



  - Padding ! help to create fixed-size ATM cell. To make total size multiple of 48-byte pieces. Each AAL5 frame goes into 5-byte header 2 ATM cell w/ total 53 bytes
  - fixed size → easier management

→ MAC sub-layer (Medium Access Control)

☆ If we have point-to-point network, don't have to consider MAC since it's only for simple sender and receiver pairs.

☆ MAC only for Broadcast networks.

☆ How to share the channel among users?

☆ located b/w Data Link and physical layers.

☆ Types of Channel Allocation mechanism

* Static Channel Allocation } depends whether allocation changes
* Dynamic ——— " ——— } based on needs

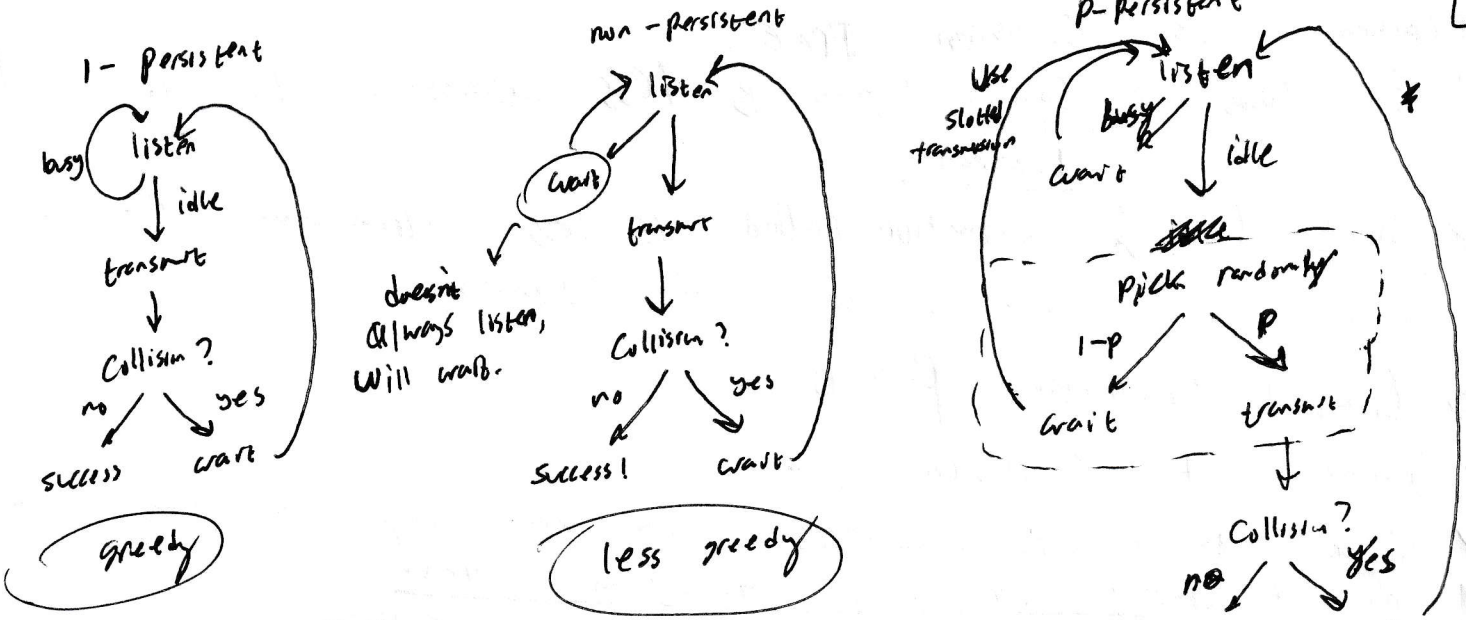
☆ Static Channel allocation : Divide a channel into segments and each user is allocated a dedicated segment for transmission

* Time Division multiplexing (TDM) (fixed schedule) (take turns on a slot may be empty → waste)
* Freq ——— " ——— (FDM) (split freq. band into N users and assign them to each user. Access part of the channel continously)

* Good for fixed no. of users, but there are peak hours and off-peak hours, where there are different no. of users at a time. Not efficient since network traffic is bursty but static methods TDM and FDM try to give consistent access to network.

☆ Dynamic Channel Allocation

* Assumptions :



→ Efficiency Comparisons

throughput = (ave. amount of data sent successfully within packet time C$ success rate)

ALOHA < Slotted ALOHA < 1-persistent < Non-persistent < p-persistent

w/ $p = 0.01$

P↓ postpones transmission but increases throughput

→ CSMA w/ Collision Detection

* Collision detection: detection during transmission
* Previous CSMA $\rightarrow$ detection occurs after transmission
* After Collision detected, abort transmission, wait random period, try again
* Channel must be continually monitored
* Reduce contention times to improve performance

→ Collision Free (make central management to avoid collisions by letting everyone known when to send frame)

* Bit Map protocol
* Reservation-based protocol
* Division of transmission right, transmission event - no collisions
* Similar to TDM, but the contention slots are dynamic where we know who should send (not static).

1 user: $\frac{d}{d+N}$	N users: $\frac{Nd}{Nd+N} = \frac{d}{d+1}$	efficiency bounds

N = bits during contention slots

* Binary Countdown Protocol
* Determine order based on binary address of stations
* higher address (higher no. stations) $\rightarrow$ higher priority

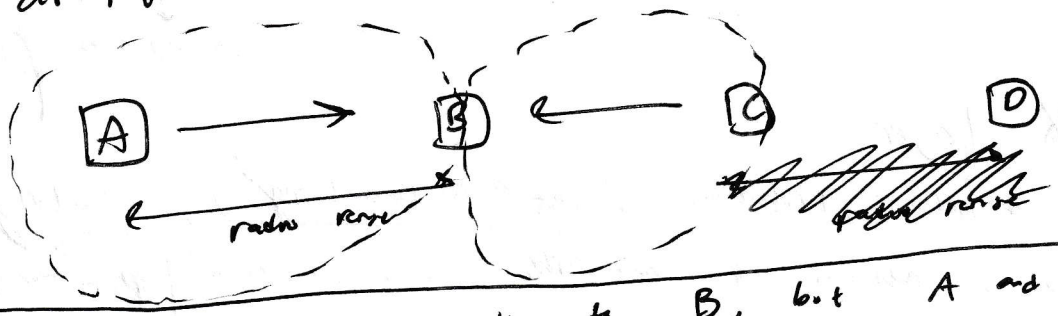
} send bit-by-bit starting w/ higher order bit

* stations send address from high-order bit to low-order bit in contention slots ($\log_2 N$ slots).

Address				
0	0	0	1	0
0	0	1	0	0
0	1	0	0	1
0	1	0	1	0

* look to each column from left to right, stations give up if they have 0
* keep going right until a station sees its full address to be sent.

→ Hidden Terminals
* Senders that cannot sense each other but nonetheless collide at intended reciever.



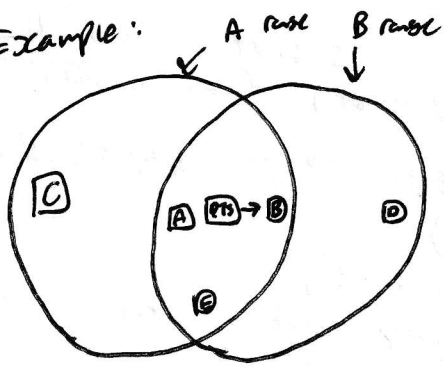
Both A and C can transmit to B, but A and C can't hear each other

* Want to prevent; loss of efficiency
* only B can tell if there is collision (reciever)

→ Exposed Terminals
* senders who can sense each other but still transmit safely to different recievers.
* Desirable concurrency; improves performance

→ MACA (Multiple Access w/ Collision Avoidance)
* Sender asks reciever to transmit short control frame
* Stations near reciever hear control frame
* Sender can then transmit data to reciever

* Example:

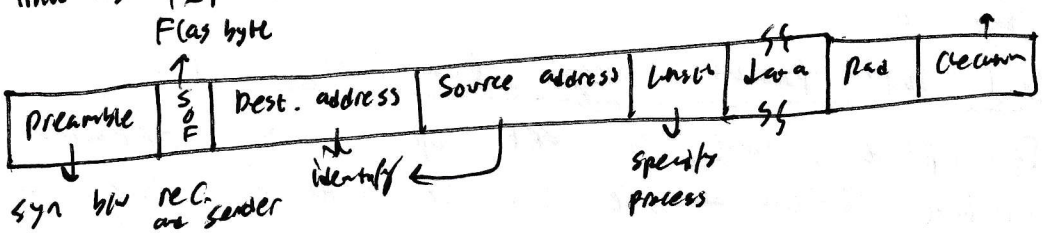


* A and D are hidden terminals for B since both can't hear each other but can transmit to B.
* RTS: Req. to send
  CTS: Clear to send
  } confirmation for B to receive data and tells other stations

1.) A sends RTS to B; C and E hear and defer for CTS
2.) B replies w/ CTS; D and E hear and defer for data

→ Ethernet
* Classic → uses bus line → 1-persistent CSMA/CD
* Switched

IEEE 802.3



→ Datagram (Connectionless) vs. Virtual-Circuit Subnets (Connection oriented) R3

Issue	Datagram Network	Virtual-Circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains full source and dest. address	Each packet contains short VC no.
Effect of router failures	None, except for packets lost	All VCs that passed through failed router are terminated
QOS	Difficult	Easy
Congestion Control	Difficult	Easy

- VC: router has to store all connections and packets on the path
  If router fails, all connections must be reset.

☆ Time Consumption
* VC: requires setup time and resources, but packet transmission is very fast afterwards
* Datagram: more address parsing time due to complicated lookup procedure. (routing table is dynamic)

☆ Memory of Router
* VC: requires entry per VC (in and out)
* Datagram: keep entry for each destination

☆ Bend width
* VC: saves forwarding overhead. Save space when storing addresses (only address for in and out). still needed for setup.
* Datagram: full dest. address in every packet.

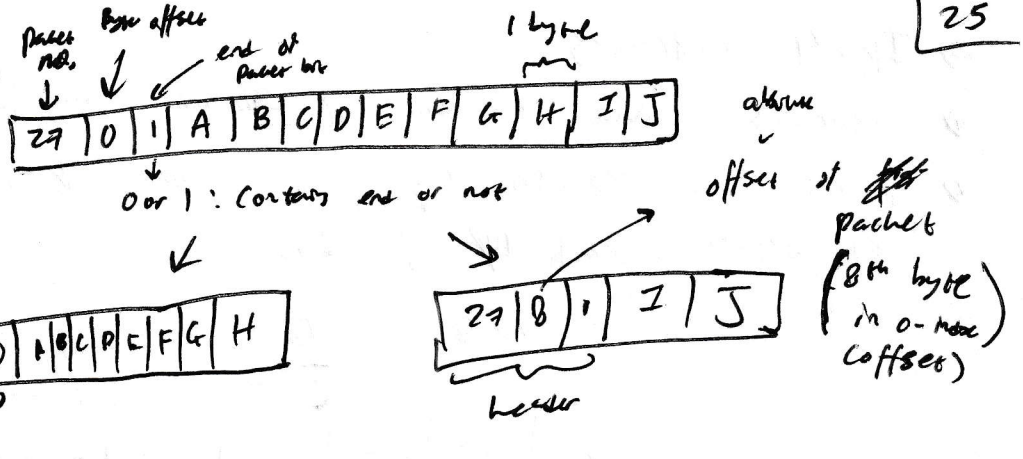
☆ Longevity (long-term or short-term service)
* VC: can be setup for repeating and long-running uses
  e.g. permanent VCs

☆ Vulnerability
* VC: have to be reset if one router fails
* Datagram: can use alternative route

→ Different Networks offered
☆ Service
☆ Packet Size: Networks may have size limit
☆ Addressing: different lengths (IPV4 vs. IPV6) (flat vs. hierarchical)
☆ QoS: Present or absent
☆ Reliability: different levels of loss
☆ security: privacy rules, encryption

e.g. MAC ↑	e.g. IP ↑

- Ip - Style Fragmentation



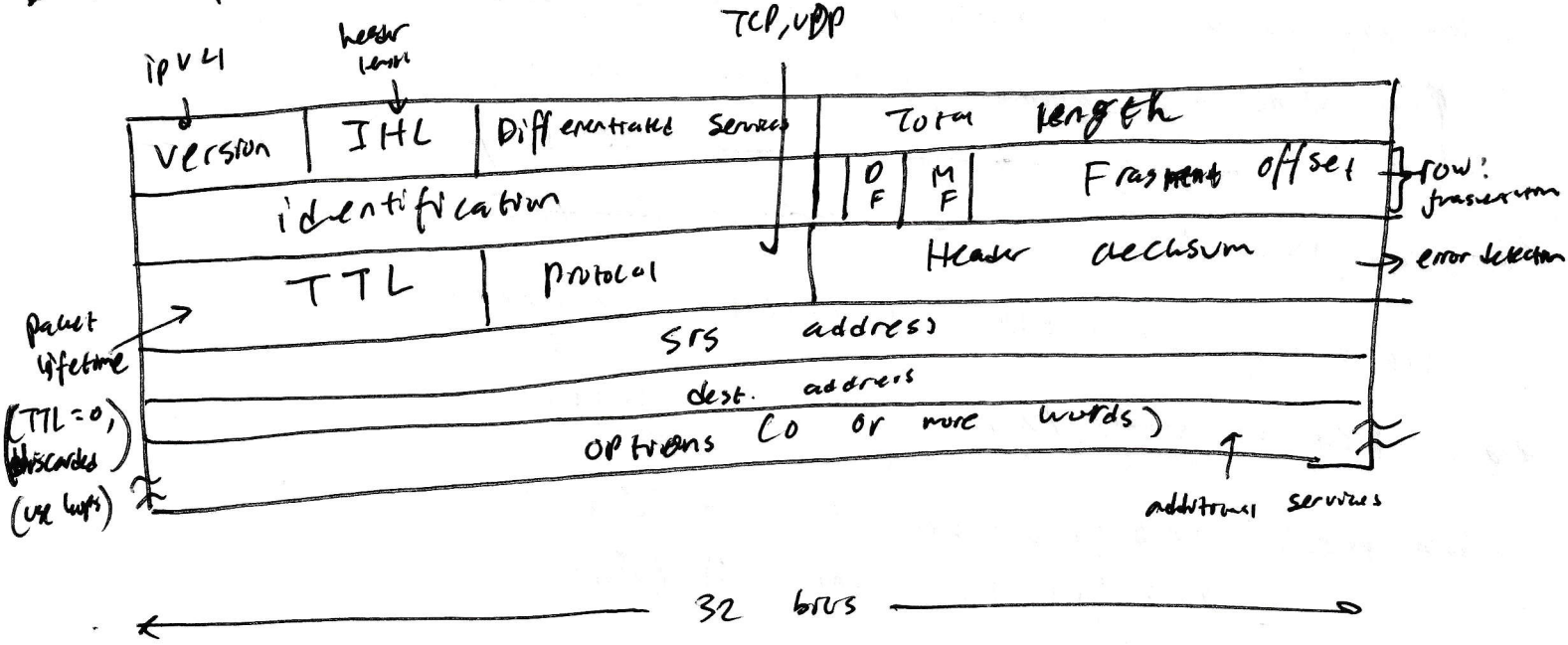
- Splitting is based on MTU

* Path MTU Discovery

- Alternative to fragmentation
- Only source and dest. split and reassemble. NO split in network
- Find MTU of the whole path. Find min MTU in the path.
  Use trial and error process to find min MTU.
- If routes and path MTU change, new error packets will trigger, and source adapt to new path.
- Router can tell source the next MTU, but not the path MTU.
- Needs to send multiple packets to find out path MTU.
- May create more overhead if only few packets need to send.

→ IPV4 Datagram Structure

- Datagram = packet
- IPv4 datagram consists of header and payload
- Header format: 20-byte fixed part + var.-length optional part



- Subnet mask: All 1s in the network portion.
  Can be used to extract prefix from any given IP address. (no. of 1s is the network portion, no. of 0s is the host portion)

→ IP Addressing and Routing Tables

- Triplet (Net dest., Subnet Mask, Outgoing line (phys, virtual) ↓ symbolic links)

Dest	Subnet	interface
128.16.3.0	255.255.255.0	Eth 0

- Subnets
  * Subnetting: allocate block to an organization and they can further divide block into smaller blocks and use based on their needs. split network into several parts.

Network is divided into subnets internally, but looks like a single prefix outside the network.

* no. of hosts a network can support = $2^{32-\text{Network portion}}$

↓ network portion → ↑ host portion → ↑ network

e.g. 1) 128.208.0.0/18 → $2^{32-18} = 2^{14}$, 14 = host portion

128.208.00|xxxxxx xxxxxxxx (14)

3) 128.208.128.0/17 → $2^{32-17} = 2^{15}$, 15 = host portion

128.208.1|xxxxxxx xxxxxxxx (15)

2) 128.208.96.0/19 → $2^{32-19} = 2^{13}$, 13 = host portion

128.208.011|xxxxx xxxxxxxx (13)

block: 128.208.0.0/16 → no. of address 2^{16}

128.208.|xxxxxxxx xxxxxxxx 2^{16}

⇒ ICMP

* Testing and monitoring antwork Conditions b/w host and routers

Message type	Description
→ Dest. unreachable	Packet could not be delivered
→ Time exceeded	TTL = 0
Parameter Problem	invalid header
Source quench	Choke packet
Redirect	Teach a router abt geography
→ Echo & Echo reply	Check if machine is alive
Timestamp req. / reply	Same as Echo, but w/ timestamp
Router advert / solicitation	Find a nearby router

→ Routing

* Consider network as a graph of nodes and links
* process of discovering network paths
* Decide what to optimise: hops, delay, etc.
* Update routes for changes in topology (e.g. router failures)
* Routing algorithm: deciding on which output line an incoming packet should be transmitted.
* Non-adaptive algo. (static)
* static routing, static decision-making process
* adaptive algo. (dynamic)
* Dynamic routing, dynamic decision-making process
* changes in network topology, traffic, etc.

→ Optimality Principle

If router B is on the optimal path from A to C, then optimal path from B to C also falls on the same route.

→ Sink Tree

* Routing algo. try to discover sink tree and utilize for all routers
* Sink tree: set of optimal routes from all sources to a given dest. forms a tree rooted at the dest.
* Each sink tree contains path for other sources to send data to destination.