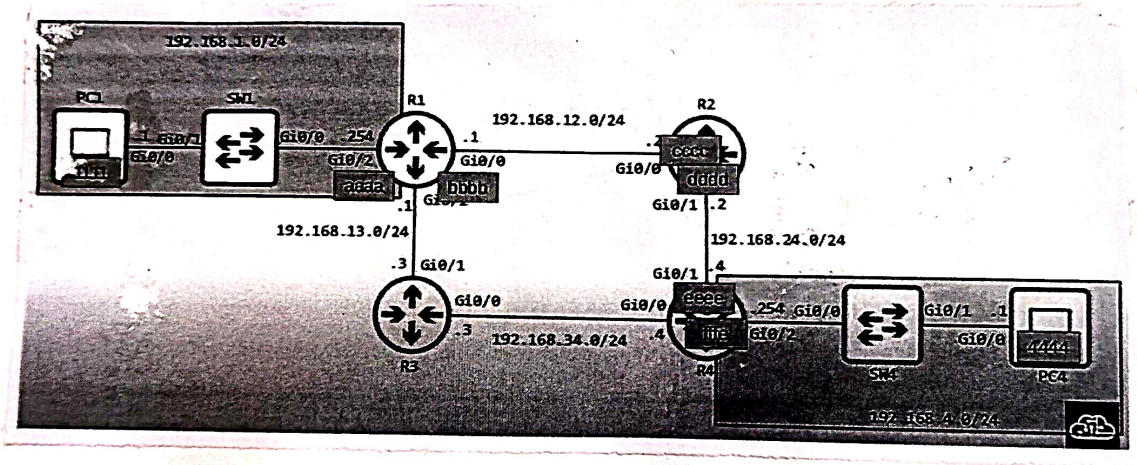
Day-10

Life of a Packet

- The entire process of sending Packet to a remote destination.
- Including ARP, encapsulation, de-encapsulation, etc.

#NETWORK TOPOLOGY



→ A packet being sent from PC1 in the 192.168.1.0/24 network, to PC4 in the 192.168.4.0/24 network.

→ Let's assume we have pre-configured static routes on these devices, so the packet will follow the same path as in the static routing topology.

That is PC1 → R1 → R2 → R4 → PC4

since were, not just looking at Layer 3 in this video Let me add MAC addresses for these devices.

As you know a MAC address is actually 12 hexadecimal oha... but just to save space shorter it 4.

NOTE:- SW1's interface also have MAC addresses, however for this its not necessary to know the MAC addresses of the switches so to avoid clutter.

→ these above things Include in 1st Diagram.

Now PC1 wants to send some data to PC4, & its encapsulated in this IP header.

Src: 192.168.1.1 (PC1)
Dst: 192.168.4.1 (PC4)

PC1 IP address is in the (192.168.1.0/24) Network, it notices that the address (192.168.4.1) is a different network. so it knowns that it needs to send packet to its default gateway.

Which is R1, something we have already preconfigured In this example PC1 has not sent any traffic yet so it needs to use ARP.

Let's look at the ARP process once more.

PC1 MAKES THIS ARP REQUEST :-

ARP REQUEST
Src IP : 192.168.1.1
Dst IP : 192.168.1.254
Dst MAC : ffff.ffff.ffff

→ So, it sends the frame, which SW1 receives & broadcasts out of all interfaces except one it received. (i.e SW1 will forward the frame out its G0/0 interface)

→ When this broadcast frame arrives on R1, it notices that the destination IP is its own, so it create this ARP reply frame to send back to PC1

ARP Reply
Src IP: 192.168.1.254
Dst IP: 192.168.1.1
Dst MAC: 1111
Src MAC: aaaa

The ARP request message was broadcast because R1 learned PC1's IP & MAC add. from the ARP reques message the ARP reply can be sent unicast directly to PC1 So, that's what R1 does.

→ So, Now PC1 knows the MAC address of its default gateway so it encapsulates the Packet with this ethernet header

Src: 192.168.1.1	Dst: aaaa
Dst: 192.168.1.1	Src: 1111

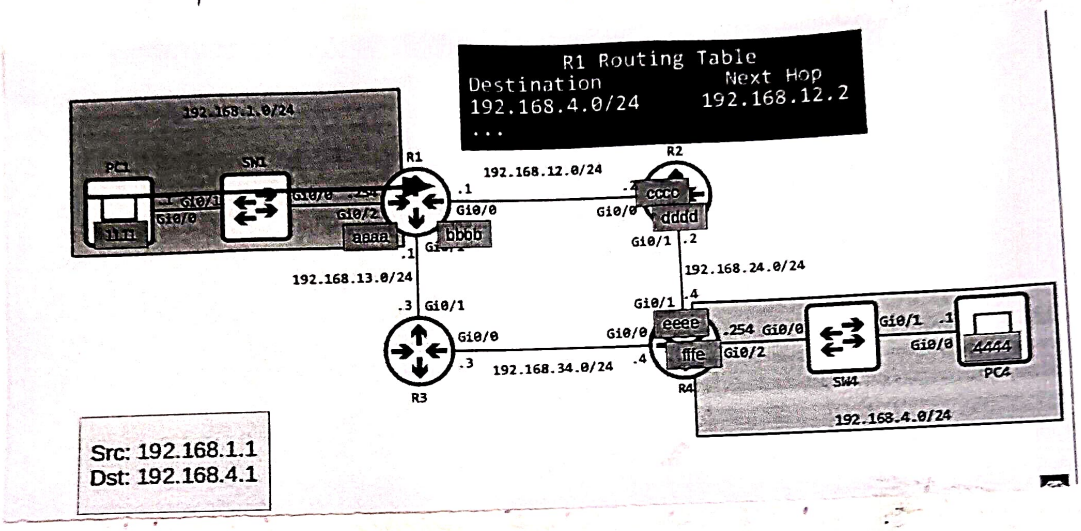
Keep In Mind, the orginal packet is not changed the destination address remains PC4's IP add... NOT R1's IP add...

Only at Layer 2 is the destination set to R1's MAC address

→ So, its send the frame to R1. R1 receives it, and removes the ethernet header.

Src: 192.168.1.1
Dst: 192.168.4.1

It looks up the destination in its routing table & Next hope is 192.168.12.2.



→

→ R1 will have to encapsulate this packet with an Ethernet frame with the appropriate MAC address for 192.168.12.2.

However, R1 doesn't know R2's MAC address yet. So how will it learns R2's MAC address?

It will use ARP Request

ARP Request
SrcIP: 192.168.12.1
DstIP: 192.168.12.2
Dst MAC: ffff.ffff.ffff
Src MAC: bbbb

→ So, it sends the arp request & R2 receives it. R2 receives the broadcast, & since the destination IP address Matches its own IP addres. it make this ARP reply to send R1

ARP Reply
SrcIP: 192.168.122
DstIP: 192.168.12.1
DstMAC: bbbb
SrcMAC: cccc

Once again, because it learned the IP and MAC addresses of R1 from the ARP request it doesn't have to broadcast the frame So, it sends this ARP reply back to R1.

→ Now, R1 knows R2 MAC Address so it can encapsulate the packet with an Ethernet Header.

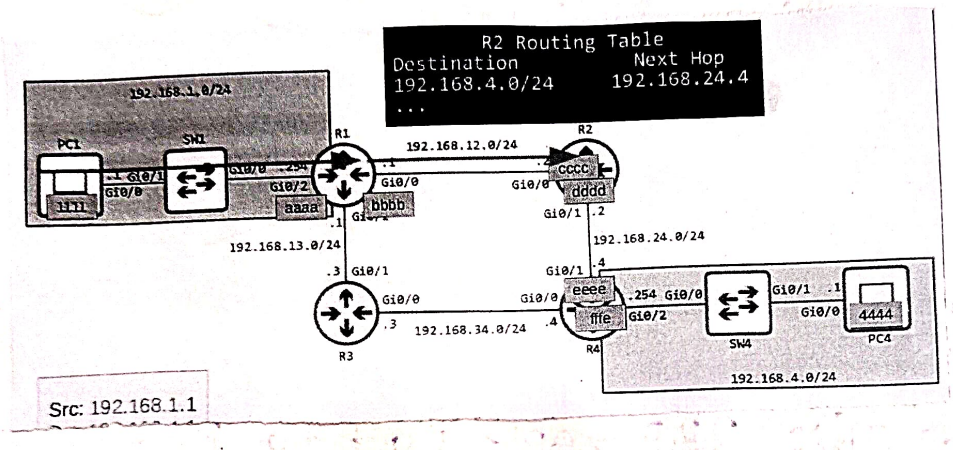
Src: 192.168.1.4	Dst: cccc
Dst: 192.168.4.1	Src: bbbb

Inserting R2's MAC address in the Destination field, and the MAC address of R1's G0/0 interface in the source field. and it sends it to R2.

→ After receiving the frame, R2 removes the Ethernet header.

Src: 192.168.1.1
Dst: 192.168.4.1

R2 looks up the destination in its routing table & Next hope is 192.168.24.4



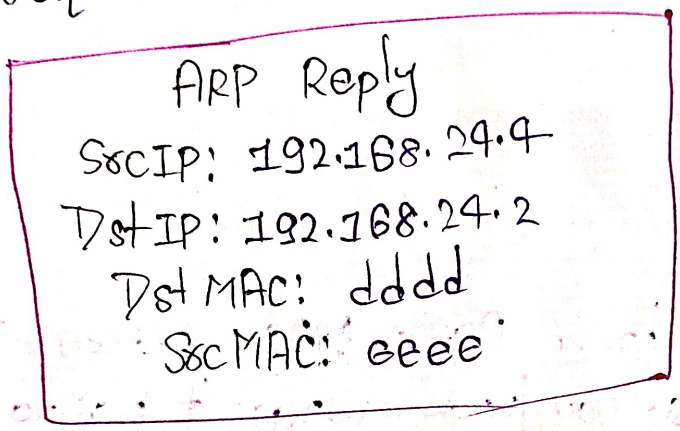
→ Although 192.168.24.0/24 is a Connected network to R2, it doesn't know the MAC address of R4. R2 will use ARP to discover R4's MAC address

ARP Request

Src IP: 192.168.24.2
Dst IP: 192.168.24.4
Dst MAC: ffff.ffff.ffff
Src MAC: dddd

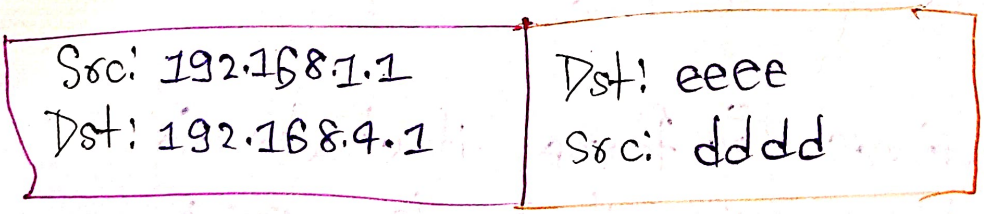
and ARP request forward it out of its G0/1 interface.

→ R4 receives the broadcast, & since the destination IP address is its own it creates this ARP reply to send to back to R2. once again it already knows R2 IP & MAC Addresses because they were used as the source addresses for the ARP request

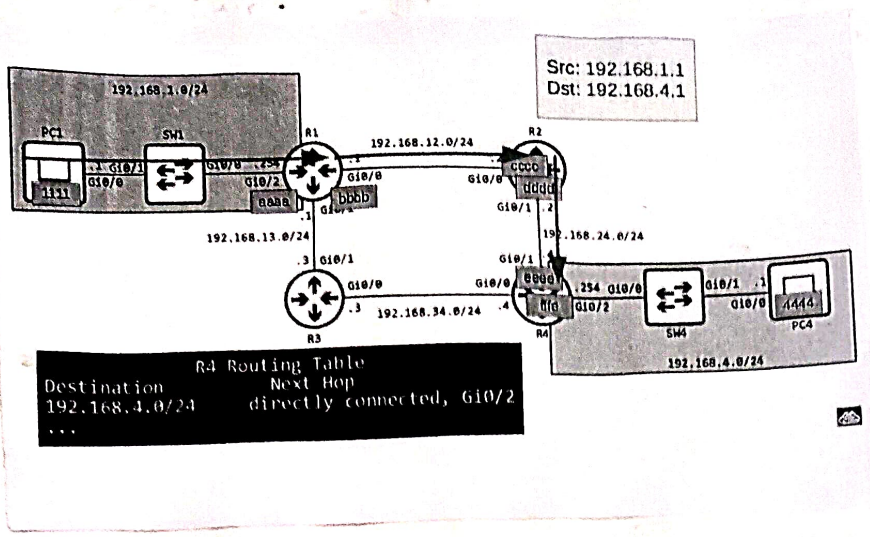


It sends the unicast frame back to R2 with ARP reply.

→ Now that R2 knows R4 MAC address it encapsulates PC1's packet with an Ethernet header



R4 receives the frame & remove the Ethernet header. It lookup the destination in its routing table & most specific 192.168.4.0/24. which is directly connected.



→ Once again R4 doesn't know PC4's MAC address of PC4's MAC address yet so you know whas it has to do next use ARP to know PC4 mac Address

ARP Request
SrcIP: 192.168.4.254
DstIP: 192.168.4.1
Dst MAC: ffff.ffff.ffff
SrcMAC: fffe

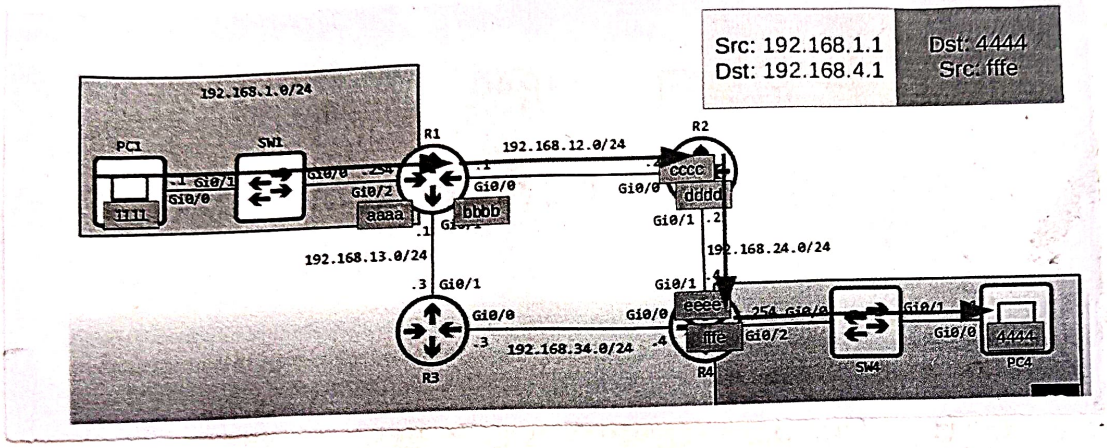
→ After PC4 receives the frame, it checks the destination IP address since it is its own IP address it will send an ARP reply.

ARP Reply
SrcIP: 192.168.4.1
DstIP: 192.168.4.254
Dst MAC: fffe
Src MAC: 4444

→ Now that R4 knows PC4 MAC address, it adds an Ethernet header to the packet

Src: 192.168.1.1 Dst: 192.168.4.1	Dst: 4444 Src: fffe

R4 sends the frame to PC4, & finally it has reached its destination



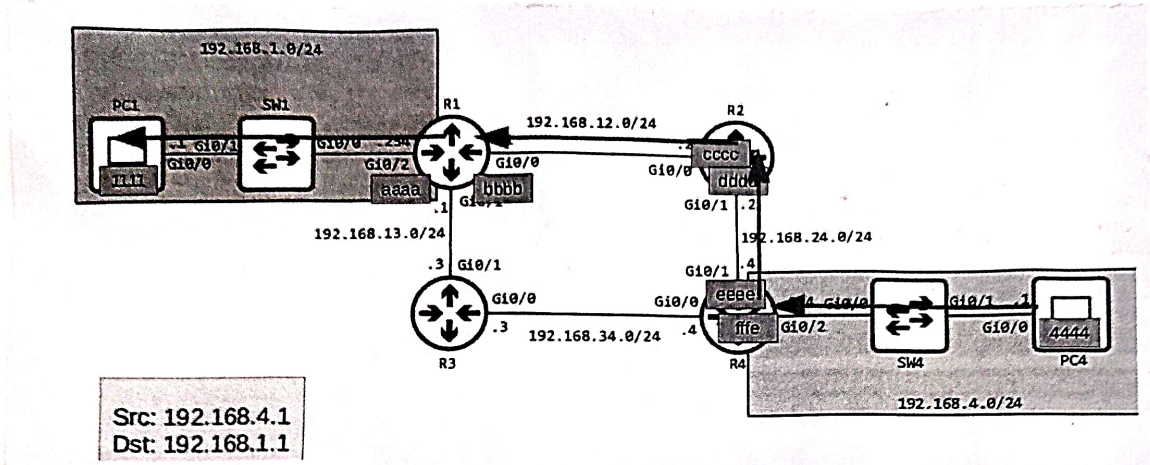
NOTICE, that the orginal packet hasn't changed throughout the process. it same IP header with a Source IP :- 192.168.1.1

Destination IP :- 192.168.4.1

Also, Notice that the switches didn't actually modify the frame at any point. The switches forwarded the frames & learned the MAC addresses, but they don't actually de-encapsulate & then re-encapsulate the packet with a new, header.

Ethernet

Now, let's say PC4 send a reply back to PC1, and we've configured static routes on the routers so, that the traffic follows the same path on the way back to PC1, going via SW4, R4, R2, R1 & then PC1.



What will be different?

→ There will be one major difference, since these devices have already gone through the ARP process, there won't be any need for ARP request & replies the packet will simply be forwarded from device to device being de-encapsulated & then re-encapsulated as it is received by and the forwarded by each routes

So, that it in this, just a basic walkthrough of how a packet is forwarded b/w routers to pass it along to its final destination.