



SIMPLE ROUTER - project 2b

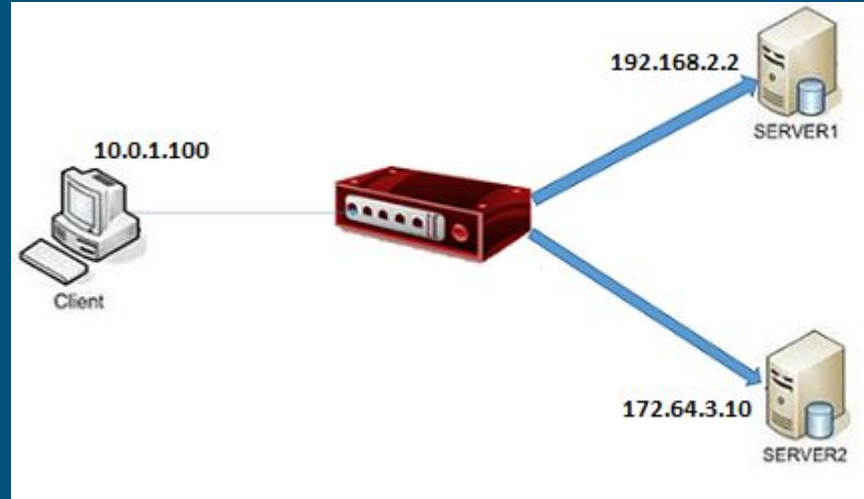


Roger Cheng



TOPOLOGY

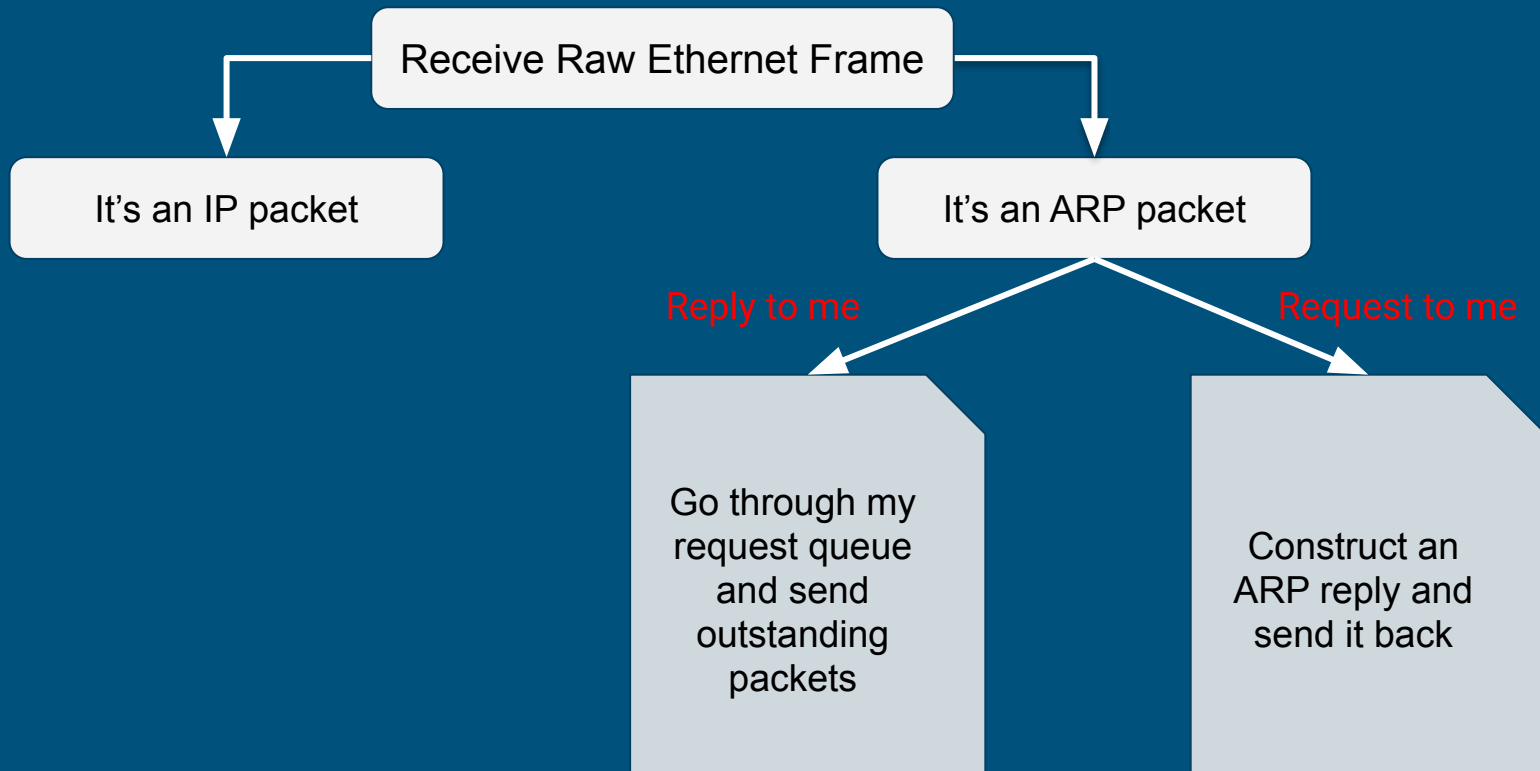
- Github repository has been setup with skeleton code for this project and a VM will be provided with suitable environment settings.
- Again, Coding will done in C.
- The topology is setup using IP_CONFIG and rtable files. Please check them out.



Router interfaces:

```
eth3    HWaddr c2:e3:78:bf:71:f9
        inet addr 10.0.1.1
eth2    HWaddr 26:e4:1f:26:49:ab
        inet addr 172.64.3.1
eth1    HWaddr 56:01:ec:fb:34:09
        inet addr 192.168.2.1
```

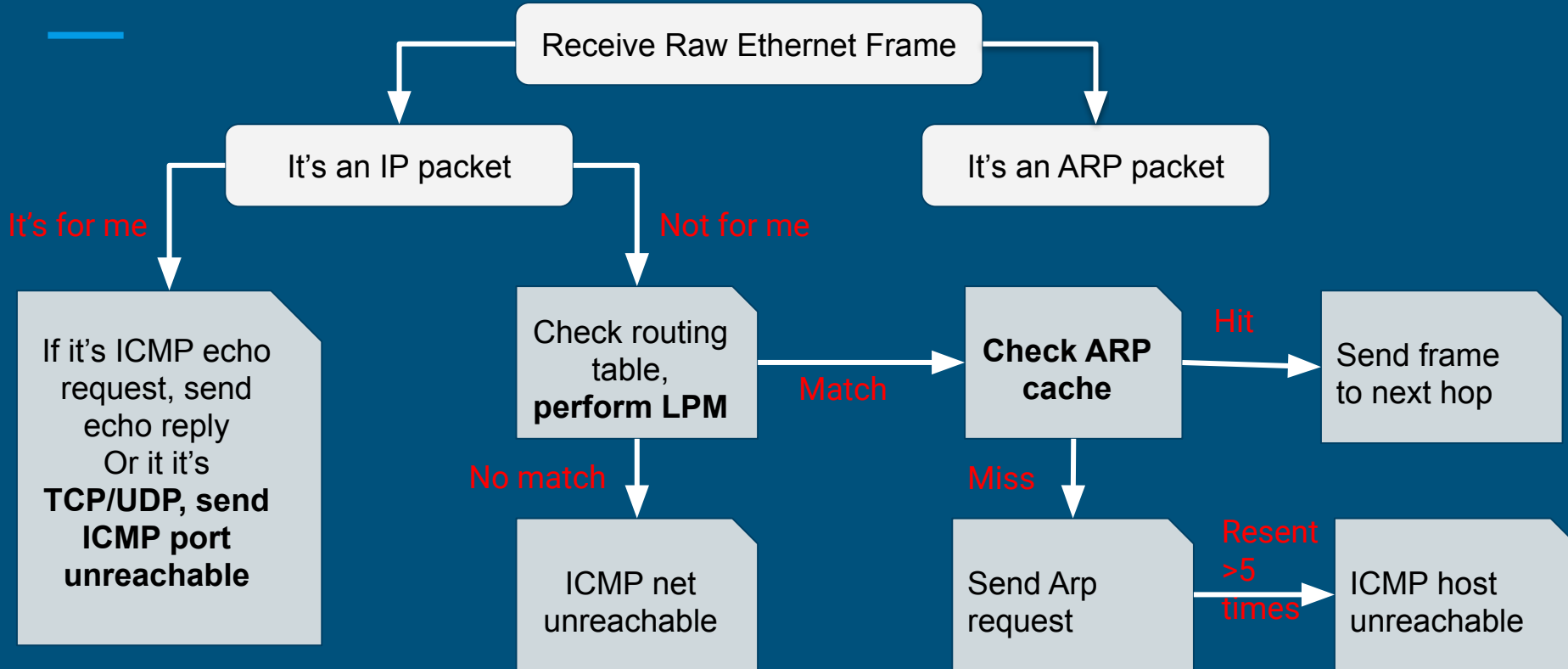
ARP FLOW



IF IT'S AN ARP PACKET

- Verify the length of the packet.
- Use `sr_get_interface` (in `sr_if.c`) to get the interface record. Get the mac address of the destination IP address/interface IPaddress from the record.
- Check the 'opcode' variable of the ARP header and see if it is `arp_op_request` or `arp_op_reply`.
- If it is an ARP Request update all the fields of the packet and use `sr_send_packet()` to send an ARP reply.
- If it is a Reply, send all the packets in the queue to the destination.

IP FLOW



IF IT'S AN IP PACKET

- Check the length.
- Validate the IP header.
 - Should not be IPv6
 - Check ip_hl
 - Check ip_len
 - Checksum

IF IT'S AN IP PACKET - continue

- Check if it is destined to you, the router.
 - Check ip_p.
 - If it is ICMP echo request (type 8), then generate ICMP echo reply (type 0).
 - **Router should not handle non-ICMP packets (tcp or udp). Otherwise generate ICMP port unreachable (type 3, code 3).**

IF IP PACKET IS NOT DESTINED TO THE ROUTER

- Check `ip_ttl`. If $TTL \leq 1$, send ICMP time exceeded (type 11, code 0).
- Look up next-hop address by doing **LPM** on the routing table using the packet's destination address. If it does not exist, send ICMP net unreachable (type 3, code 0).
- If it does exist, then reduce `ttl` and update checksum.
- From next-hop address, determine outgoing interface and next-hop MAC address
- If necessary, send ARP request to determine MAC address
- Encapsulate IP datagram in Ethernet packet
- Forward packet to outgoing interface

CREATING ARP REQUESTS

- If LPM entry (type `sr_rt`) is found, then reduce TTL and update checksum for the IP header. Now, you need to update the frame header's source and destination fields
- Do a `sr_arpcache_lookup`. '`gw.s_addr`' (next hop IP address) is one of the variables to be passed to the function.
- If it returns NULL, use `sr_arpcache_queuereq` function to add the ARP request to the ARP request queue. Send the `sr` instance and the queue to `handle_arpreq`. `Handle_arpreq` - Function to be implemented. Check comments in `sr_arpcache.h` for pseudocode.
- If the lookup returned an arp entry, then modify the Ethernet source and destination values and use `sr_send_pac`

LPM – LONGEST PREFIX MATCH

- If packet not destined to router and `ttl != 1`, check the routing table to see if a matching entry for the destination IP address exists (LPM)
- You have the routing table (`sr->routing_table`) and destination IP address. The `routing_table` is a structure of type `sr_rt` (defined in `sr_rt.h`). The routing table has `dest` and `mask` variables of type `in_addr`.
- Do `&` (bitwise AND) between `dest` and `mask`. Also, between the destination IP address and `mask`. Compare to see if they match.
- If multiple matches, check to see which match has the longest `mask.s_addr`.

To Test Your LPM LOGIC

- To test your routing table lookup, you can change the entries in rtable file to have different subnet masks.
- For instance, to check your router's longest prefix match logic, you can have multiple entries in your rtable file that match a single destination IP.
- The following routing table has two matching entries for the packet destined to 192.168.2.2

192.168.2.2	192.1.1.1	255.255.255.0	eth4
192.168.2.2	192.168.2.2	255.255.255.255	eth1
172.64.3.10	172.64.3.10	255.255.255.255	eth2
10.0.1.100	10.0.1.100	255.255.255.255	eth3

Questions?
