Home exam in 4320

Task 1)

IPv4 and MIP (Minimal interconnection protocol)

Our MIP implementation is different from IPv4 in several ways. First of all, our MIP implementation is much simpler than the implementation of IPv4, which results in lower overhead, faster transmission and makes it more suitable for smaller networks. Consequently this means IPv4 is more suitable for larger and more complex networks, where IPv4 offers more address spaces and a more robust way of sending information with additional features. Another difference is that our MIP has a flat addressing model where every node has a unique address, which saves us the trouble of subnetting or routing.

Our MIP only communicates between directly connected nodes while IPv4 can operate over vast and complex global networks. IPv4 provides better scalability and flexibility which is advantageous.

In conclusion both protocols have their advantages and disadvantages where MIP is simpler, faster and has lower overhead while IPv4 is more suitable for scalability, offers more features and is robust.

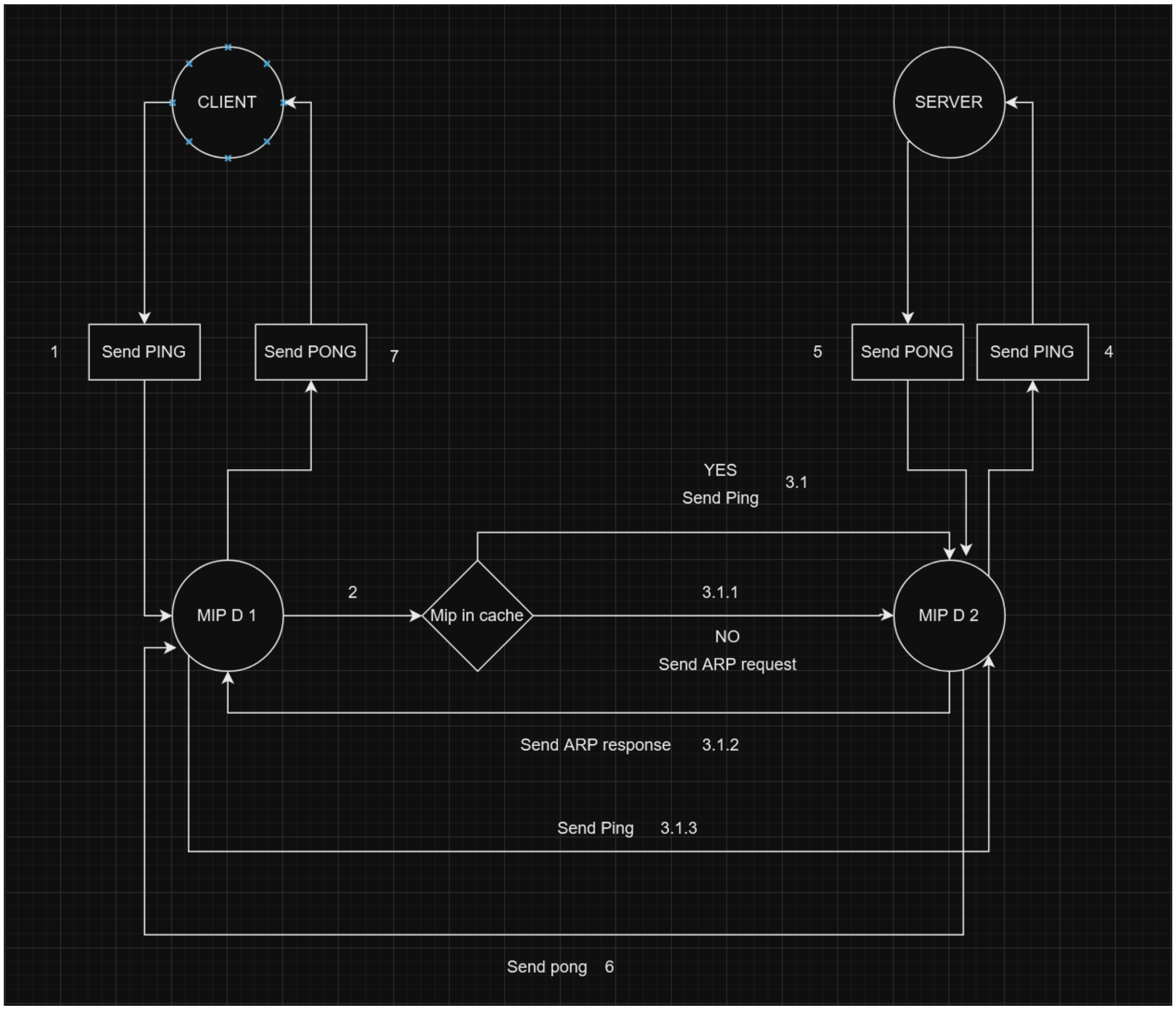
Task 2)

MIP-ARP protocol

Because we operate on the ethernet level our main concern is to map the MIP addresses of our nodes to their respective mac addresses. We implement the MIP-ARP protocol in our MIP daemon because it is essential to have created these maps in order to communicate. When our MIP does not have a MIP address in the cache it sends a broadcast message to each connected node containing our destination MIP address. Then two nodes get information about each other. This results in a lot of nodes having to process the request for no apparent reason.

One alternative implementation would be to send a broadcast message to all neighboring nodes straight away, and store all the replies, mapping every node from the start. This would result in more complexity early, but less complexity if the program persists. Based on how the program is being used this could be advantageous.

The main reason why we need to handle the MIP-ARP as a special case is because it interacts with both the link and the network layer, sort of in between, needing both ethernet mac addresses and MIP addresses. Therefore we have implemented it directly in the MIP where it automatically solves this problem and map the connections.



Explanation:

1. Client sends ping message to MIPD 1 over unix socket (MIP address & message)
2. MIPD 1 receives packet, checks if MIP address is in cache

3.1 MIP address is in cache, construct PDU and send to MIPD 2 over raw socket

3.1.1 MIP address is not in cache, we send an ARP request

  3.1.2 MIPD 2 receives a ARP request and responds with an ARP response, both MIPs update their cache

  3.1.3 MIPD 3 sends the ping message

4. MIPD 2 sends the ping message to the server over unix socket
5. The server constructs a pong response message and sends it back to MIPD 2
6. MIPD 2 now has the MIP address of MIPD 1 stored and sends pong message back
7. MIPD 1 receives the pong message, and sends the message to the client

8. The client assesses the pong response checking if the message is equal to the original message and prints the result if so, if not it prints an error message.