02-Link Layer-
PhysicalLayer

1. **Short Answer Questions**

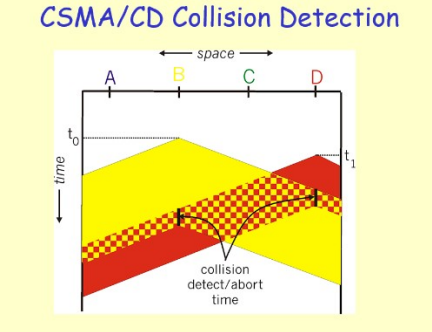
    a. What's **encapsulation** as it applies to protocols. Briefly explain.

Answer: Encapsulation is the process of wrapping the data with the necessary information. For example, placing the header for each packet is encapsulation like adding destination and source MAC addresses.

    b. Briefly explain the "**framing**" problem in the link layer and how PPP solves it.

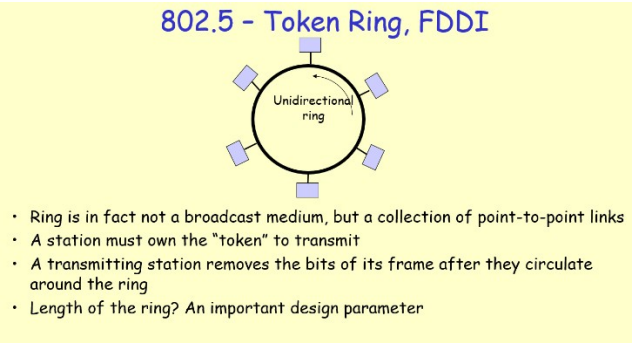
Answer: framing is used to detect frame boundaries. The issue with this is that how can the receiver determine where those bits start and end. PPP solved this by denoting the end and beginning using a special character called the flag (0x7E).

    c. In an Ethernet, why does doubling the bit rate (if everything else remains the same) require halving the maximum cable length?



Answer: Ethernet uses CSMA/CD to detect errors. If two frames are being sent without detecting the collision, then they think that they sent the data successfully despite it being corrupted. With double the bit rate the wire needs to be halved because it would allow the errors to be detected.

    d. In a token ring, a station must wait for the token to come around to it before sending. Why is it not possible for the station to sense the ring and then start transmitting if there is no traffic?



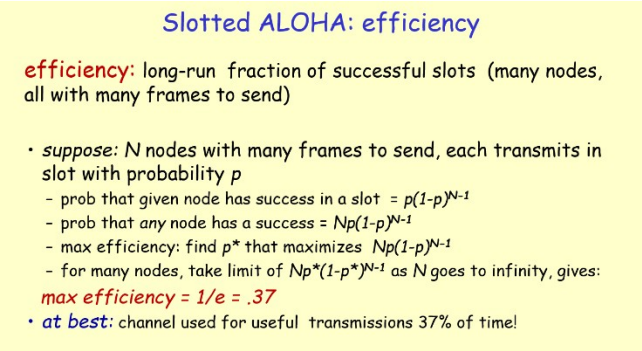
Answer: The token ring system only allows the device to transmit if it has the token. This helps prevent collisions because if they all think no one is transmitting but they want to go there is

no type of authority to decide and they will end up colliding. The token is what allows them to transmit and without it they cant.

e. Smarty Smart thinks that having a **minimum frame size is wasteful for Ethernet**. He proposes that the minimum frame size be reduced to 15 bytes, 14 bytes for the header and 1 byte for the payload. Explain why this may not be a good idea.

Answer: Not only does this waste resources since you are just sending 1 byte worth of data for every 14 bytes of headers which will result in slow speeds, but it also hinders collision detection because it needs to be long enough to reach the other station.

f. In a broadcast channel, the link bandwidth is wasted due to multiple hosts trying to send at once and canceling each other's communication. A simple model of this problem is that time is divided into discrete slots. If a network has n hosts, and the probability of any single host trying to use a slot is p, **what fraction of slots are wasted** due to collisions?



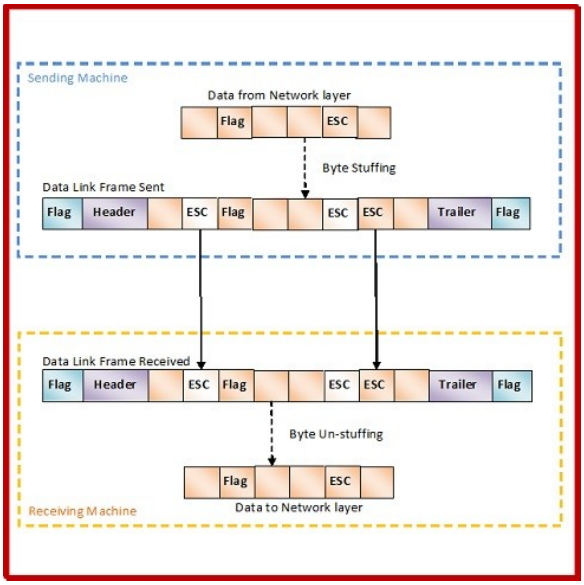
Answer: about 63% because 37% is successful

g. Explain the following terms: MTU, byte stuffing, bit stuffing

Answer:

MTU: Maximum Transfer Unit is the maximum LL frame payload size excluding the LL header, trailer

Byte Stuffing:



- similar to bit stuffing but this involving entire bytes.
- If the flag (ox7e) is detected when an escape character (ox7d) is added.
- If ox7d is detected another one is added.

Bit Stuffing: Used to solve the problem of 0x7e appearing inside the body.

Sender:

- A 0 bit is stuffed after 5 consecutive 1's, i.e. , 11111

Receiver: The receiver looks for 5 1's

- If the 6^{th} bit is a 0 its removed.
- If the 6^{th} is a 1 and 7^{th} is a 0 then it's the end of the frame.
- If the 6^{th} is a 1 and 7^{th} is a 1 then the sender is indicating an abort condition.

h. Consider 4 hosts, A, B, C and D attached together using an **Ethernet hub** into a star topology. Assume that A is sending some data to B. Is it possible for C to send some data to D at the same time? Justify your answer.

Answer: No, it is not possible because this is involving hubs not switches or bridges. Hubs broadcast the data to all connected devices so if it gets something from A it sends it to all other devices.

i. What's the "**type**" or "**protocolNo**" field in a Link Layer (LL) header used for? Do all LLs have to have a "**type**" field in their headers? If a protocol does NOT have this field, what is the implication?

Answer: type field in LL header specifies upper layer protocol. No all LLs do not have to have a type field. If they do not have a type field, then the next protocol layer must be determined in some other way.

j. Consider a link layer that does **NOT** add error detection/correction bits to the end of its frames? What are the implications of this design?

Answer: No error detection means that there is no way for the receiver to verify that the transmitted bits are correct with no errors. It could also be relying on other layers to be the error detection portion of the mechanism.

k. Why is it important for protocols configured on top **Ethernet** to have a **length** in their header indicating how long the message is?

Answer: It helps determine if any bits were lost or accidentally added. It also helps with CRC.

l. What's a MAC address. What is it used for?

Answer: MAC or Medium Access Control address is used to identify a device in the LAN.

m. Briefly describe how Ethernet's Carrier Sense Multiple Access/Collison Detection (CSMA/CD) work? What's the advantage of CSMA/CD over CSMA?

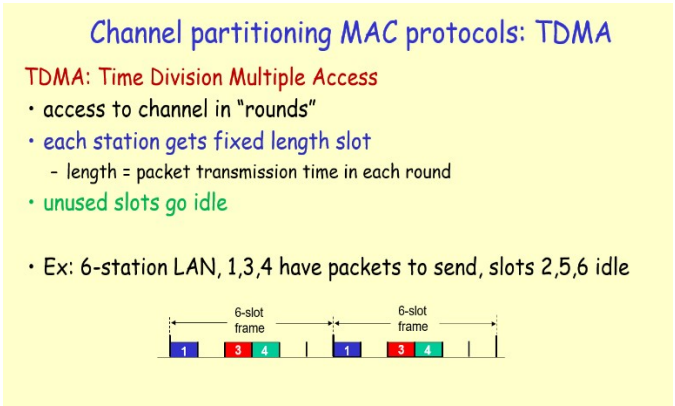
Answer:

CSMA (carrier Sense Multiple Access): listens before transmitting. If the channel is sensed as idle then it transmits the entire packet. If it senses it to be busy then it defer transmission. If it collides then the entire packet transmission time is wasted

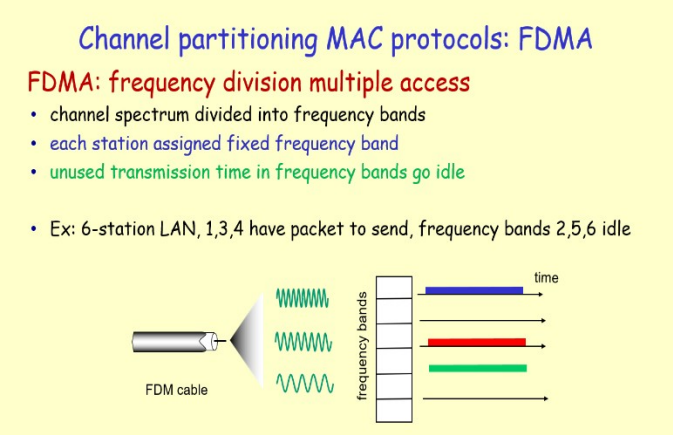
CSMA/CD (collision detection): Similar to CSMA but if a collision is detected then the transmission is aborted reducing channel wastage.

n. Briefly describe how channel partitioning MAC algorithms work. Also describe their advantages and disadvantages.

Answer:



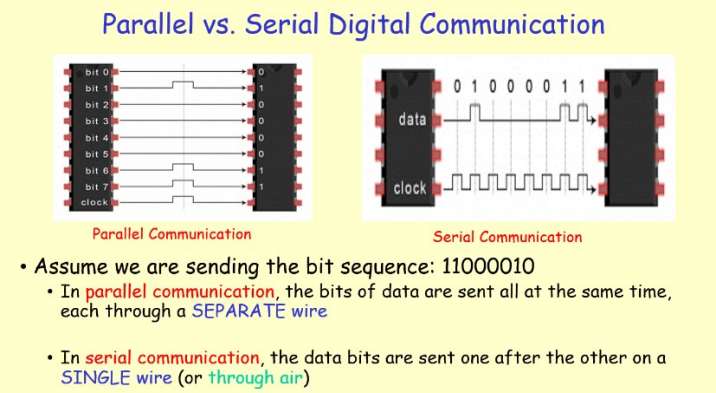
Fixed slots means no collision. Wastes time if not used



simultaneous transmission. Limited bandwidth

o. Briefly explain the difference between serial and parallel communication. Which is preferred in long distance communication?

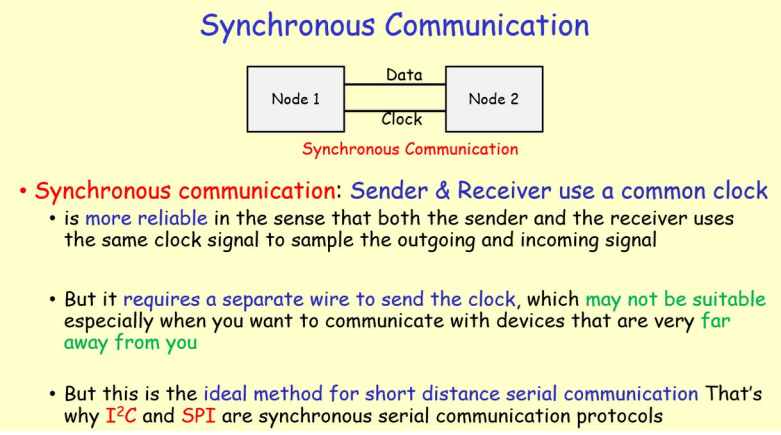
Answer:

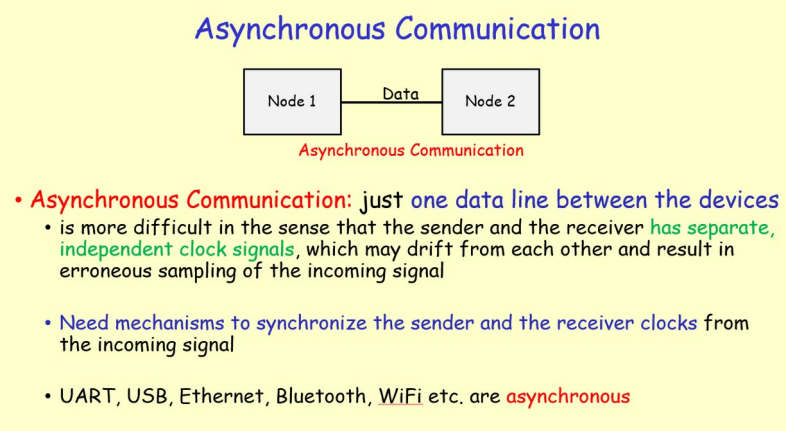


Serial communication is preferred because its more reliable, requires 1 wire, and can reach longer distances.

p. Briefly explain the difference between synchronous and asynchronous communication. Which is preferred in long distance communication and why?

Answer:





Asynchronous communication would be used for long distance communication because it doesn't require a second wire.

q. Give a list of the 3 types of cables used in communication networks. Which cables do high speed Ethernet use?

Answer: Twisted Pair, Coaxial, Fiber. High speed Ethernet uses twisted pair cables.

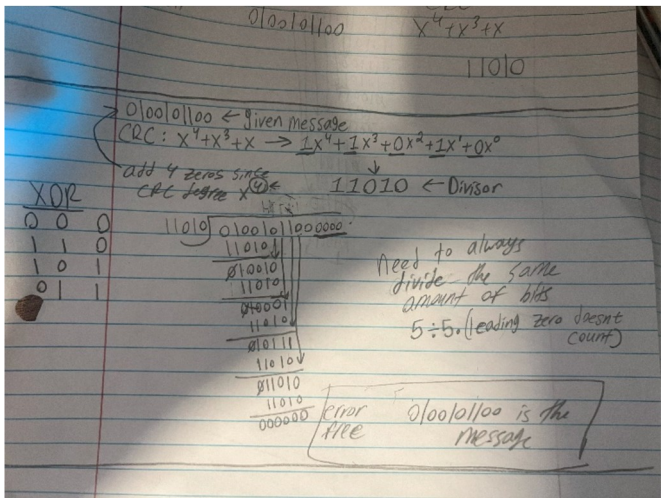
r. Briefly explain the difference between modulation and encoding.

Answer: Modulation is about converting digital data into analog signals suitable for physical transmission while encoding is about representing data in a specific format for transmission.

2. Suppose you want to transmit the following 10-bit message "0100101100".

a. Suppose you want to protect the message from errors using the CRC polynomial $x^4 + x^3 + x$. Use polynomial long division to determine the message that would be transmitted.

Answer:



$x^4 + x^3 + x$ = 1 x^4 + 1x^3 + 0 x^2 + x + 0 = 11010

Since the crc above is a degree of 4, add 4 zeros to the end of the message

0100101100 → 0100101100**0000**

Then divide (we do xor) the message by the crc. Start with 5 bits then drop bits down as needed like long division. Then the remainder gets added to the end of the original message.
Result: 01001011000000

b. Suppose you want to protect the message from errors using **two-dimensional odd** parity. Assume that the message is divided into 5-bit chunks for parity computation purposes. Determine the message that would be transmitted.

Answer:

2d ***ODD*** Parity means we count number of 1s and **add a 1** to make it an odd amount or **add a 0** if its already odd. We stack the chunks and look at each row and column.

Original: 0100101100 (10 bits)

Chunk 1: 01001 (5 bits)

Chunk 2: 01100 (5 bits)

01001 **1** = odd ones

01100 **1** = odd ones

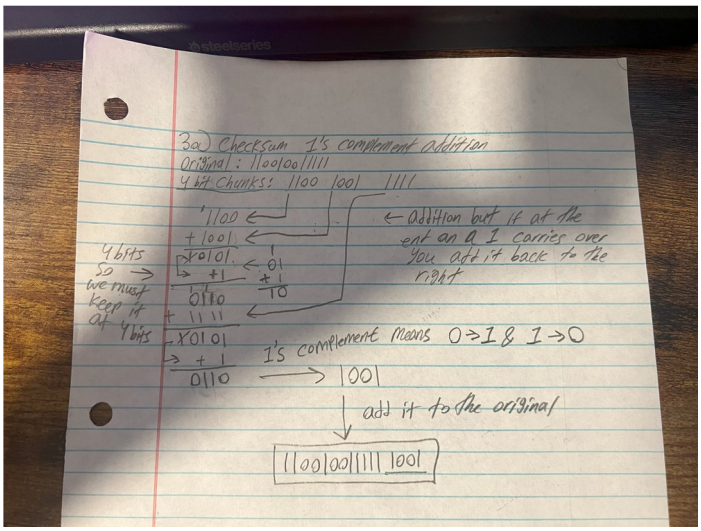
11010 1 = each column now has odd ones

Transmitted message: 010011011001110101 (just combine everything together)

3. Suppose you want to transmit the following 12-bit message "110010011111".

    a. Suppose you want to protect the message from errors using a **checksum** by dividing the message into 4-bit chunks and performing 1s complement addition. Show the EDC bits that will be transmitted. Show your work.

Answer:



    b. Suppose you want to protect the message from errors using **one-dimensional even** parity. Assume that the message is divided into 4-bit chunks for parity computation purposes. Determine the message that would be transmitted. Show the parity bits for each chunk.

Answer:

Original: 110010011111

4-Bit Chunks: 1100 1001 1111

One dimensional even parity means we want the number of ones in each chunk to be even and since its not 2d we just focus on the rows.

1100 **0**

1001 **0**

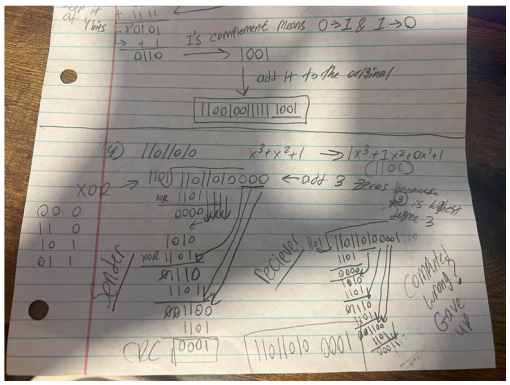
1111 **0**

Combine them: 1100**0**1001**0**1111**0**

4. Suppose you want to transmit the message 11011010 and protect it from errors using the CRC polynomial $x^3 + x^2 + 1$. Use polynomial long division to determine the message that would be transmitted. Assume no bit errors occur during transmission. How does the receiver know that the frame was received without any errors?

Answer: The sender computes the CRC by doing the division above. Then it sends the data.

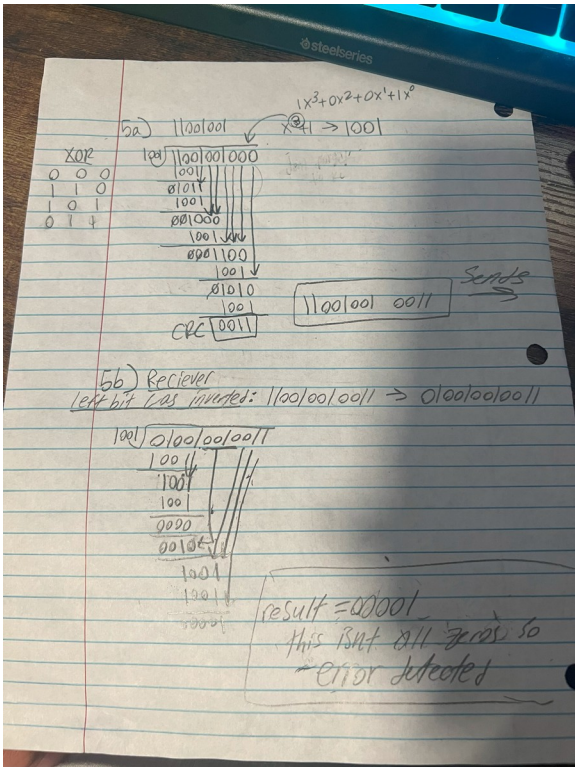
The receiver does long division with the same CRC polynomial on the new frame that was sent (the one with CRC included). If it gets zeros then its correct if not then there was an error. Receiver does not add any zeros based on degree it should just divide.



5. Suppose you want to transmit the message 11001001 and protect it from errors using the CRC polynomial x^3+1.

a. Use polynomial long division to determine the message that would be transmitted.

Answer:



b. Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What's the receiver's CRC calculation? How does the receiver know that an error has occurred?

Answer: Answer above

c. Describe the advantages and disadvantages of using a checksums instead of CRC for error detection and correction at the link layer?

Answer:

Advantage: It is easier to implement compared to CRC. Doesn't add as many bits as CRC. Found to be enough in ARPANET by experience.

Disadvantage: Less reliable error detection and not as accurate as CRC.

6. Suppose you want to transmit the following 4-byte message "0xfb 0x7e 0x7d 0xff" with 0xfb being transmitted first.

a. Assume that the link layer uses byte-counting for framing and has a one-byte length field in the header. Show the bytes transmitted on the wire as hexadecimal numbers.

Answer:

The message is 4 bytes, the length field is 1 byte. Length field will be 5 to indicate the total length of frame

4 + 1 = 5 bytes

5 = 0x05 length message

On the wire: **0x05** 0xfb 0x7e 0x7d 0xff

b. Assume that the link layer uses byte-stuffing, using 0x7e as the frame delimiter and 0x7d as the escape character. Show the bytes transmitted on the wire.

Answer:

Ox7e and 0x7d are delimiter and escape characters so we have to add an escape character when they appear and add the delimiter to the front and back of the message.

Message: 0xfb 0x7e 0x7d 0xff

Add Delimiter: **0x7e** 0xfb 0x7e 0x7d 0xff **0x7e**

Add Escape Character: 0x7e 0xfb **0x7d** 0x7e **0x7d** 0x7d 0xff 0x7e

Final: **0x7e** 0xfb **0x7d** 0x7e **0x7d** 0x7d 0xff **0x7e**

c. Assume that the link layer uses bit-stuffing, stuffing an extra 0 bit after 5 consecutive 1 bits. Show the bits transmitted on the wire clearly marking the stuffed bits. Recall that a link that uses bit- stuffing still uses 0x7e as the frame delimiter.

Answer:

Convert hexadecimal message into binary then add a 0 when you see 5 consecutive 1's

0xfb = 1111 1011 = 11111 **0** 011

0x7e = 0111 1110 = 011111 **0** 10

0x7d = 0111 1101 = 011111 **0** 01

0xff = 1111 1111 = 11111 **0** 111

Combine: 11111**0**01101111**0**10 011111**0**0111111**0**111

Then add the delimiters to the front and back:
0111 1110 11111**0**01101111**0**10 011111**0**0111111**0**111 **0111 1110**

7. Consider a broadcast link L1 containing hosts A and B. Further consider another link L2 containing nodes C and D. Answer the following questions:
    a. Assume L1 and L2 are attached together with a **hub**. Does host A now need to compete with hosts C and D to gain control of the link? Why or why not? What's your answer if L1 and L2 are attached by a **bridge**?

Answer: Yes, because hubs broadcast the data so if A wanted to send C but D wants to send to B, they would compete due to it being a hub. If it was a bridge it would not have to compete because the bridge would allow the data to transfer over to the destination and doesn't broadcast like hubs.

    b. Assume L1 and L2 are attached together with a **bridge**. Assume A sends a packet to B immediately after the links are attached. Does the bridge forward the packet to link L2? Why or why not?

Answer: No, it doesn't because a bridge doesn't broadcast like hubs so if you send within your link it just travels within that link. Bridges create separate collision domains for their links and A and B are on the same link.

    c. Assume L1 and L2 are attached together with a **bridge**. Assume A sends a packet to B immediately after the links are attached. Further assume that B immediately replies back to A. Does the bridge forward the packet to link L2? Why or why not?

Answer: No for the same reason as B. If they are not sending over the bridge there is no point of L1's packets to be sent to L2.

8. Some network applications are a better match for an Ethernet, some are a better match for an FDDI (token ring) network. Which network would be better match for a remote terminal application (e.g., Telnet) and which would be better for a file transfer application (e.g., FTP)? Give a general explanation

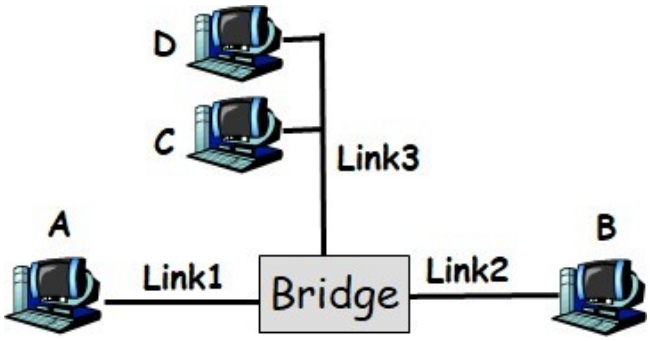
for what it is about each of these applications that suggest that one type of network is better match than the other?

Answer:

Remote Terminal Application: Since Telnet typically involves low bandwidth and bursty traffic (i.e., short, interactive sessions), Ethernet is a better match. Ethernet provides fast access to the network when traffic is light and doesn't suffer much from the collision issue in a light-traffic environment. Telnet requires lower latency for quick responses, which Ethernet handles well in a typical local area network (LAN).

File Transfer Application: FDDI offers consistent, predictable data transmission rates, which is ideal for file transfers. The token-passing mechanism ensures that the network isn't bogged down by collisions, making it more reliable for high-throughput, sustained data transfers. FTP benefits from the predictable bandwidth and higher capacity of FDDI, especially when transferring large files over long periods of time.

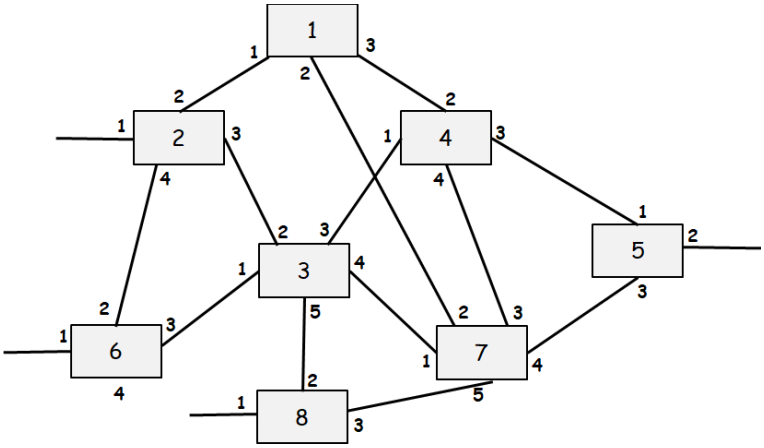
9. Consider the following LAN consisting of 3 links attached by a **bridge**. For each of the following cases, describe which links does the bridge forward the packet to and show the bridge forwarding table after the packet is sent.

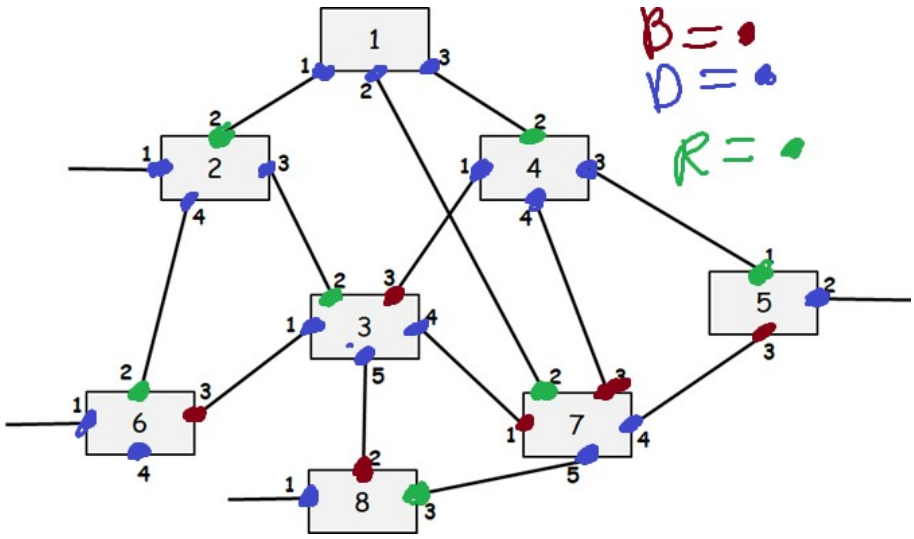


The bridge forwarding table only list the devices to the links as they are discovered so if you never sent a packet to B it would remain unknown but A is discovered so A would be listed with link 1.

a. A sends a packet to B.
Answer: Link1, Link2
b. C sends a packet to B.
Answer: Link3, Link2
c. A sends a packet to C.
Answer: Link1, Link3
d. B sends a packet to D.
Answer: Link2, Link3
e. C sends a packet to D.
Answer: On the same link so they wont access the Bridge

10. Consider the following LAN consisting of 7 bridges numbered 1-7 with the given connections. Each interface of each bridge is also labeled starting with 1. Run Perlman's Spanning Tree Algorithm and for each interface of each bridge write down whether the interface is a "Root port (**R**)", a "Designated port (**D**)" or "Blocking Port (**B**)" in the tables given below. Finally, show the final spanning tree.





Answer: (Also look at tables below)

Elect the Root Bridge:

- The bridge with the lowest Bridge ID becomes the root bridge. If all bridges have unique IDs, the bridge with the lowest number is selected as the root. In this case, Bridge 1 is the root.

Select the Root Ports:

- Each non-root bridge selects one port, called the **Root Port**, that has the lowest cost path to the root bridge. The cost is determined by the number of hops or link costs to the root.

Select the Designated Ports:

- For each LAN segment (link between bridges), the bridge with the lowest cost to the root bridge is selected as the **Designated Bridge**, and the port used for that connection is called the **Designated Port**. If two bridges have the same cost, the one with the lowest Bridge ID becomes the designated bridge.

Block the Remaining Ports:

- Any port that is neither a root port nor a designated port is placed in the **Blocking state**

to prevent loops.

Since 1 is the lowest id number it will be the root port.

This means bridges 2, 7, 4 have root ports at 2, 2, 2.

This takes care of bridge 1 since it has no other paths

Bridge 2: Port 1 allows devices on it to go to the root. Port 2 is a root port because it leads to the root port. Port 3 is a designated port because Bridge 3 goes to the root bridge through bridge 2. Port 4 is a designated port because it allows B6 to go to the Root.

Bridge 3: P1 is a designated port because devices on it still need to go to the root but it doesn't connect on to B6 because B6 can go through B2. P2 is root because it goes through B2 to root. P3 is Blocking because the smaller id to the root is B2 and not B4. P4 is designated and blocked for the same reason B6. Same for P5.

Continue: this continues with the other ports and bridges. If its port leads to the root or to a bridge with a lower id than another that leads to the root it can be selected. Port from becomes root port while the destination port becomes the designated port. Designated ports are also allowed so that hosts in that wire can still traverse but then the other end has to be blocked. Refer to picture above its kinda hard to explain correctly.

Bridge 1:

Interface	Type
1	D
2	D
3	D

Bridge 2:

Interface	Type
1	D
2	R
3	D
4	D

Bridge 3:

Interface	Type
1	D
2	R
3	B
4	D
5	D

Bridge 4:

Interface	Type
1	D
2	R
3	D
4	D

Bridge 5:

Interface	Type
1	R
2	D
3	B

Bridge 6:

Interface	Type
1	D
2	R
3	B
4	D

Bridge 7:

Interface	Type
1	B
2	R
3	B
4	D
5	D

Bridge 8:

Interface	Type
1	D
2	B
3	R