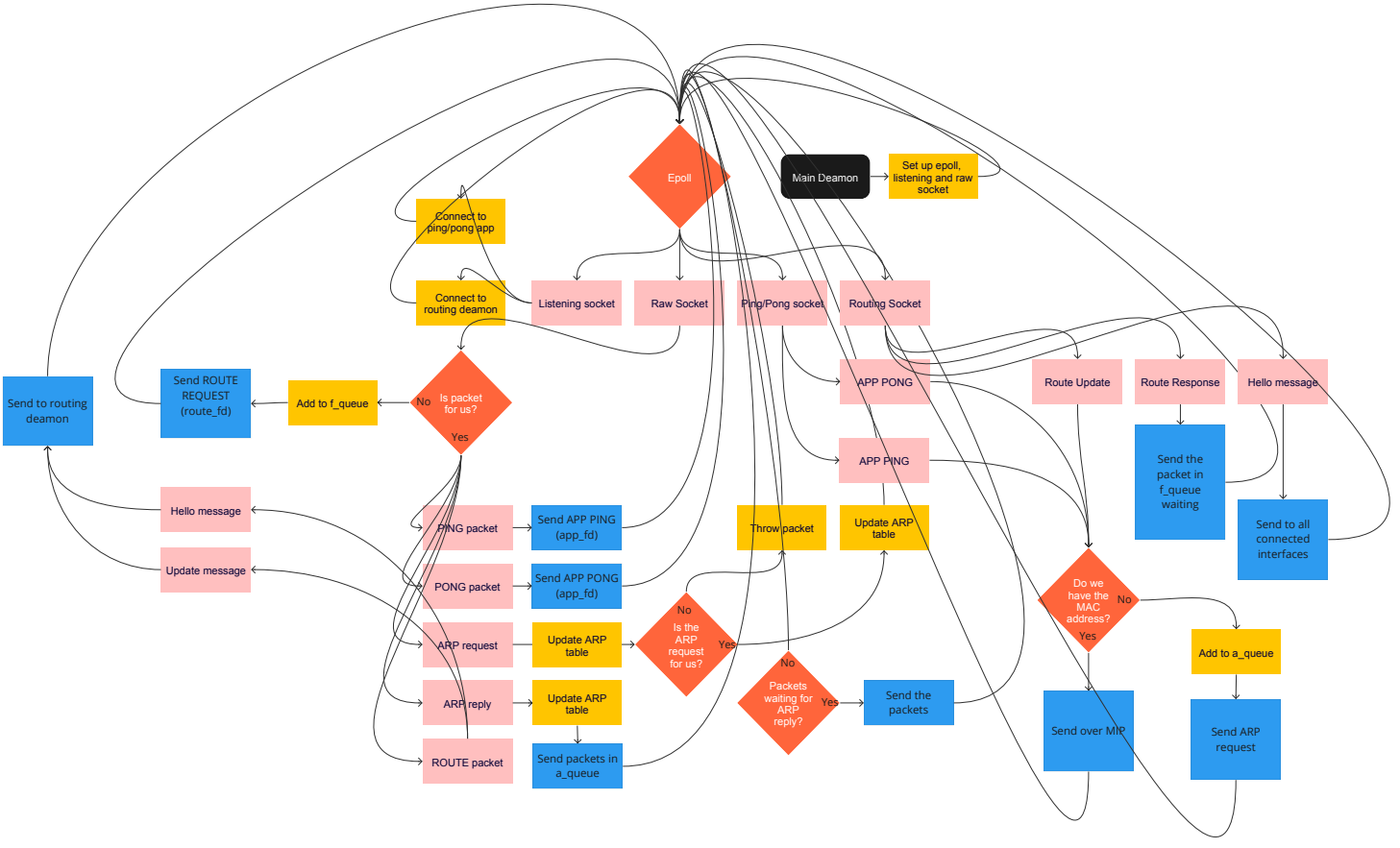
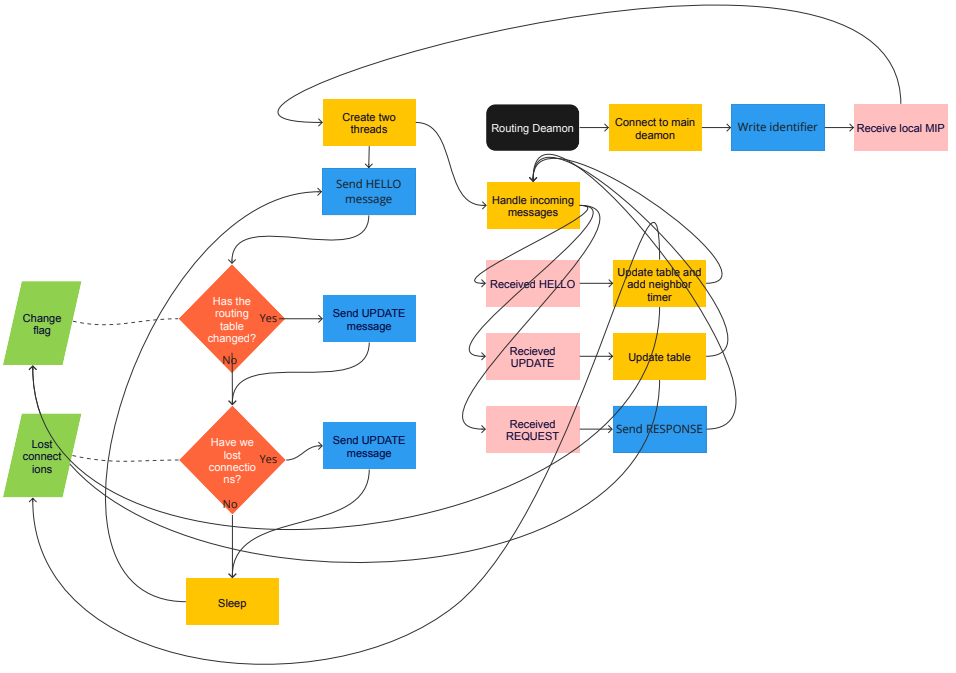
Home Exam 2
IN4230

Candidate 150010

Fall 2023


Epoll
Main Deamon
Set up epoll, listening and raw socket
Connect to ping/pong app
Connect to routing deamon
Listening socket
Raw Socket
Ping/Pong socket
Routing Socket
Send to routing deamon
Send ROUTE REQUEST (route_fd)
Add to f_queue
No
Is packet for us?
Yes
APP PONG
APP PING
Route Update
Route Response
Hello message
Send the packet in f_queue waiting
Send to all connected interfaces
Hello message
Update message
PING packet
Send APP PING (app_fd)
PONG packet
Send APP PONG (app_fd)
ARP request
Update ARP table
ARP reply
Update ARP table
ROUTE packet
Send packets in a_queue
Throw packet
Update ARP table
No
Is the ARP request for us?
Yes
No
Packets waiting for ARP reply?
Yes
Send the packets
Do we have the MAC address?
No
Yes
Add to a_queue
Send over MIP
Send ARP request


Create two threads
Routing Deamon
Connect to main deamon
Write identifier
Receive local MIP
Send HELLO message
Handle incoming messages
Change flag
Has the routing table changed?
Yes
No
Send UPDATE message
Received HELLO
Update table and add neighbor timer
Recieved UPDATE
Update table
Lost connect ions
Have we lost connectio ns?
Yes
No
Send UPDATE message
Received REQUEST
Send RESPONSE
Sleep

Count-to-Infinity Problem in DVR Protocols

The "Count-to-Infinity" problem in Distance Vector Routing (DVR) protocols refers to a situation where incorrect routing information circulates endlessly within a network. This issue arises when a link failure occurs, and the routers continuously increase the metric (like the number of hops) for a certain route, believing that the destination is still reachable through a longer path. This leads to routing loops and increased network traffic.

To handle this issue, one common approach is to implement a maximum limit on the metric. For instance, setting a maximum hop count beyond which a route is considered unreachable. This prevents the metric from increasing indefinitely. Another method is to use more sophisticated algorithms like Split Horizon or Poison Reverse, which help to prevent incorrect information from spreading.

Limitation of Split Horizon and Solution with Poison Reverse

Split Horizon is a technique used in DVR protocols where a router is prevented from sending information about a route back in the direction from which it was learned. However, this method has limitations, especially in networks with more complex topologies. It may fail to prevent routing loops in cases where more than one router is involved in the loop.

Poison Reverse is an enhancement to the Split Horizon technique. It addresses its limitations by not only preventing information about a route from being sent back in the direction from which it was learned but also actively marking the route as unreachable (often by setting the metric to an infinite value) when sending updates back in that direction. This helps in quicker convergence and resolution of incorrect routing information, effectively mitigating the risk of routing loops in more complex network scenarios.

My routing protocol

When a packet needs to be sent, whether it originates from an application or another routing daemon, it is first added to a queue for processing. Simultaneously, a routing request is dispatched to the routing daemon responsible for finding the appropriate route. Upon receiving a response from the routing daemon, the packet is forwarded along the discovered route, assuming a valid route is found.

The routing daemon, in turn, operates by periodically sending hello messages to its neighboring nodes. These hello messages serve as a form of communication to maintain the network's status and connectivity. If hello messages stop being received from a particular neighbor, the routing daemon takes action by removing the non-responsive neighbor from its list of active connections and subsequently updating the routing table to reflect the change in network topology. This process ensures the network's robustness and adaptability in the face of changing conditions.