Link Layer — OSI Model

- Prepares data for transmission over a physical medium. Before it is sent

- Links include;

  - wired
  - wireless
  - LANs

- Takes packets from network layer and wraps them into smaller units called "<u>Frames</u>"

- Adds another header to the frames and sometimes a trailer

- Header contains MAC addresses to tell the the network where the frame is coming from and where it is heading (on the local network level)

- When multiple devices share the same connection, the link layer helps manage access, who comm[illegible]

- It handles the transfer of data ~~frame~~ from node to physically adjacent node.

  Pace of sent and recieved nodes controlled by link layer

- Detect Errors caused by noise or attenuation, And ~~corrects errors~~ discards recieved errors or retransmits perceived error frames

- Error correction to some degree and will correct bit errors) without retransmission

- Half and Full Duplex:
  Half ~~du~~ allows communication only from one node at a time between two ~~end~~ nodes.
  Full allows simultaneous transfer of data in both directions, like a phone call

- Every host has a link layer implementation, This is on a Network interface card or on a chip

- On the NIC, functions such as;

encapsulating packets into frames and all previously mentioned functions.

- The link layer is a combination of hardware, software and firmware
- The sending side's data link frames and adds error checking bits
- recieving side extracts data from, passes to upper layer at ~~the~~ recieving side

Checksums & error detection

- odd parity and even parity bits make the ~~total~~ parity for the entire binary string either odd or even, this way you know you have an error or not

odd parity protocol;

i.e 1101(0)

- Parity bit is 0 to make sure total parity stays odd, if it is recieved as even parity, something has gone wrong

- its the inverse for even parity protocols

- 2D parity bitchecks detect errors in blocks of data and can even correct some errors

- take this matrix as data; even parity system;

0 1 0 1 (0)
1 1 0 0 (0)
1 0 0 0 (1)
0 1 1 1 (1)
(0)(1)(1)(0)

→ row ~~Column~~ parity = [0,0,1,1]

column parity = [0,1,1,0]

- Then ~~using~~ is the row or colum parity does not match when recieved, that indicates an ~~ero~~ error.

- And because the rows equate to x and Y positions of the bits, you can triangulate wherere the error has occured

is row Parity = 1101

calum parity = 1000

flipped bit

flipped bit

- You know the bit at row [3] and calum [1] is ~~th~~ the specific bit, thus allowing you to flip it back

- only works for a smaller ammount of errors

- **Check Sums** ~~or~~ first divide the data into specific word sizes (16 bit for networking)

- then add them up using one's complement addition

- If there is any overflow (if a bit is carried past the most significant 16th bit, thus exceeding the 16 bit word size) simply add it back to the least significant bit; end-around carry

- Then take the one's complement (inverted bits) of the final sum to get the Check Sum

- then the reciever adds the check sum to their added sum and it should all equal 1s or there is an error

- ensures greater integrity and avoids double flip bits from being missed like in parity checks

take the ~~checksum~~ value

$$\underbrace{1\ 0001\ 0001\ 0111\ 0000}_{=44644}$$

overflow bit

$$\underbrace{}_{=70000}$$

adds 1 for wrap around carry $1 << 16 = 65536$

$70001 - 65536 = 4465$

ond so our ~~[illegible]~~ Final sum is 4465

- to get the modulus as a binary number in python, we can use the & (bitwise and) operator.

- create a mask of 1s like so;

1111 1111 1111 1111

→ (1<<16) - 1 = 1 0000 0000 0000 0000 − 1

65535 = 1111 1111 1111 1111

- then and the two binary strings together and cut off the 17th highest bit

44 70001 = 1 0001 0001 0111 0001

&

(1<<16)-1 = 1111 1111 1111 1111

and together = 0001 0001 0111 0001

thus giving us the modulus doable in code.

~~. Then convert the result to get our check sum;~~

~~0001 0001 0111 0001~~

~~1110 1110 1000 1110~~

~~Check Sum of 70000~~

. One's complement of a binary number x is; the max binary − x

So the biggest 16 bit number is 65535 or 1111 1111 1111 1111

1111 1111 1111 1111

- 0001 0001 0111 0001

$\boxed{1110 \quad 1110 \quad 1000 \quad 1110}$

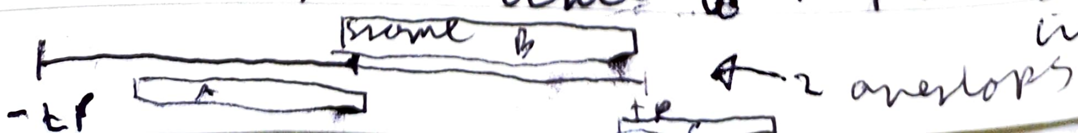
Final check sum

* Crossed out calculations were actually correct

MAP: Multiple Access Protocols

Aloha Pure - Random Access Protocol

- When multiple nodes want to send data down a single channel we use MAPs, one of these is aloha
- As soon as a node has a frame ready to be sent down a channel it will do so.
- ~~It waits f~~ If there is a collision with another frame, both are destroyed
- Senders wait 2 * propogation time (td) to ~~recieve~~ recieve an acknowledgement
- If none is recieved, a random ammount of time is then waited until it resends the frame, this randomness helps avoid collisions with other frames
- This means there is a vulnerable period for any frames to be sent from ~~the~~ - propogation time to + propogation time



- Frame A overlapped with frame B as it was sent between $-t_p$ + start and frame C overlapped with frame B and frame C.

- So the vulnerable period is from $-t_p$ to $+t_p$ for a net ammount of $2t_p$

- Max Throughput:

the equation $S = G \times e^{-2G}$ tells us the ~~[illegible]~~ throughput calculation.

when $G = 0.5$, this is most optimal (one transmission per $2t_p$).

G is the average number of frame transmissions per frame propogation time

which comes to $0.5 \times e^{-1}$

$= 0.184$ to 358

- Why use e for this equation?

we use the Poisson probability formula

the formula states that for a given occourence rate ~~x~~ λ, the probability of having zero events $P(0)$ is;

$$P(0) = e^{-\lambda}$$

Since the vulnerability rate is 2G

~~P~~ the odds no transmissions occur is e^{-2G}

So for every transmission, the odds it occurs (G) at the same time ~~no~~ no overlapping events occur $P(0)$ is

$$G \times e^{-2G}$$

therefore throughput (S) =

$$S = G \times e^{-2G}$$

Slotted Aloha

- designated times to transmit and only at those times.
- The time between transmit times

correspond to the time for faster transmission exactly, reducing collisions

- all users have to sync up their time slots

- If an acknowledgement is not recieved within time-out delay, ~~the~~ frame is retransmitted at next slot

$$S = G \times e^{-G}$$

max throughput at $G = 1$

$$S = \cancel{G \times e^{-G}} \quad 1 \times e^{-1}$$

$$= \frac{1}{e} \quad = 0.368$$

CSMA (carrier Sense Multiple Access)

- recieves datagram from network layer
- each station senses idle or busy for the medium before transmitting. will wait for station to idle before sending

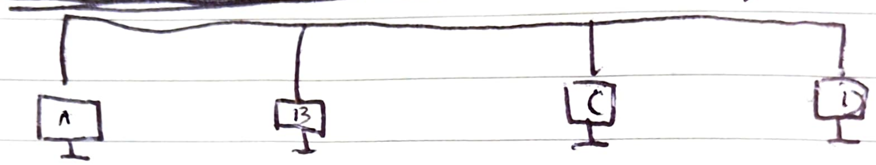
- There is some chance of collision with CSMA due to propagation delay, If there is the frame will continue to be transmitted anyway until ~~...~~ a higher level protocol sends the corruption

- This is because of the time taken for the first bit of its ~~packet~~ frame to reach the destination

~~CSMA/CD (collision D) detection~~

The NIC & MAC explained

Shared medium



- Physical layer implemented within ~~the~~ the NIC (Network Interface card) detects the physical electrical signals, ~~and~~ This is called a transceiver

- These signals are then converted from analog to digital signals using the transceiver's ADC (analogue to digital converter)

- the digital signals are demodulated and decoded into the raw bit stream that was transmitted. ~~that~~

link layer but still on NIC

- The NIC's MAC (media access control) will then take over and process the bit stream into actual segmented frames by ~~the~~ identifying bit sequences defined in its framing protocol

- The error free frame is then ~~transferred to the~~ link layer has been constructed within the MAC* sublayer on the NIC as part of the link layer's mac functions

- The NIC performs physical and data link layer functions particularly those involving the MAC

CSMA/CD (collision Detection)

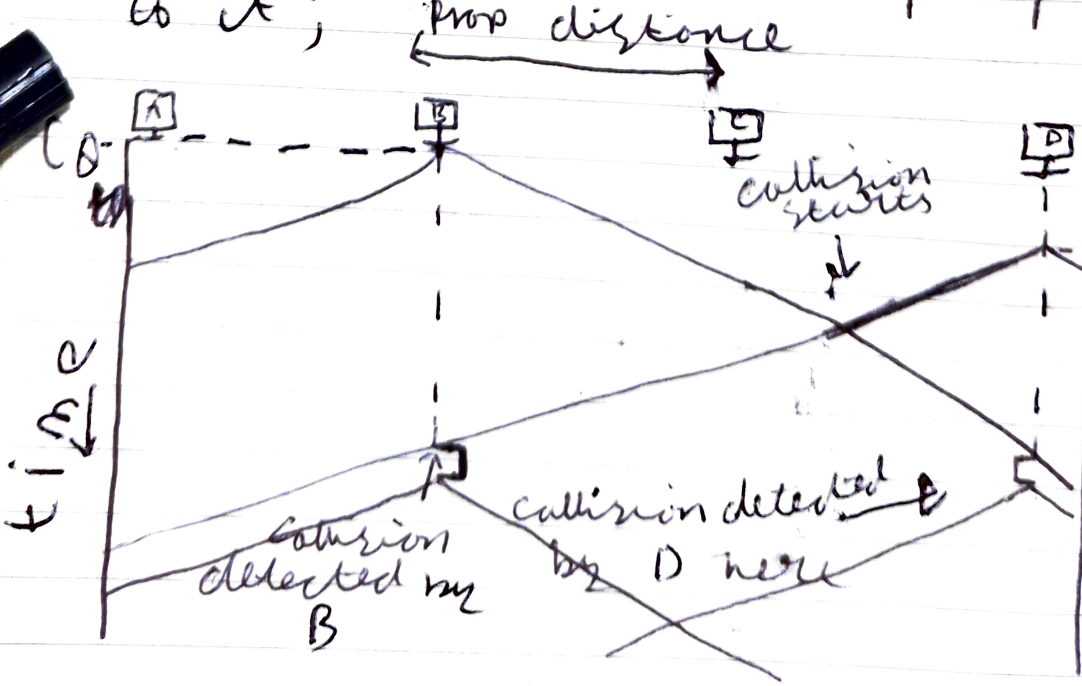
- Unlike CSMA which has no way to detect frame corruption due to overlapping signal, ~~CSA~~ CSMA/CD includes medium signal detection even while transferring a frame.

- This additional MAC functionality will sense a signal transmit elsewhere on the network and abort the transmission, having confirmed that its transmission has indeed been interfered with

- The devices that suffer a collision, typically two but could be more if multiple transmissions are propogated during a vulnerability period, will emit a jam signal onto the network telling all devices to stop transmitting.

- The devices involved in the collision will then enter a "back off period" for a randomized amount of time before trying again

- It is important to remember how propogation times work. If there is a device further along the ~~___~~ cable (medium), it will take longer to propagate to it;



~~If~~ If collisions continue to occur with same frame, back of period ~~ff~~ will increase exponentially