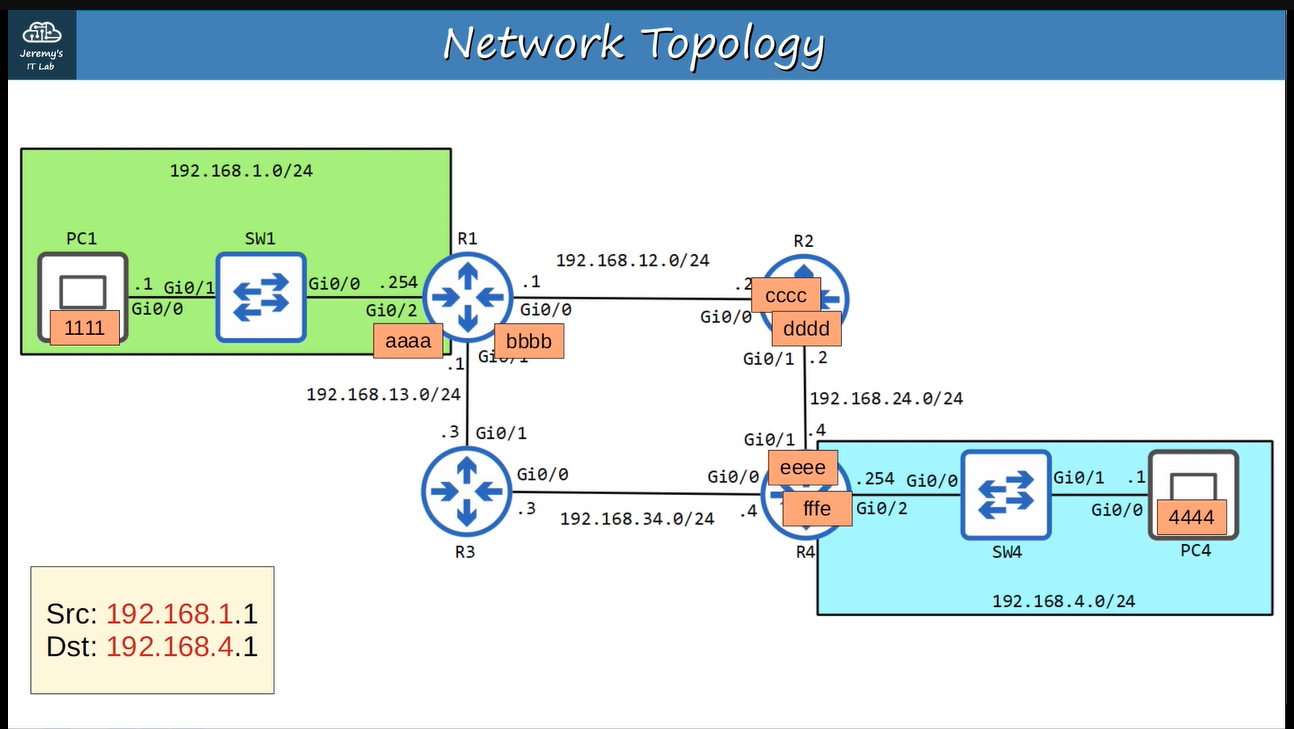
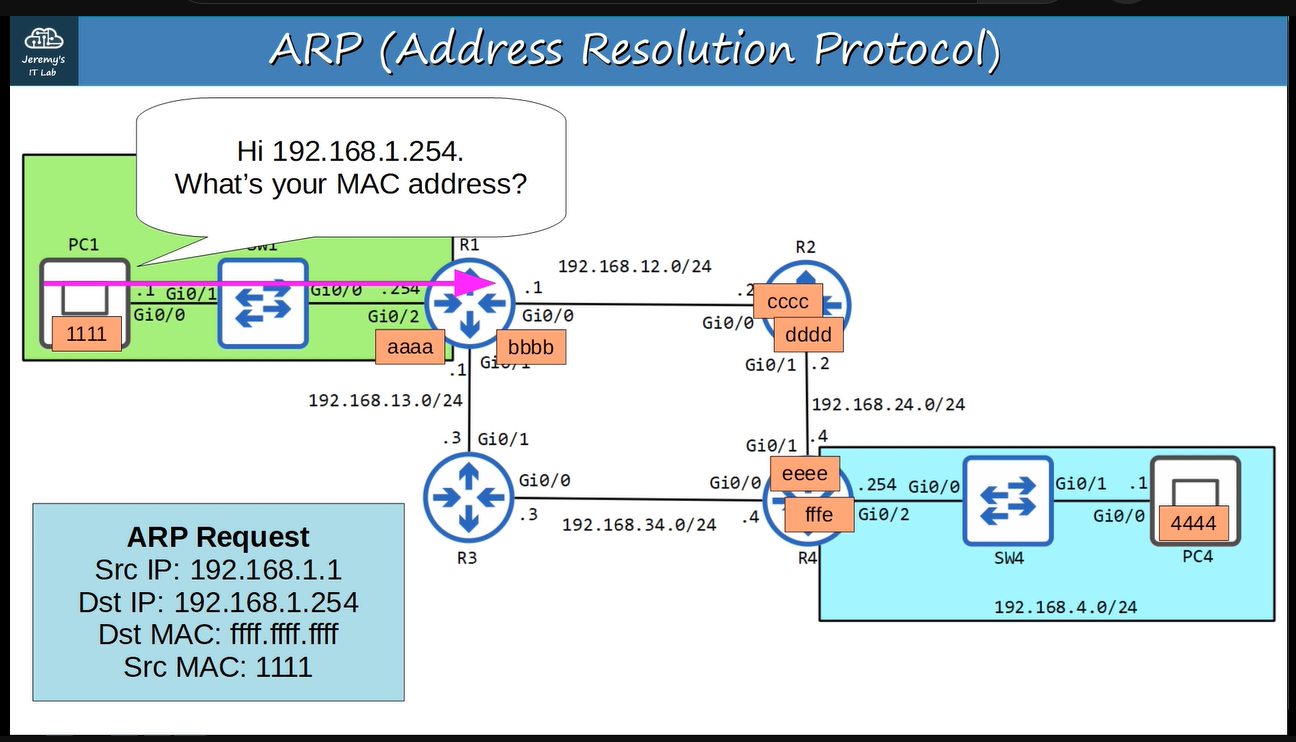
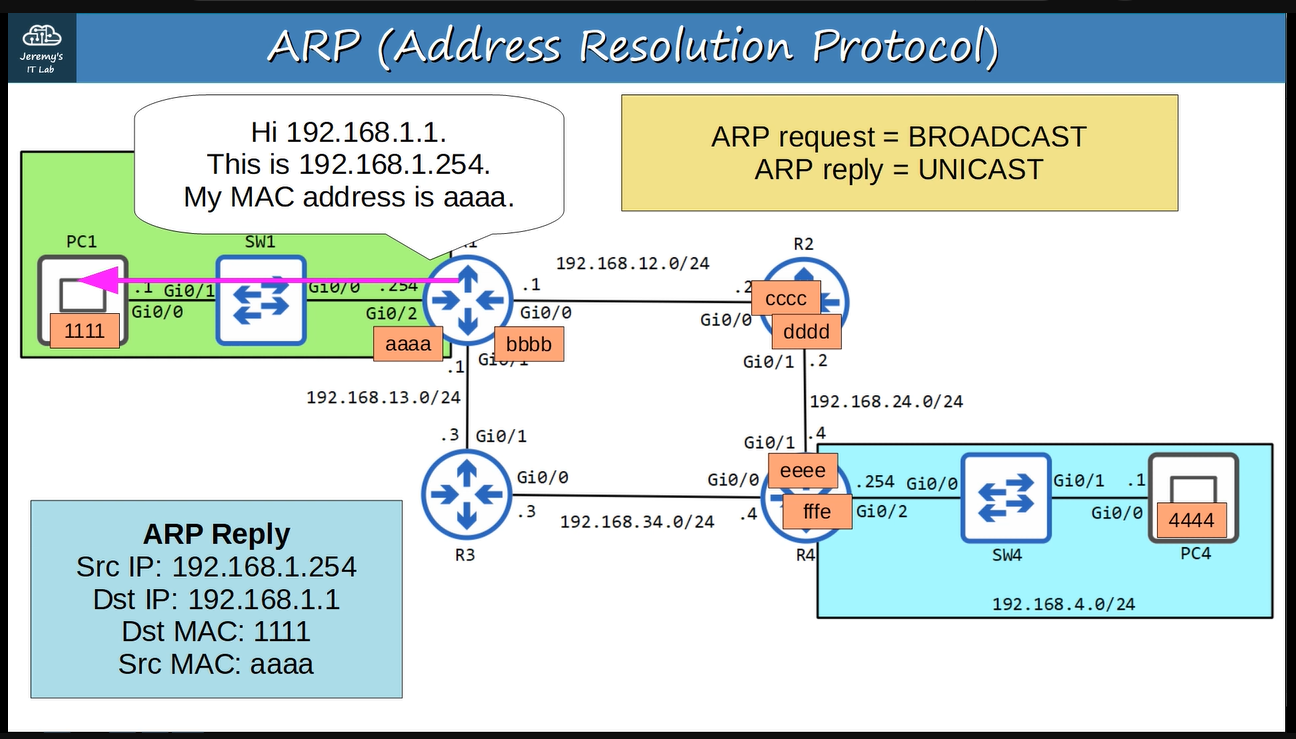
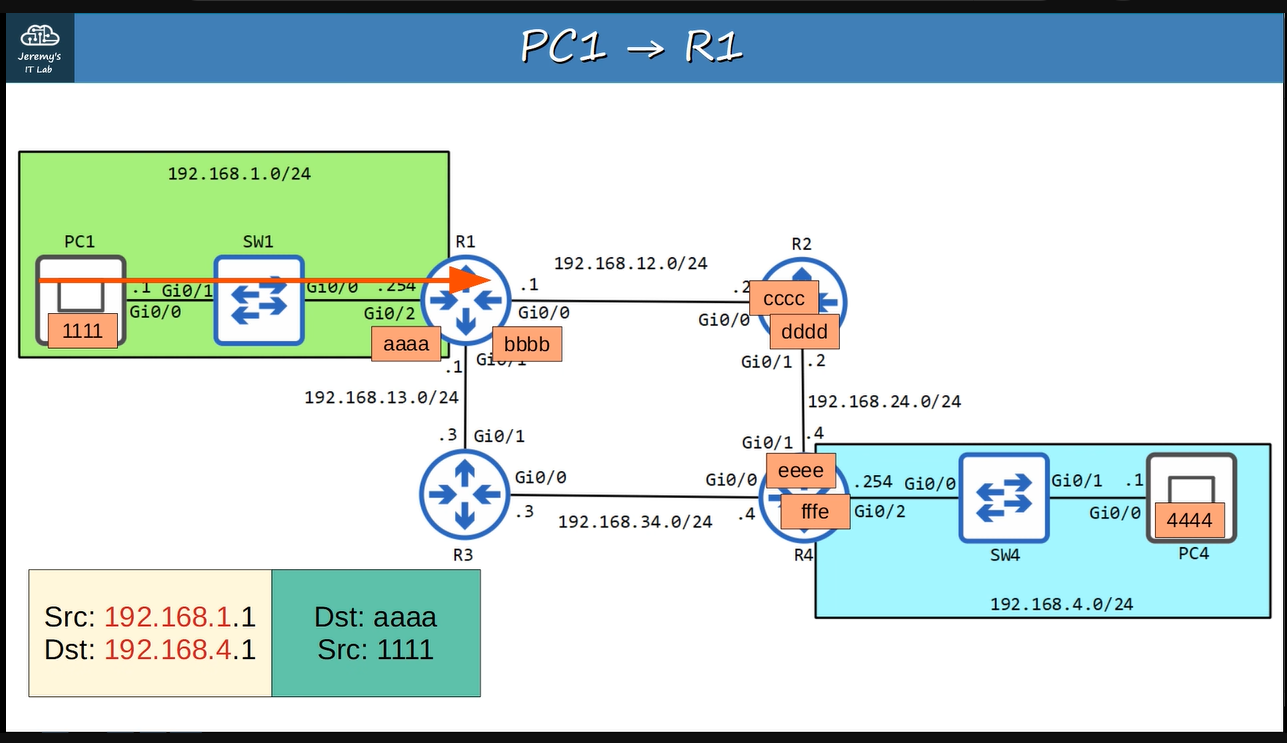
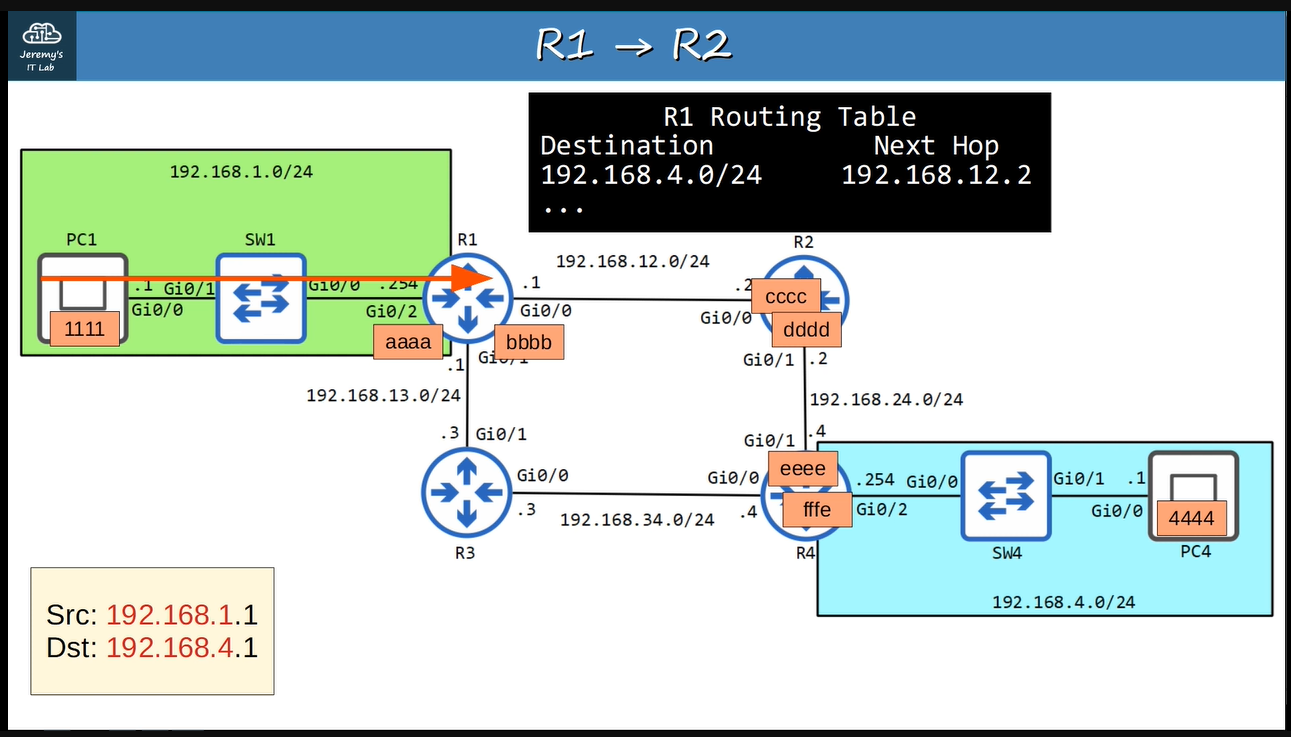
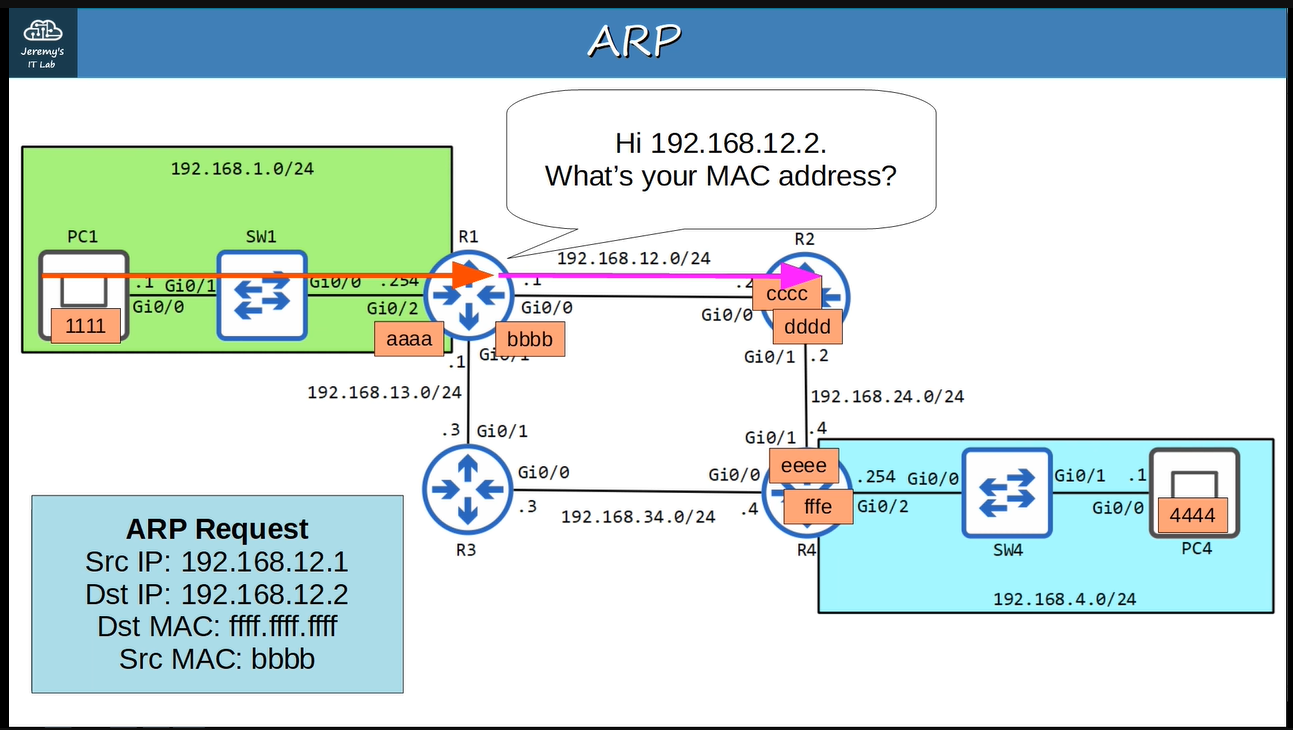
**We will use this topology to understand the travel of a packet from PC1 to PC4 ↓**

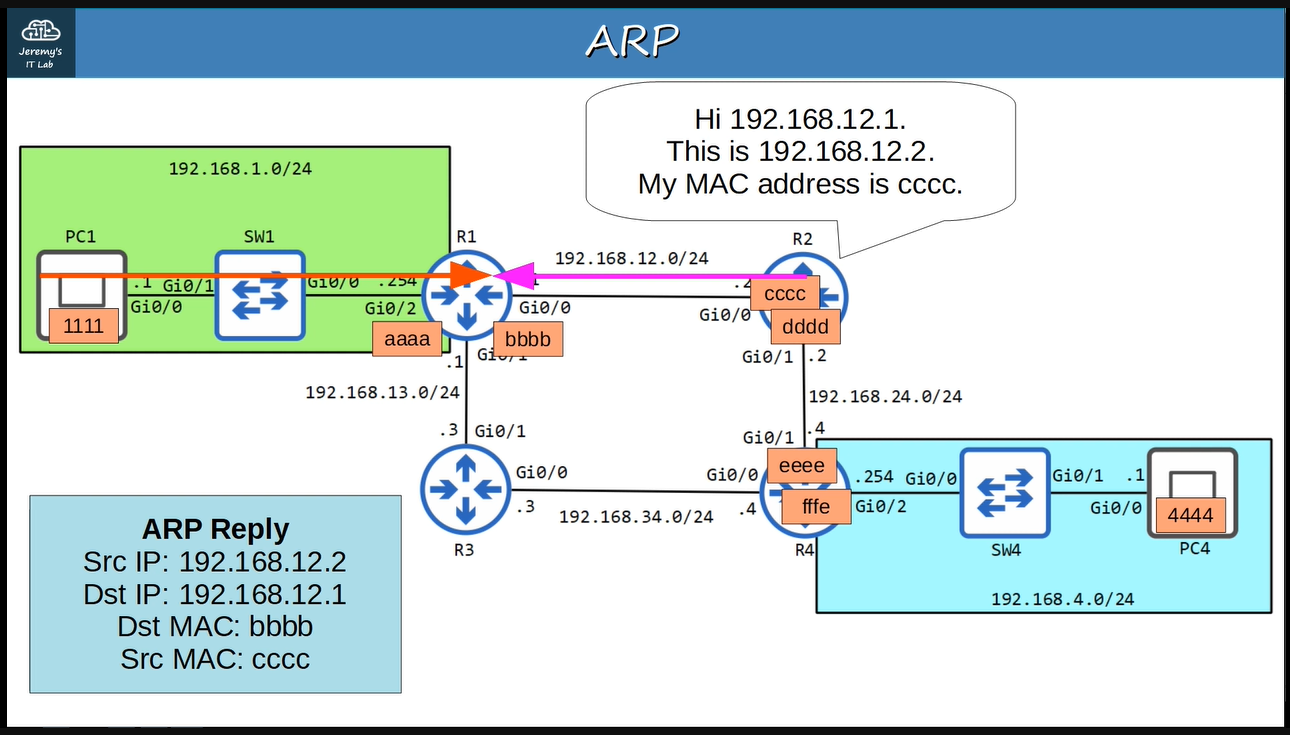
**First PC1 will send ARP-request to R1 to know the MAC address of Gi0/2 ↓**

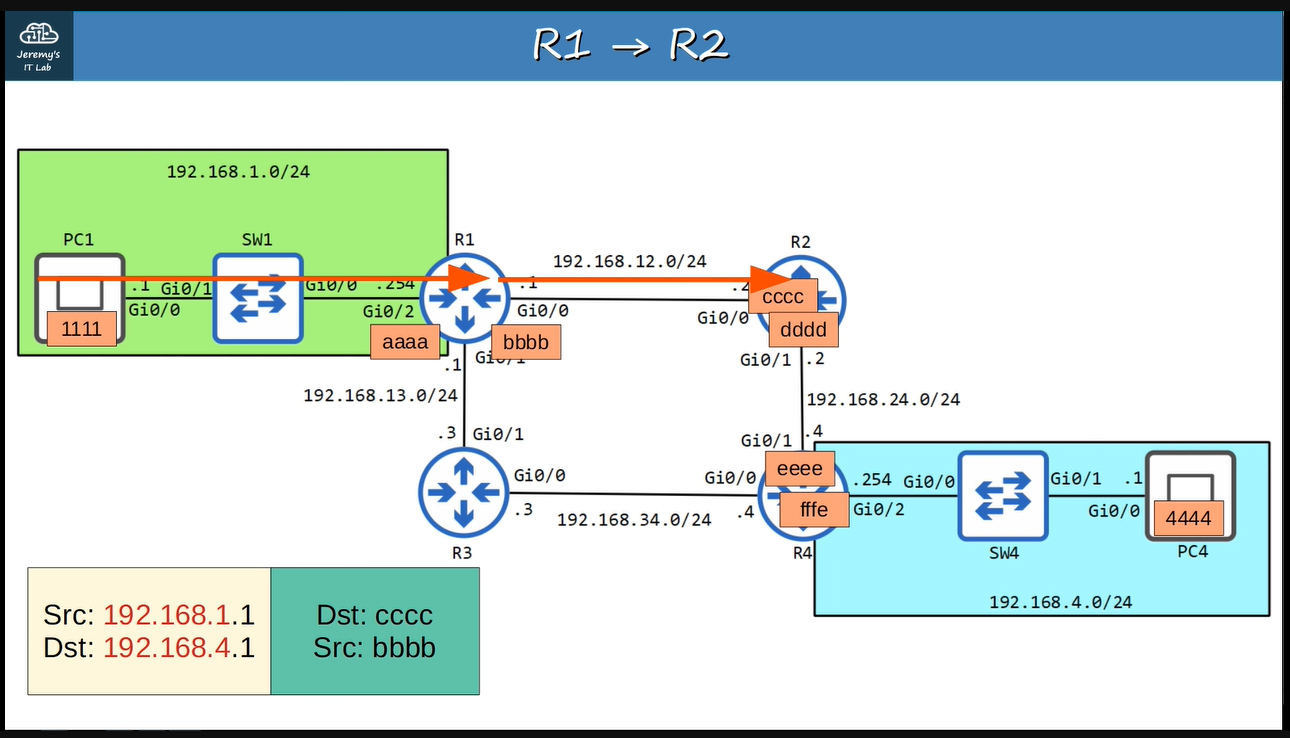
**Then the R1 will send ARP-reply to PC1 ↓**

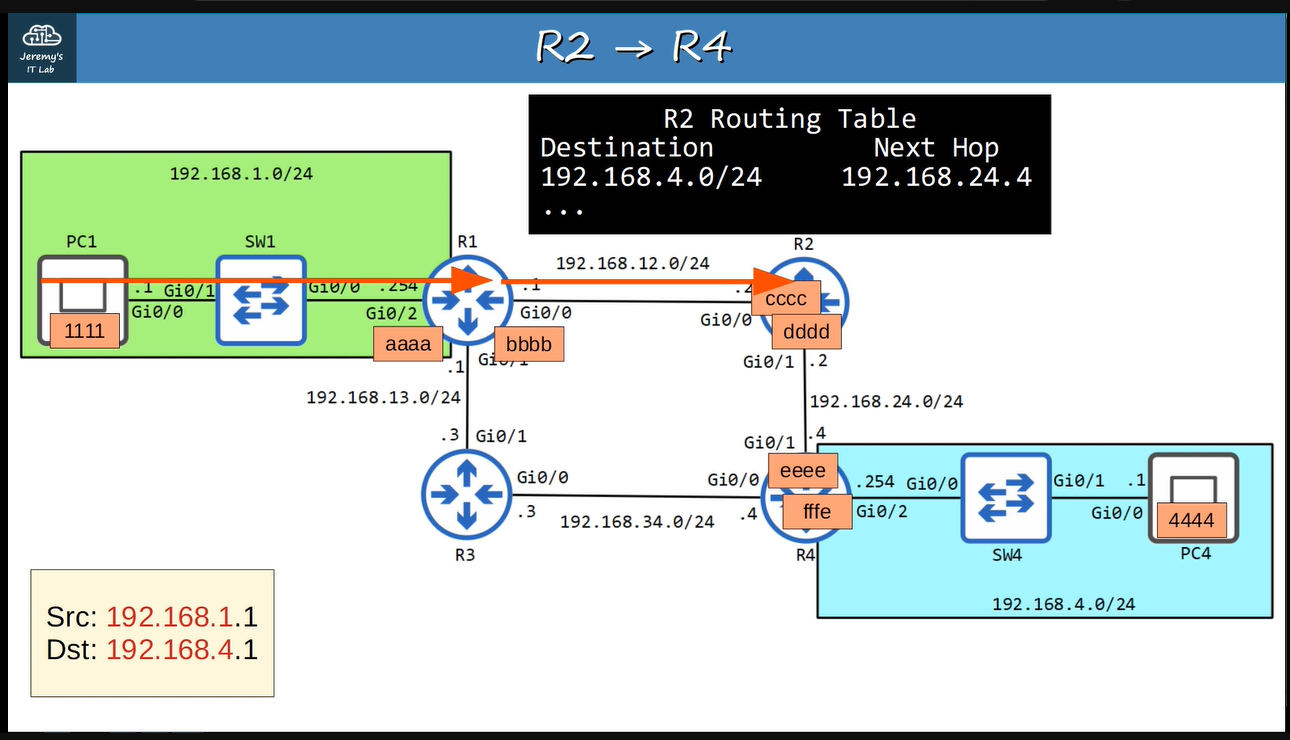
**Now the PC1 will send a frame to R1 at “aaaa” for PC4 ↓**

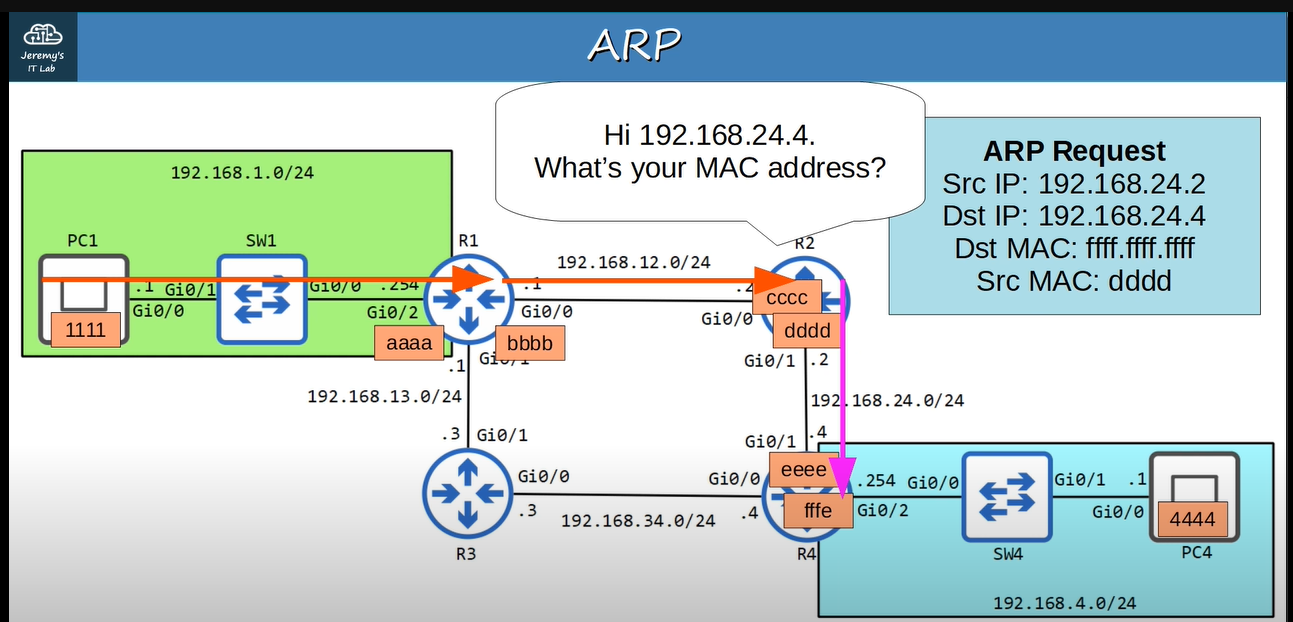
**Now R1 will receive it and de-encapsulate ethernet header and look up the destination in its routing table and next-hop ↓**

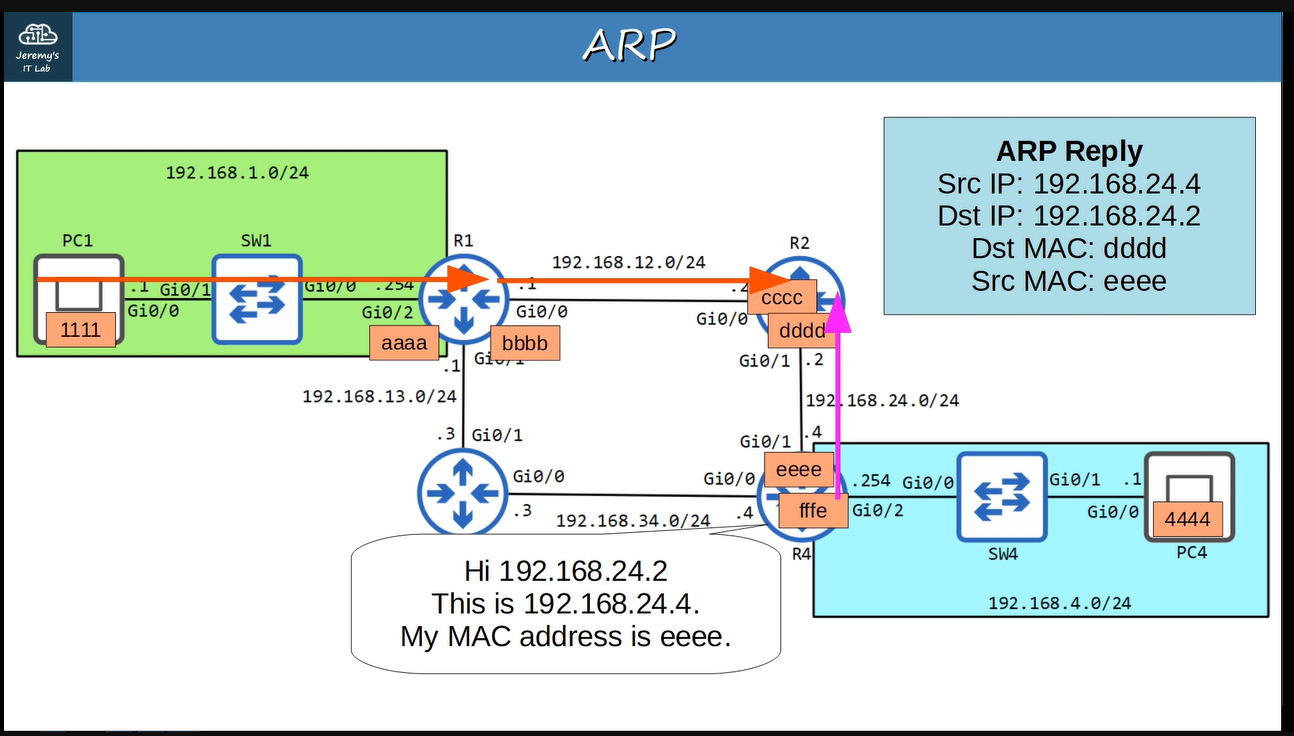
**Now R1 encapsulate this packet with the MAC address for R2’s Gi0/0, suppose it does not know it so it will send ARP-request to R2 to know the MAC address of R2’s Gi0/0 ↓**

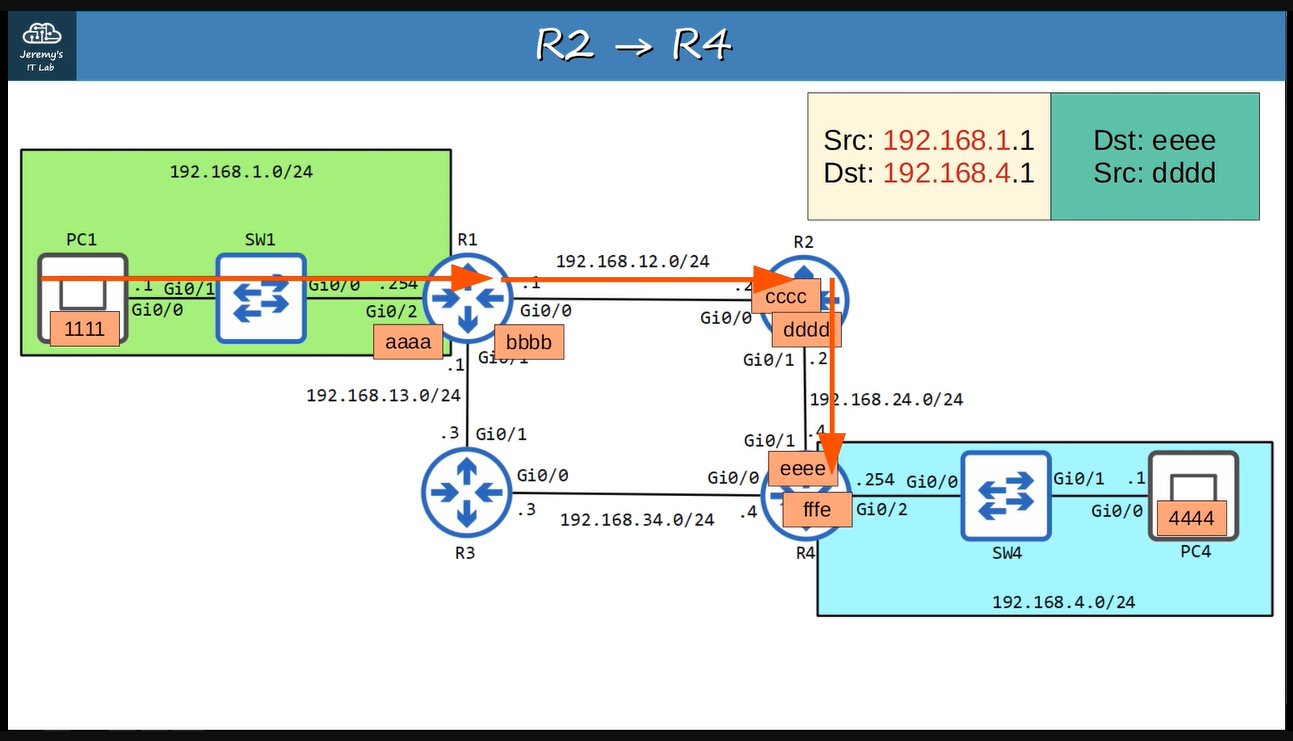
**Then R2 will send ARP-reply to R1 ↓**

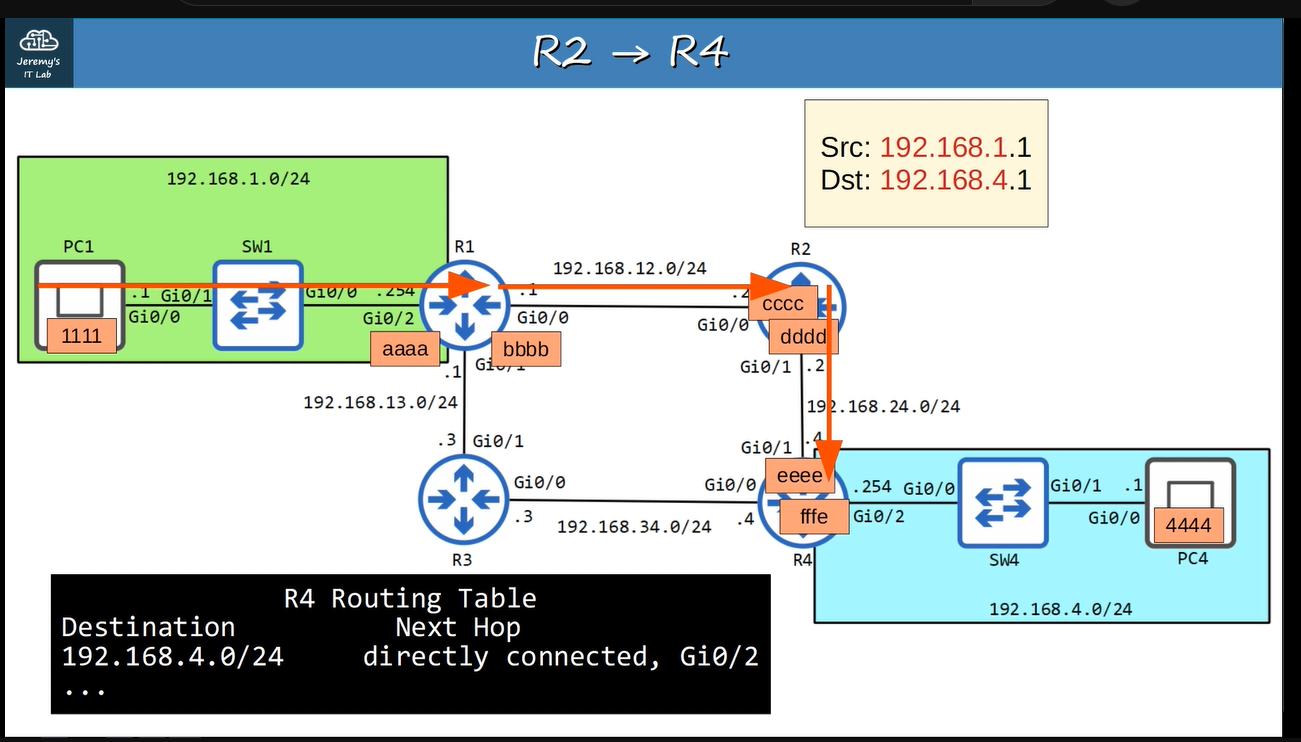
**Now R1 will send that packet to R2 at Gi0/0 with destination Mac of “cccc” ↓**

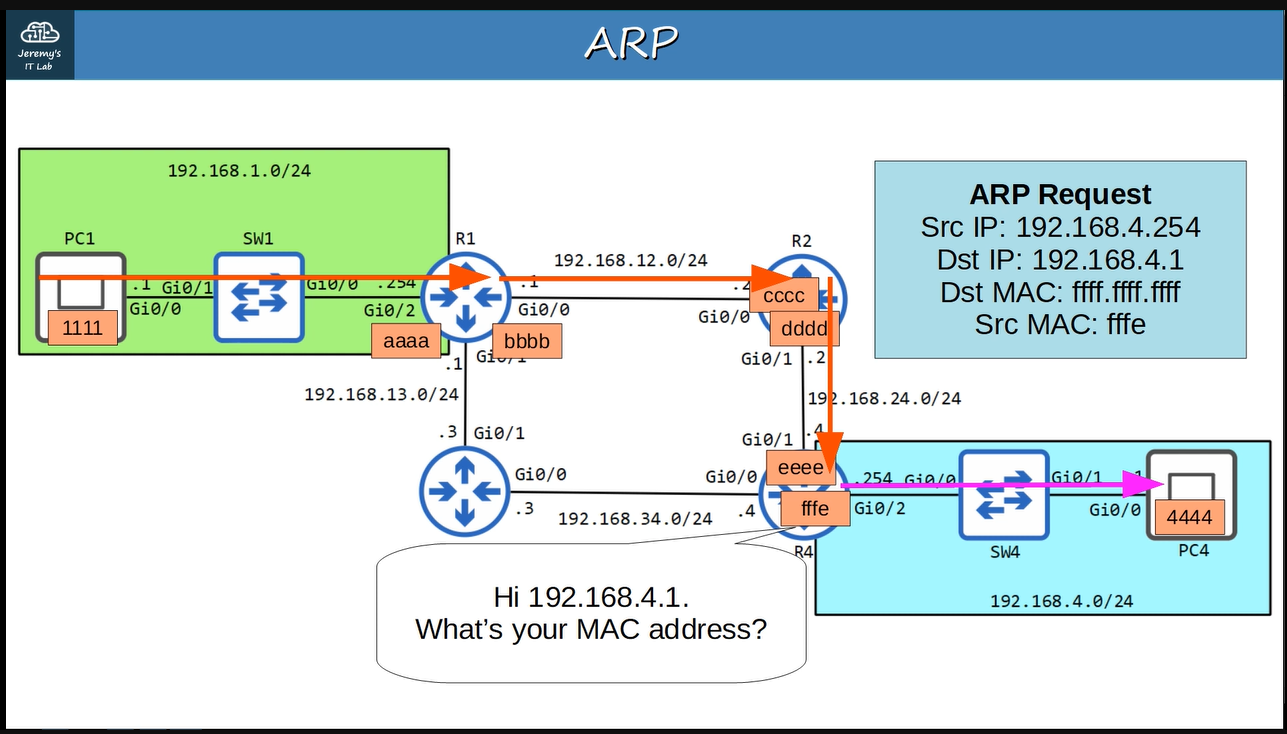
**Now R2 will receive it and de-encapsulate ethernet header and look up the destination in its routing table and next-hop ↓**

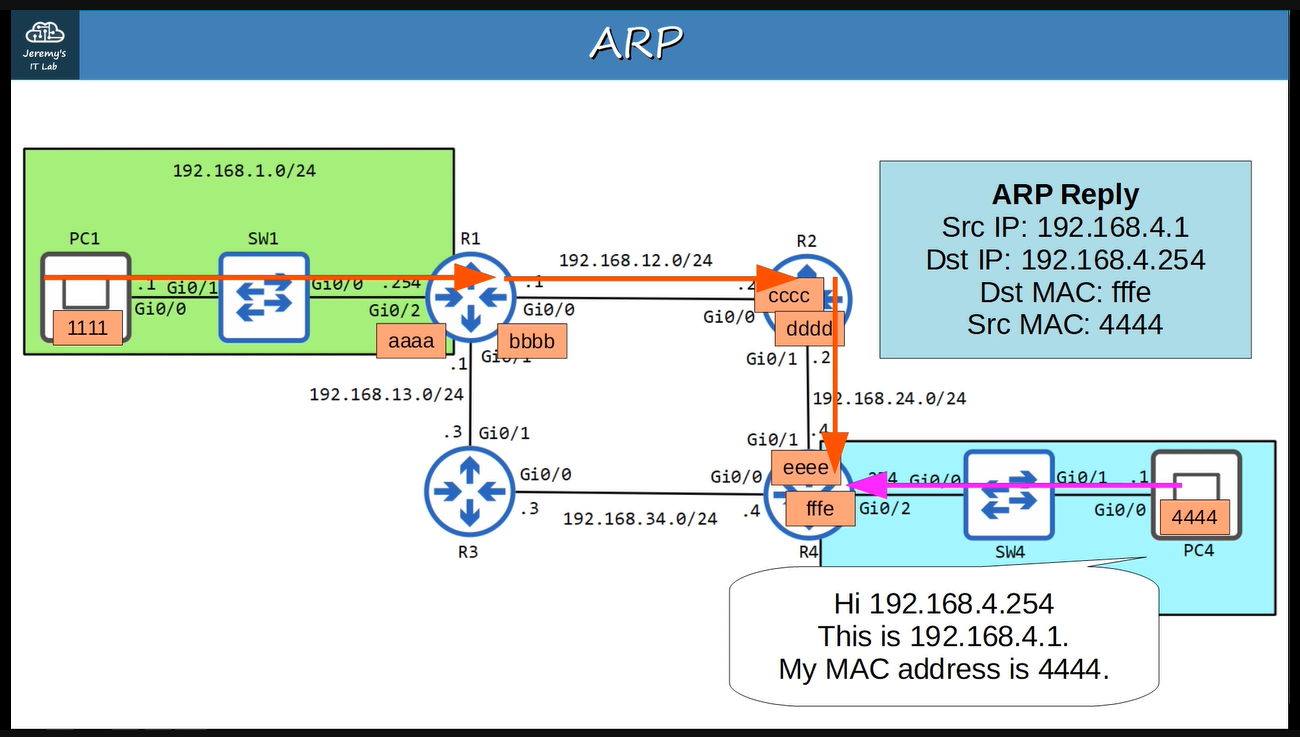
**Now R2 will encapsulate this packet with the MAC address for R4’s Gi0/1, suppose it does not know it so it will send ARP-request to R4 to know the MAC address of R4’s Gi0/1 ↓**

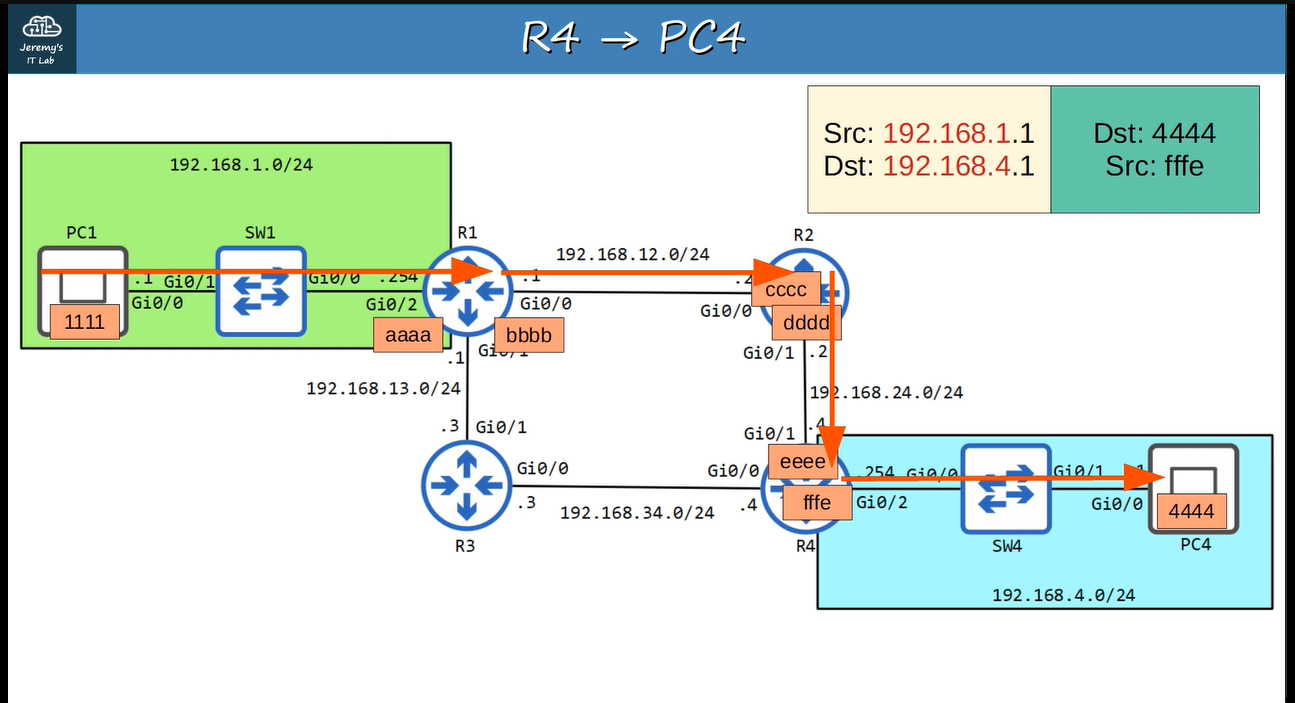
**Then R4 will send ARP-reply to R2 ↓**

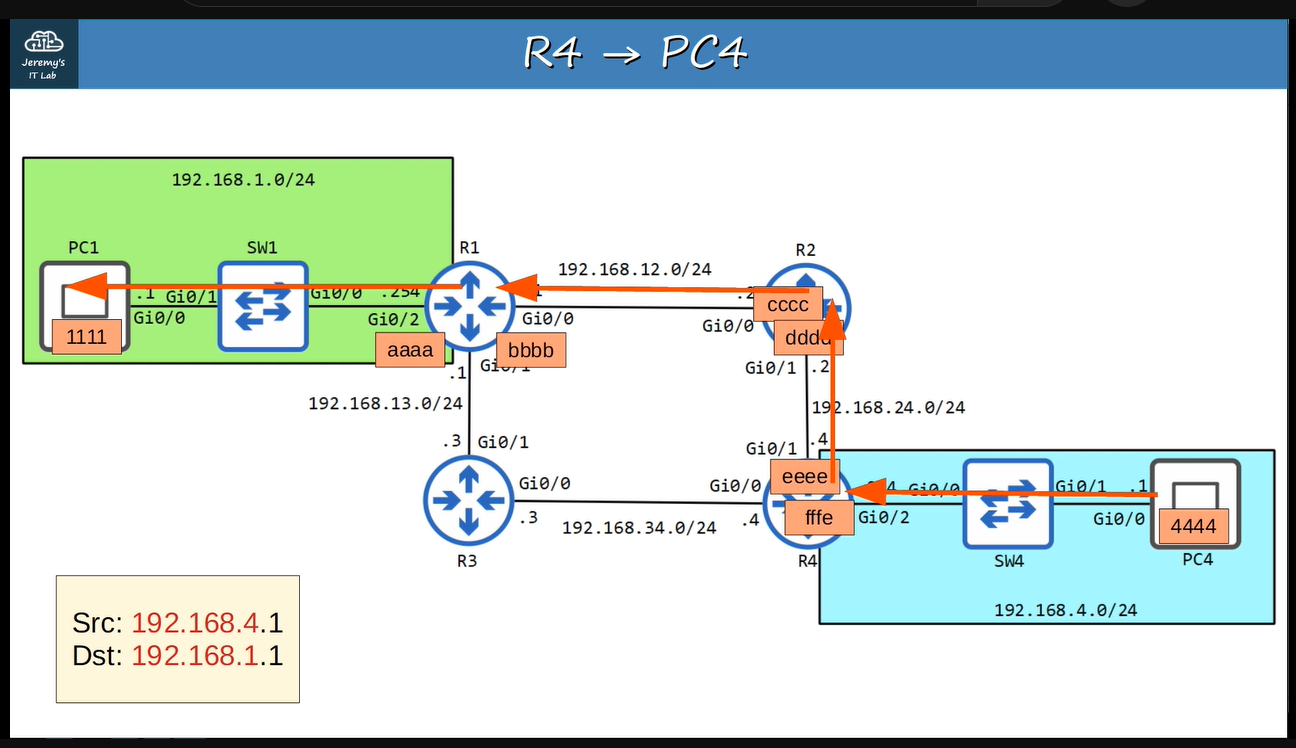
**Now R2 will send that packet to R4 at Gi0/1 with destination Mac of “eeee” ↓**

**Now R4 will receive it and de-encapsulate ethernet header and look up the destination in its routing table and next-hop ↓**

**Now R4 will encapsulate this packet with the MAC address for PC2, suppose it does not know it so it will send ARP-request to PC2 to know its MAC address ↓**

**Then PC2 will send ARP-reply to R4 ↓**

**Now R4 will send that packet to PC2 with destination Mac of “4444” ↓**

**Now if PC1 sends data to PC4 or PC4 sends data to PC1 then the ARP-request and the ARP-reply are not required ↓**