Problem 1

1. Ethernet has a period of silence, which separates each frame. Without a minimum packet size, we no longer require a length field. Ethernet originally requires a length field in case we need to pad a message that is shorter than the minimum packet size. Without a length field, we will not know which bits the padding and which bits are the actual data. However, Nethernet will not pad packets because there is no minimum packet size and does not require a length field.
2. Chart, line chart

   Description automatically generatedNo, this rule is no longer valid in Nethernet. In Ethernet, we must discard packets of length less than 64 bytes because the minimum packet size is 64 bytes. However, in Nethernet, there is no minimum packet size and packets of length less than 64 bytes are valid.
3. Consider stations A and B as shown in the figure to the right. A is trying to send a small packet (< 64 bytes) to B and B is trying to send a large packet (>> 64 bytes) to A. While B is sending the large packet, the small packet that A sends also arrives. According to the new rule given in the problem, a station waits and detects possible transmissions only when the packet size is less than 64 bytes. Therefore, in this case, B would not detect any collisions if we did not implement normal means of error detection in Nethernet.

Diagram

Description automatically generated

1. We want all nodes on the network (even those not sending or receiving) to detect when collisions happen. Consider the scenario in the figure on the right. A is sending a small packet to B and B is sending a small packet to A at the same time. Both A and B will detect collisions because of the new rule of Nethernet (A and B both detect transmissions within the 51.2 usec period). On the other hand, there is no voltage spike at node C. Also, node C is not sending any packets, so it is not waiting for 51.2 usec.

Diagram

Description automatically generated

1. Consider the figure to the right where A is sending to C and B is sending to A at the same time. When A first sends to B, B will receive the clean packet and store it. It will not detect any collisions, so it will not expect retransmissions. However, A will detect a collision while waiting 51.2 usec after sending its small packet (it detects a signal from C). So, A will retransmit its message to B. B will receive the second packet, which is a duplicate.

Problem 2

1. Both bridges will map MAC address V to the upper interface. The packet will go to all three LANs (LANs 1, 2, 3). When B1 receives the packet, it will map V to upper interface and flood (i.e., send the packet to the other interface). So, the packet will be on LAN 2. When B2 receives the packet, it will map V to the upper interface and flood (i.e., send the packet to the lower interface). Then, the packet will be on LAN 3 and the victim V will receive the packet.
2. When S replies to V, the packet will not go to the real victim. B1 will receive S and will look at the destination MAC address V. It will look at its table and see that V is mapped to the upper interface. Thus, B1 will not transmit the message to the lower interface. To pick up the packet, H must be listening for packets whose destinations are MAC address V.
3. Below is the sequence of sends:
   1. **H sends a packet to S with forged source address V.** Both bridges will flood and send the packet to their lower interfaces. Both bridges will also map V to their upper interfaces.
      1. B1 bridging table: V 🡪 Up
      2. B2 bridging table: V 🡪 Up
   2. **V sends a packet to S**. Both bridges will flood and send the packet to their upper interfaces. Both bridges will relearn the entry for V and map it to their lower interfaces.
      1. B1 bridging table: V 🡪 Down
      2. B2 bridging table: V 🡪 Down
   3. **S sends a reply to V.** Both bridges will send the packet to their lower interfaces since a mapping for V exists. They will also learn S and map it to their upper interfaces. The reply from S will end up on all LANs.
      1. B1 bridging table: V 🡪 Down, S 🡪 Up
      2. B2 bridging table: V 🡪 Down, S 🡪 Up
   4. **Both H and V receive the reply since it is on all LANs.**
   5. **V sends a RESET to S.** Both bridges are mapping S to their upper interface, so the RESET message will end up on all LANs. Bridging tables are not changed.
   6. **S receives the RESET.**
4. To detect and report such attacks, a bridge could keep track of and look at how much time has elapsed between learning V was on LAN X and relearning that V is now on LAN Y. If it is impossible for a human to disconnect V from LAN X and connect it to LAN Y in that time period, then there is a high possibility of an attack.

Problem 3

1. When a bridge receives a message and floods (because the destination address is not stored in the bridging table), then the bridge will send the packet to all interfaces except: the interface that the message comes from and the interface that is mapped to the source address of the message (if it is stored in the bridging table).

In the given example, when the counterclockwise packet reaches B2, although the bridge floods, it does not send the packet to any of its interfaces because:

1. The packet is coming from the lower interface
2. The source address of the packet (S) is mapped to the upper interface

There are no other interfaces (other than the upper and lower interfaces) to flood to, so the bridge stops circulating the packet.

Similarly, when the clockwise packet reaches B1, we do not send the packet to any interface.

1. What exactly is Alyssa’s method? Is her method the one we came up with for part a?

Problem 4

What exactly is a broadcast storm?? Flurry of messages?

What does it mean to forward a packet?

1. Below are the sequence of events:
   1. D broadcasts an ARP request on the LAN requesting A’s MAC address (given A’s IP address)
   2. E receives the ARP broadcast, sees that the IP address is different from itself, and forwards the packet. (Isn’t this IP address in the data message, so would the nodes also forward this packet)
   3. The bridge receives both ARP requests (the original and the forwarded message). Both will have a source MAC address of D and a destination of all 1s (broadcast). First message will create a mapping from D’s MAC address to the upper interface. The first message will flood and transmit to bottom interface. Similarly, second message will also transmit to bottom interface.
      1. Bridging table: D 🡪 Up
   4. B and C receive the messages on the LAN. For each of the 2 messages, they will forward another message because the IP address is different from their IP addresses. There are 4 broadcast messages on the bottom LAN.
   5. A receives the 4 broadcast messages and will send 4 ARP replies to D that contains A’s MAC address.
   6. The bridge receives the ARP replies. It maps A’s MAC address (all 1s) to the lower interface. The destination (D) is stored in the bridging table, so it transmits the messages to the upper interface.
      1. Bridging table: D 🡪 Up, A 🡪 Down
   7. D receives the ARP replies on the LAN
   8. D puts the message packet with destination MAC address of all 1s on the data link
   9. All nodes on the data link (E, A, B, C) will read this packet since the MAC address of all 1s is the broadcast address
2. Assume there are a total of T nodes on a network. Assume a node on a LAN broadcasts an ARP request. There are at most M nodes on this LAN. The ARP broadcast will be picked up by a router connected to this LAN and will not be transmitted to the other LAN. When the other M – 1 nodes on this LAN receive the ARP broadcast and see that the IP address in the message defers from their own, they will forward the packet. Now, only M nodes in the network are experiencing a broadcast storm due to the forwarding protocol (instead of all T nodes as described in the previous problem).

SHOULD WE KEEP THIS PROBLEM GENERAL??

What does forward mean? Do we duplicate messages on the same LAN

What is the problem that gets better?? I don’t really get this question. I don’t see how broadcast storm is applicable here.

D sends ARP req to router. Router sends A’s MAC to D. D sends message with A’s MAC (all 1s). Router forwards to other LAN. Now there is a broadcast on all links… Do end nodes forward (as per the problem) messages that are broadcasts?

Bridge table maps MAC to Interface

2. H must look for packets with destination of V?

3. Find a sequence of sends and receives that result in V picking up the packet

Multicast and unicast MAC addresses. All nodes are listening for specific addresses

4.

Problem 3

1. When an unknown destination is seen by bridge, we flood (i.e., send to all other interfaces).

Multicast: A sends to both B1 and B2. B1 sends to B3. B3 sends to B2, so B2 gets the same packet again. The original packet B2 got will go to the lower right LAN. Then, B3 will send it to B1 again.

We have two packets circulating (one clockwise and one couterclockwise) moving through the system very fast.

Solution (maybe): If we are sending to (src, dest) at bridge, also do not send to the interface of src if we have it saved in our bridging table

Problem 4

Masking IP address to only look at the prefix:

IP & mask, where mask == 11111111 111111111 0000000000 0000000000 &

127. 1 2 3

==. 127 1 0 0

A thinks its MAC address is all 1s == FF ; FF ; FF ; FF ; FF ; FF

Broadcast addrtess is FF FF FF FF FF FF, so all nodes will pick up this destination address

Address resolution protocol -> ARP, which maps IP addresses to MAC addresses

1. Hint: it causes a broadcast storm (read problem)

E.g., T = 5, M = 3

But keep T and M general when answering the question