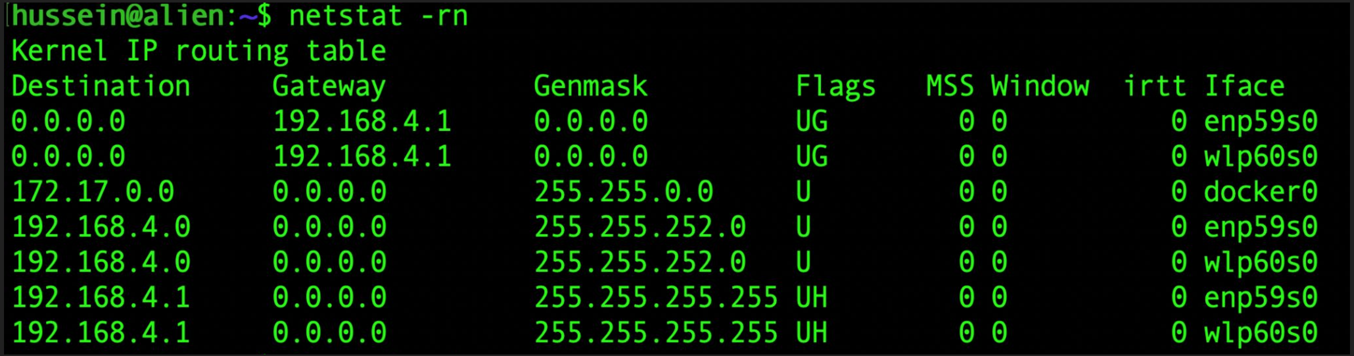


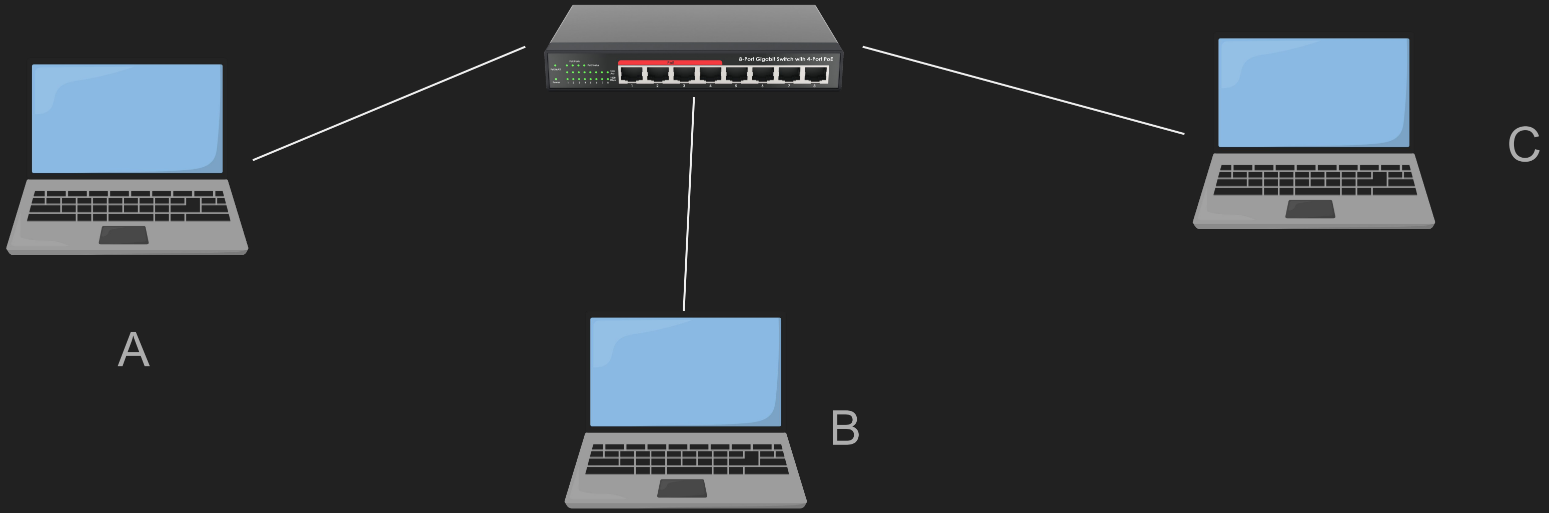
```
hussein@alien:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.4.1     0.0.0.0         UG        0 0          0 enp59s0
0.0.0.0         192.168.4.1     0.0.0.0         UG        0 0          0 wlp60s0
172.17.0.0      0.0.0.0         255.255.0.0     U         0 0          0 docker0
192.168.4.0     0.0.0.0         255.255.252.0   U         0 0          0 enp59s0
192.168.4.0     0.0.0.0         255.255.252.0   U         0 0          0 wlp60s0
192.168.4.1     0.0.0.0         255.255.255.255 UH        0 0          0 enp59s0
192.168.4.1     0.0.0.0         255.255.255.255 UH        0 0          0 wlp60s0
```

Routing Explained

From Zero

Data Links

- Each device has a link address (or MAC address)
- Devices that directly reachable devices can communicate with link address
- Using layer 2 frames



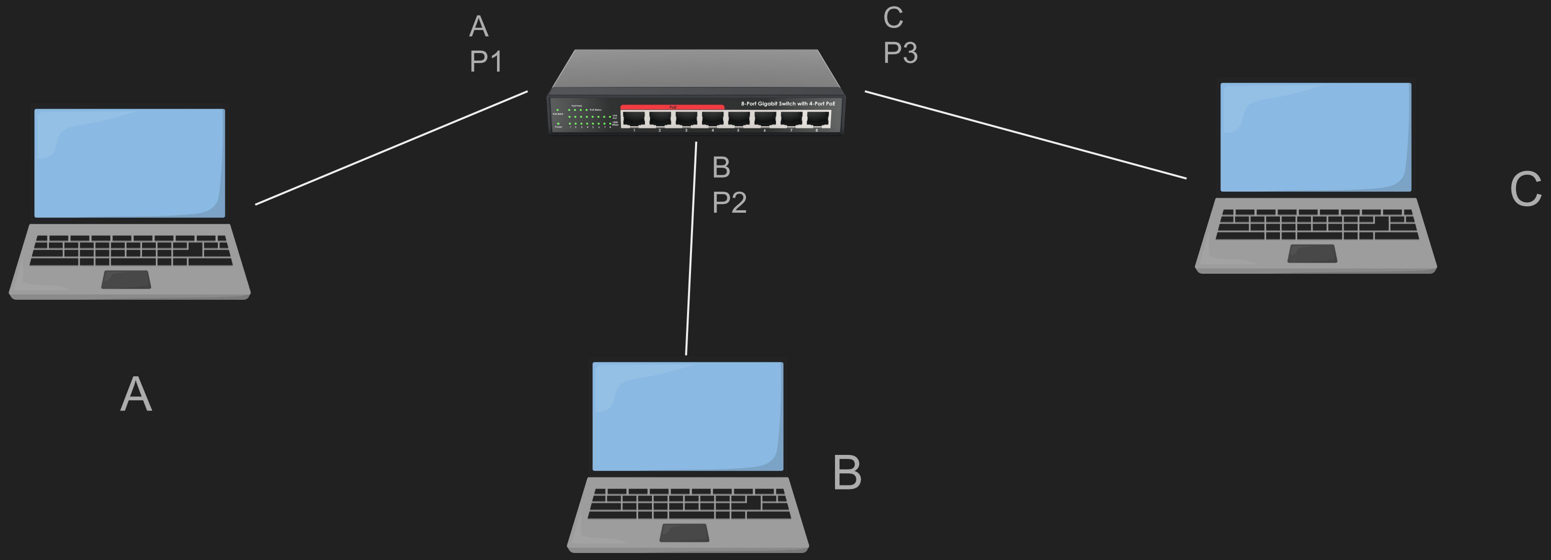
A

B

C

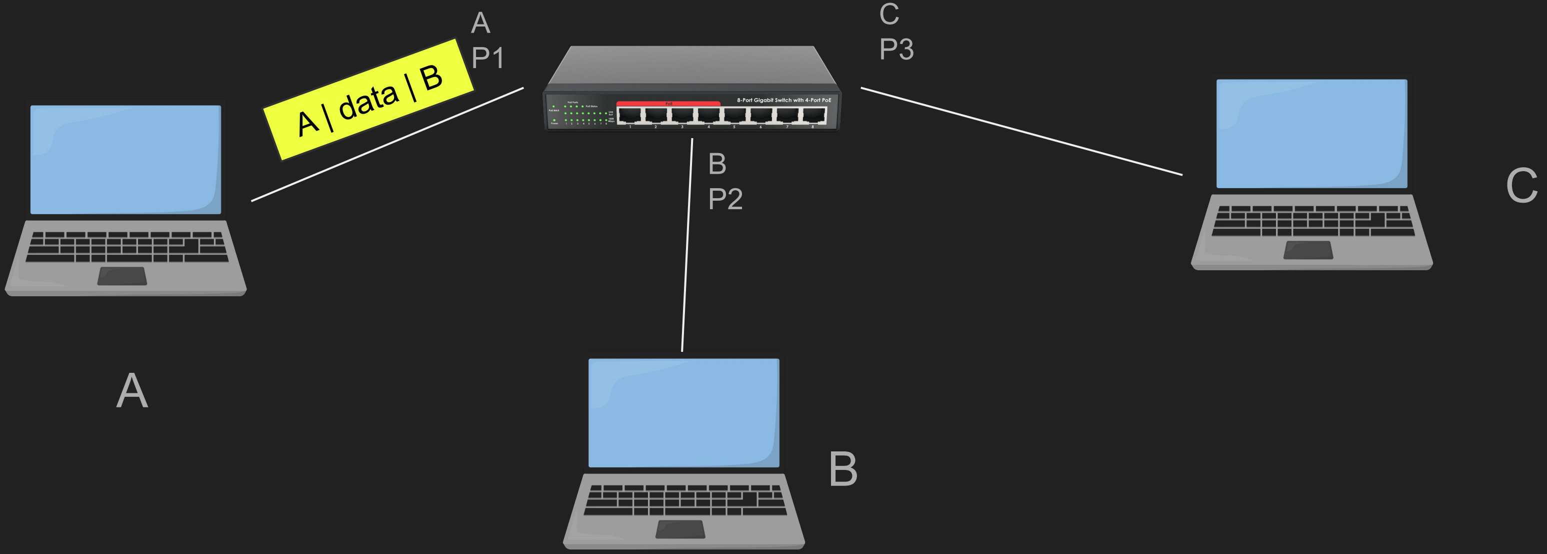
Data Links

- Each device advertises their presence
- Switches in the middle remember which port each device is at
- This is critical for performance



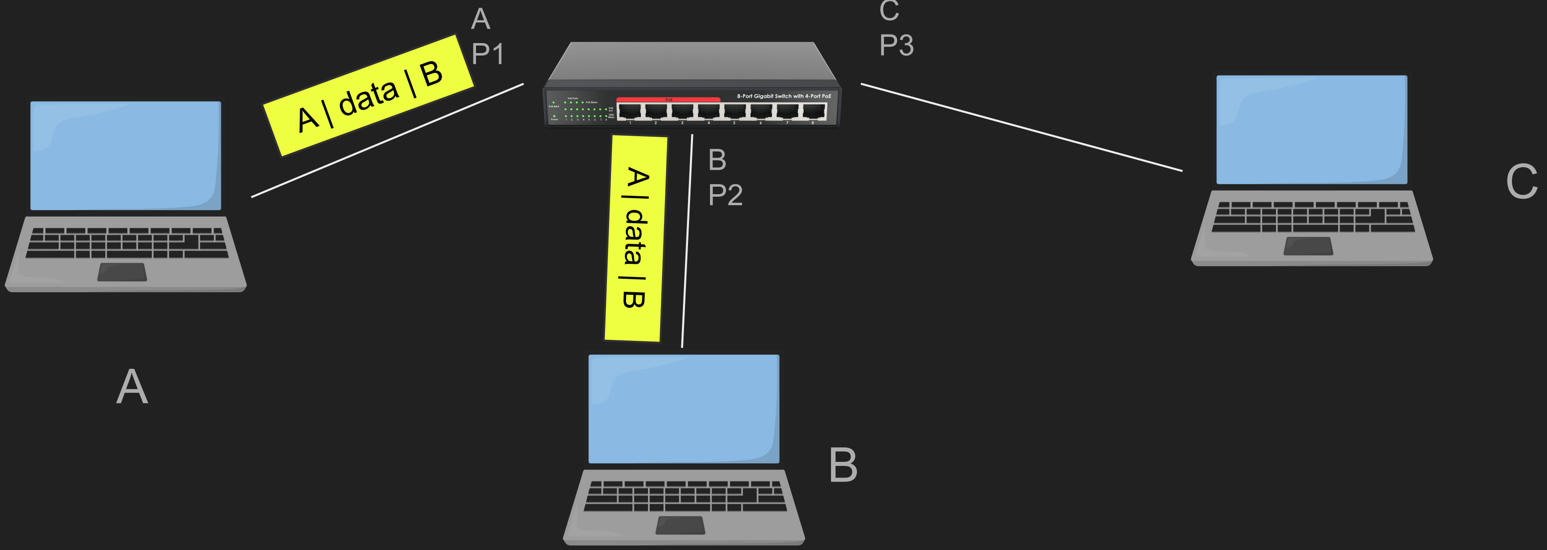
Links

- A wants to send to B (How does A knows B’s MAC? Will come to that)
- Sends it across the network



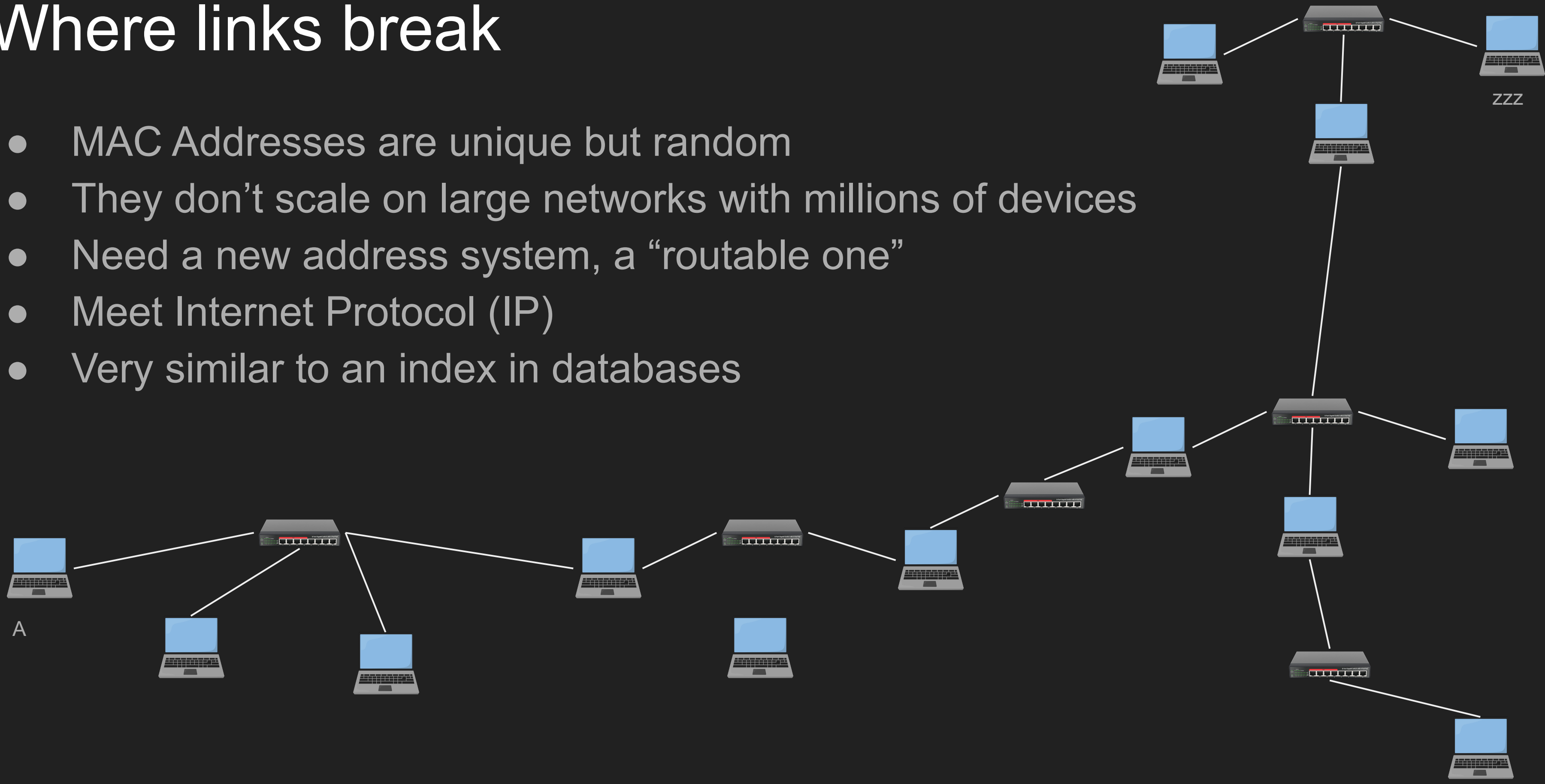
Links

- The switch receives it
- Switch knows B is at P2
- Switch only forwards the frame on port 2 (P2)



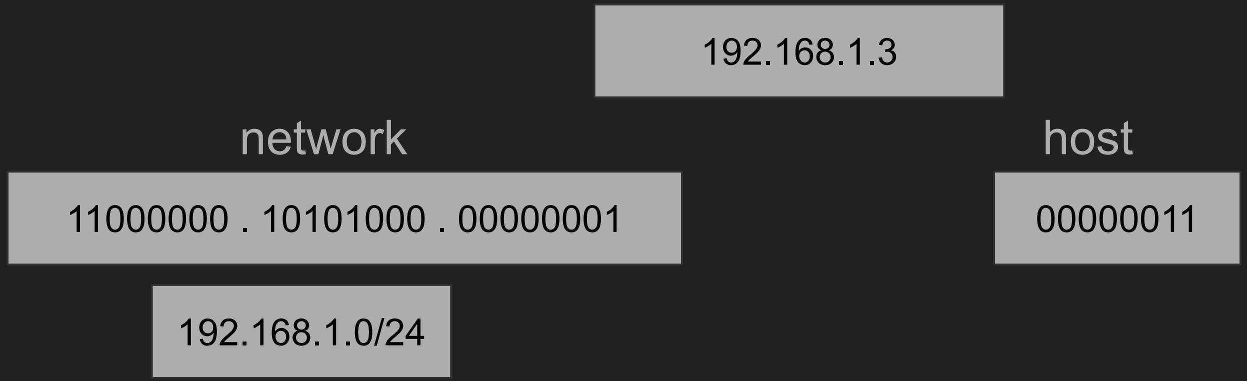
Where links break

- MAC Addresses are unique but random
- They don’t scale on large networks with millions of devices
- Need a new address system, a “routable one”
- Meet Internet Protocol (IP)
- Very similar to an index in databases



IP Address

- Internet Protocol
- 4 bytes, 32 bit (IPv4)
- Network and host sections
- E.g. 24 bit network, 8 bit for host
- RULE: Hosts within the same networks can use link address



192.168.1.3

network

host

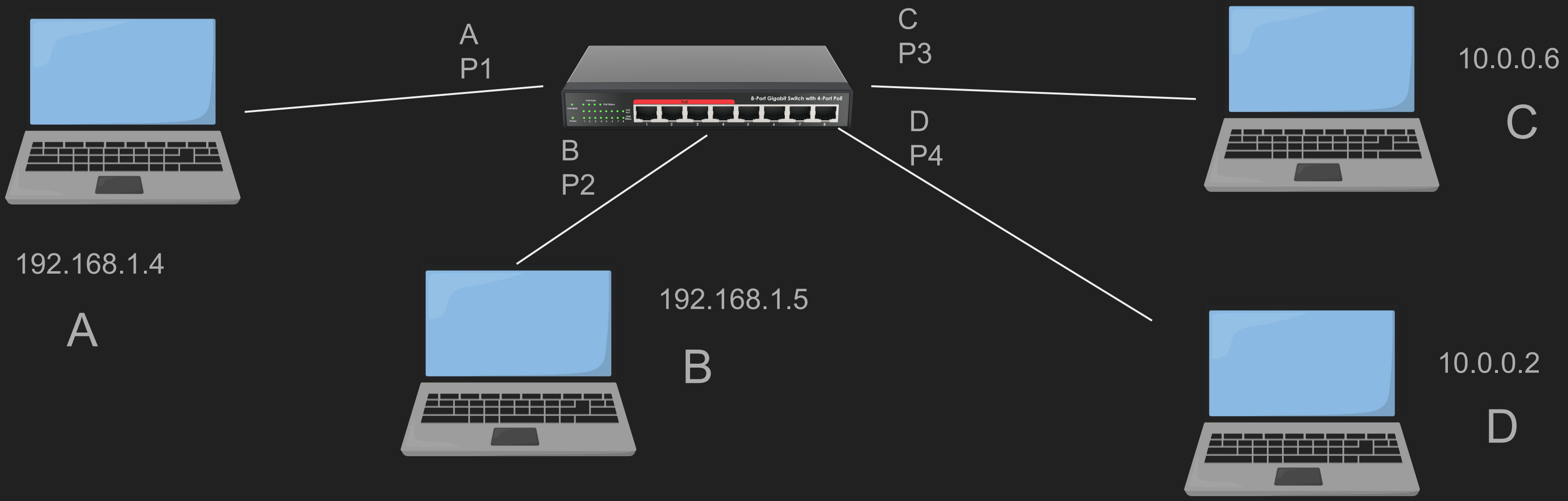
11000000 . 10101000 . 00000001

00000011

192.168.1.0/24

How to check if an IP is in my network?

- We use the network mask or subnet mask
- E.g. 192.168.1.4 is in network 192.168.1.0/24 or 255.255.255.0
- 192.168.1.5 AND with 255.255.255.0 = 192.168.1.0 (same network)
- 10.0.0.6 AND 255.255.255.0 = 10.0.0.0 (not same as 192.168.1.0)



A
P1

C
P3

B
P2

D
P4

10.0.0.6

C

192.168.1.4

A

192.168.1.5

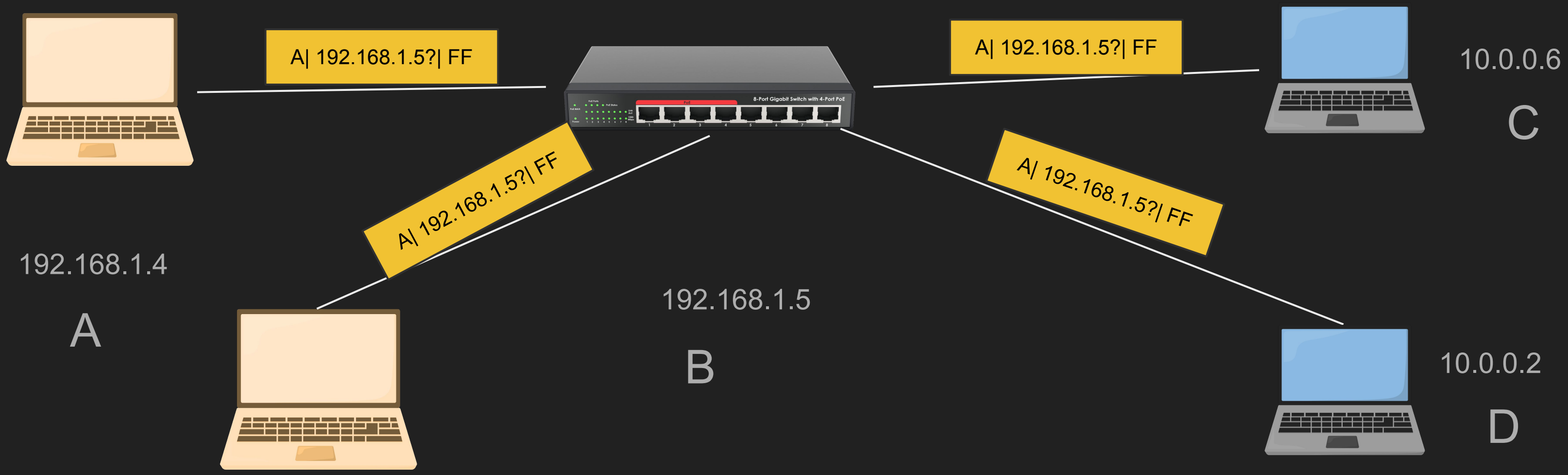
B

10.0.0.2

D

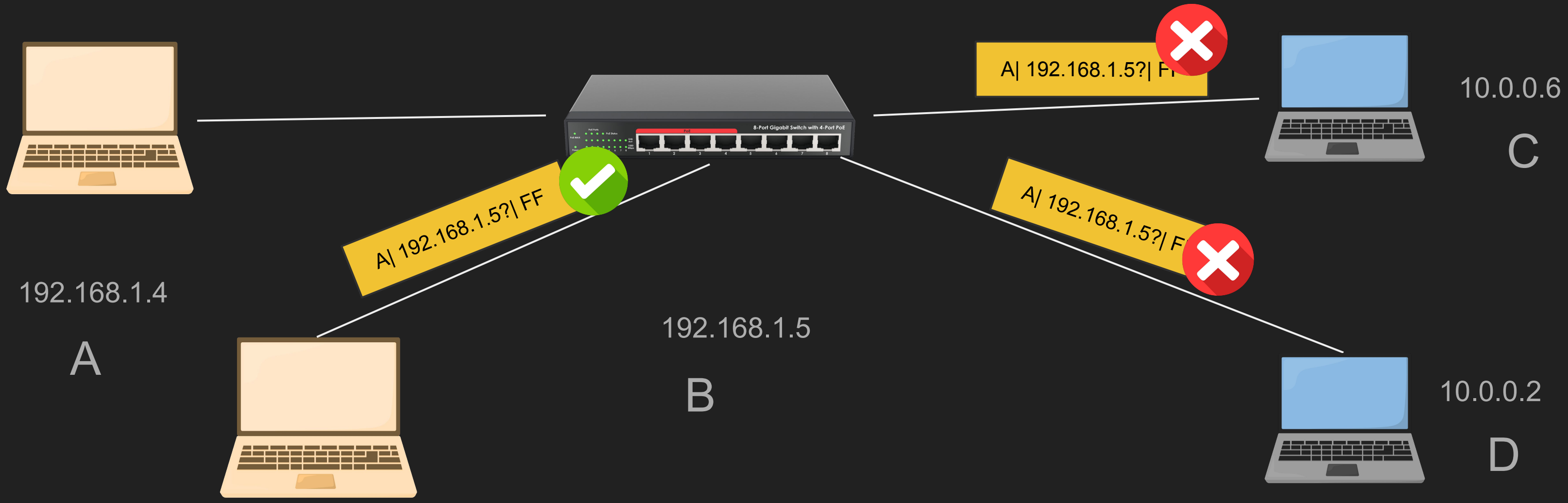
ARP

- 192.168.1.4 (A) wants to send a message to 192.168.1.5 (B)
- No communication can happen until A knows 192.168.1.5 mac (which is B)
- A sends a broadcast ARP request, who has 192.168.1.4?



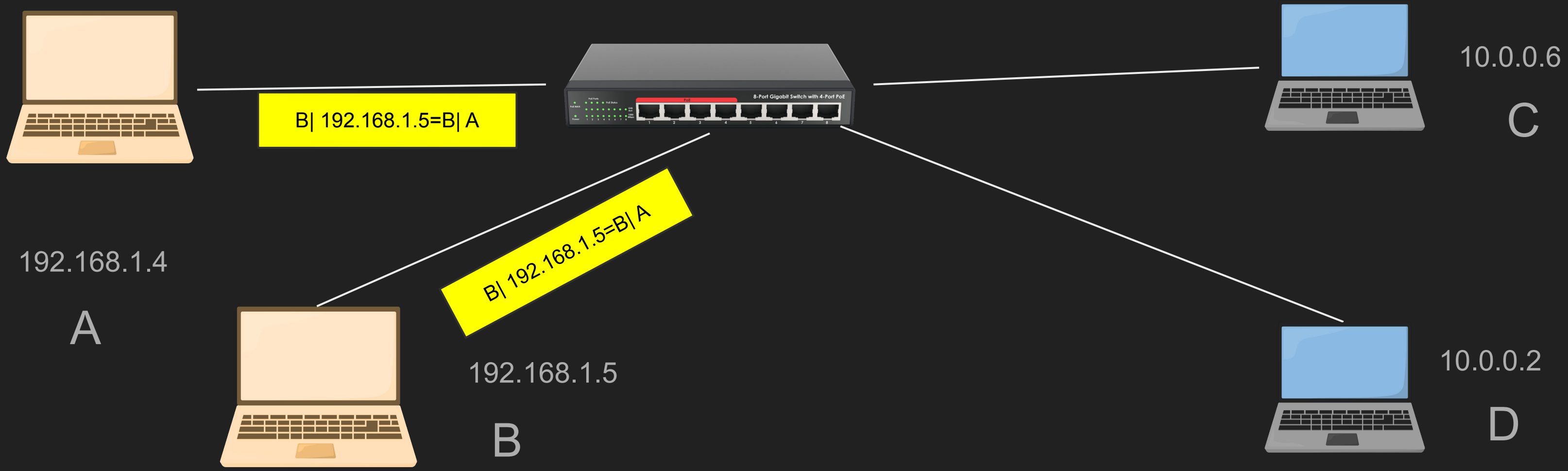
ARP

- Everyone in the broadcast receives the message.
- Only 192.168.1.5 matches, accepts it, everyone else drops it
- B now prepare to reply



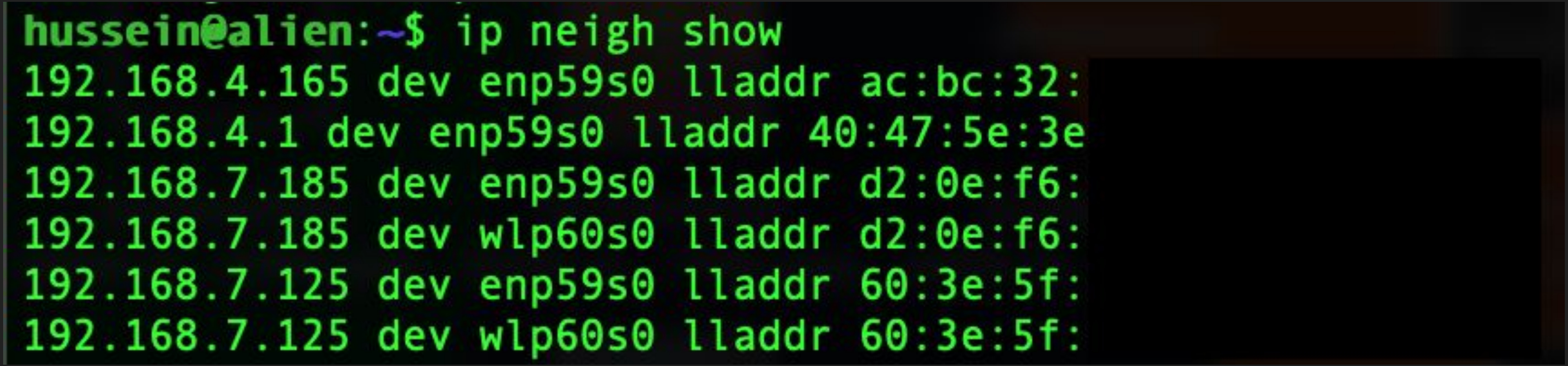
ARP Response

- 192.168.1.5 replies back to A
- 192.168.1.4 caches the result, 192.168.1.5 is B



ARP Table

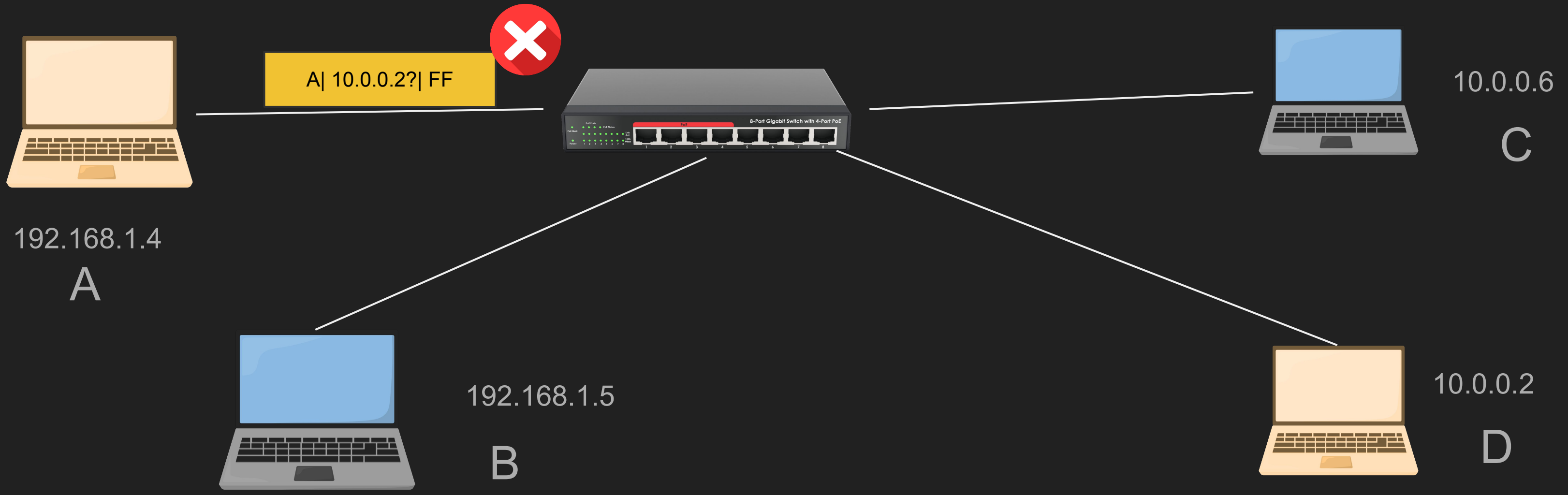
ip neigh show



```
hussein@alien:~$ ip neigh show
192.168.4.165 dev enp59s0 lladdr ac:bc:32: [redacted]
192.168.4.1 dev enp59s0 lladdr 40:47:5e:3e
192.168.7.185 dev enp59s0 lladdr d2:0e:f6:
192.168.7.185 dev wlp60s0 lladdr d2:0e:f6:
192.168.7.125 dev enp59s0 lladdr 60:3e:5f:
192.168.7.125 dev wlp60s0 lladdr 60:3e:5f:
```

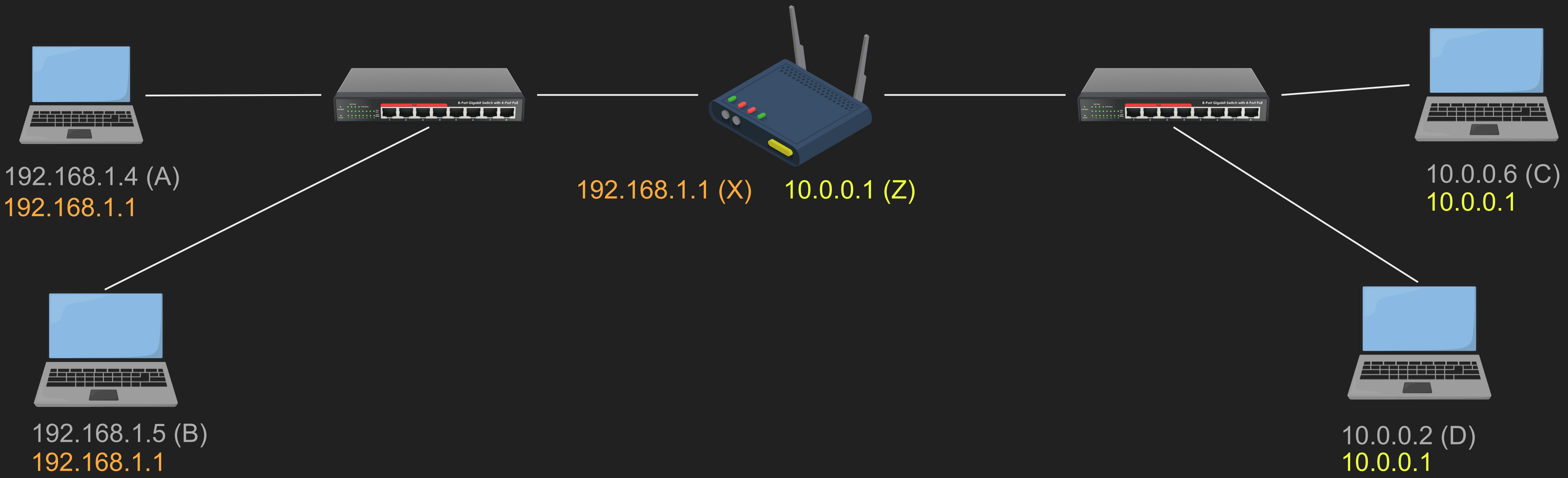
ARP notes

- ARP only works on the same subnet
- 192.168.1.4 cannot ask for 10.0.0.2’s MAC
- No ARP request will be sent.
- How do we then talk to different networks?



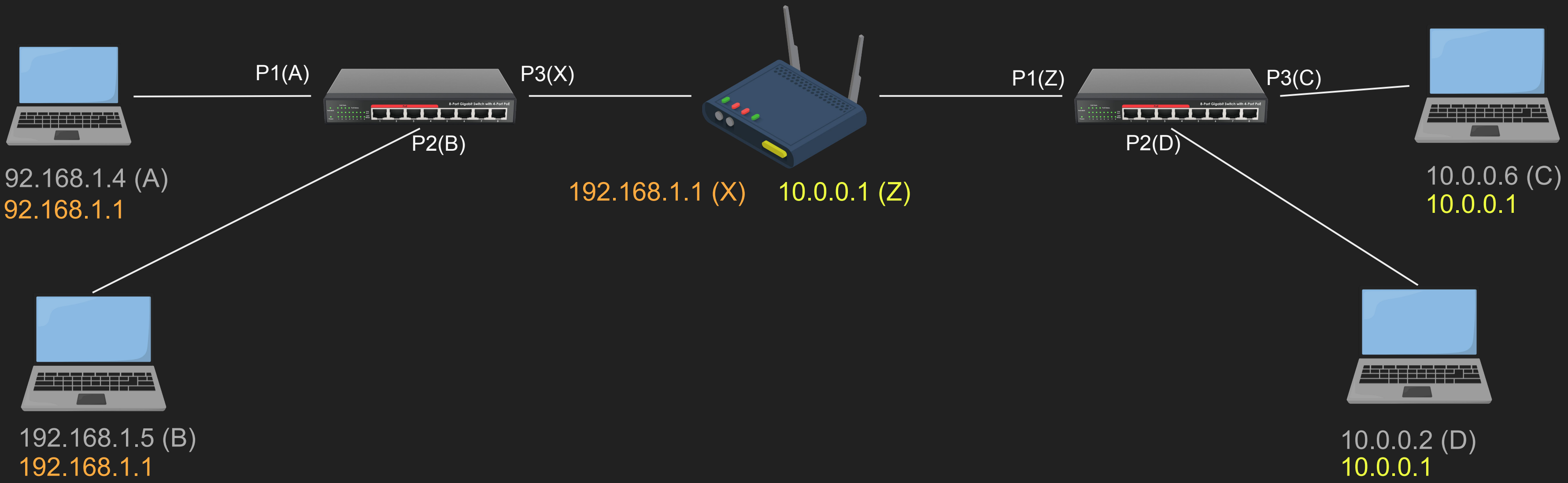
Meet the Gateway, next hop or, the router!

- For a device to talk on a device on another network needs a gateway
- The gateway belongs to both networks
- Devices are aware of their gateway (configured)



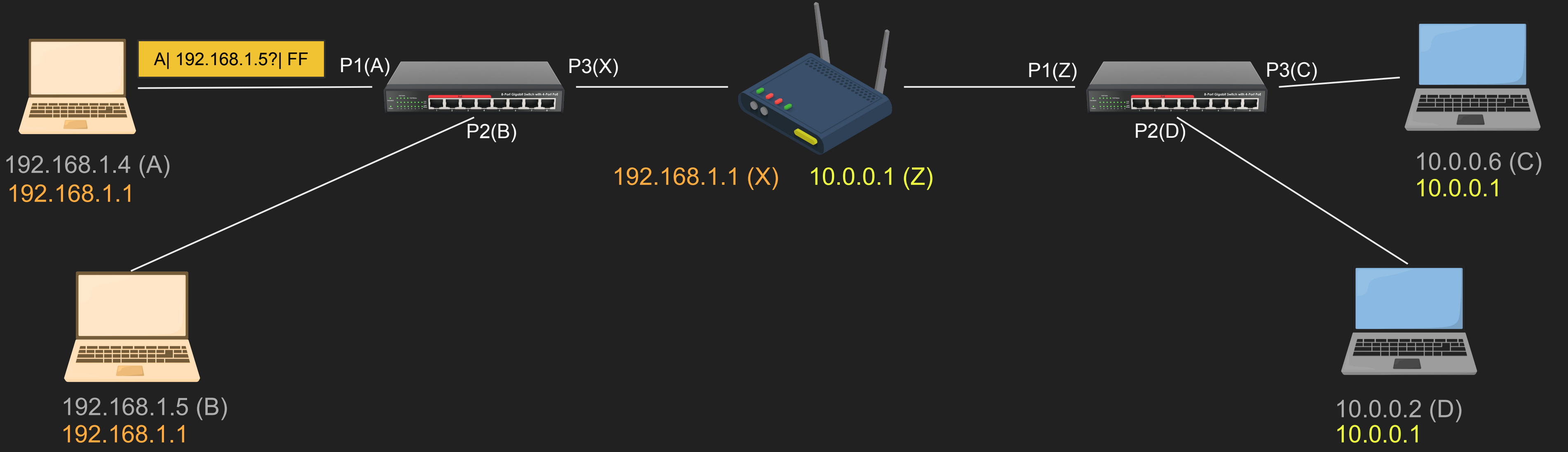
Switches get updated

- Router/Gateway is just another device with a MAC
- Switches get updated with the MACs of the connected devices



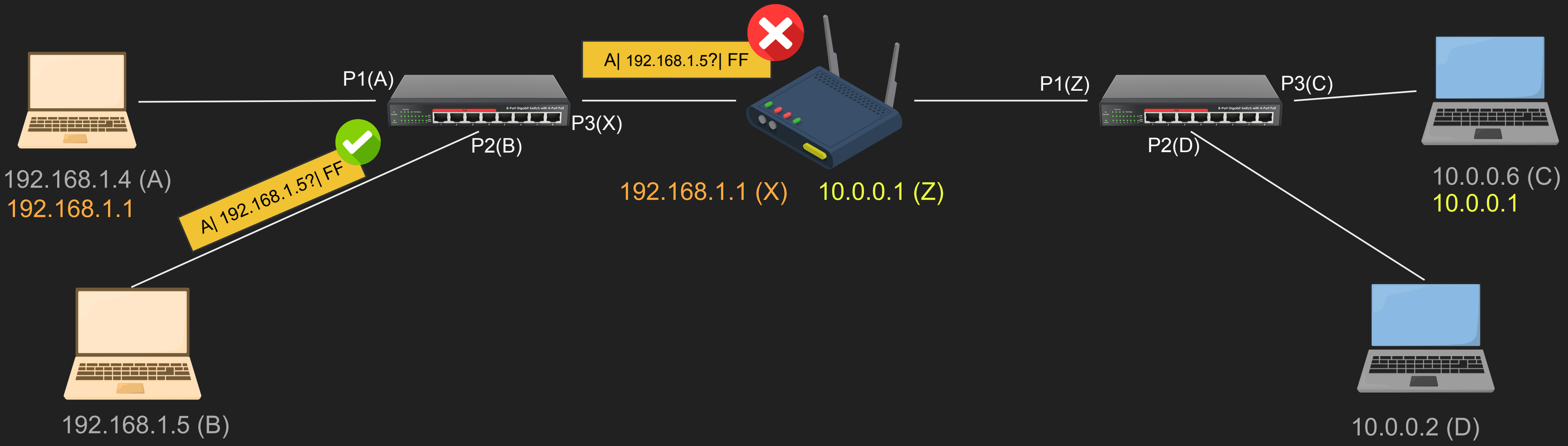
Link networking example

- 192.168.1.4 wants to talk to 192.168.1.5 (It only knows the IP)
- Checks if its in the same network
  - 192.168.1.5 x 255.255.255.0 = 192.168.1.0, same network
  - Means we can directly communicate with data link (MAC Addresses)
  - 192.168.1.4 needs the MAC address of 192.168.1.5 (Sends ARP)



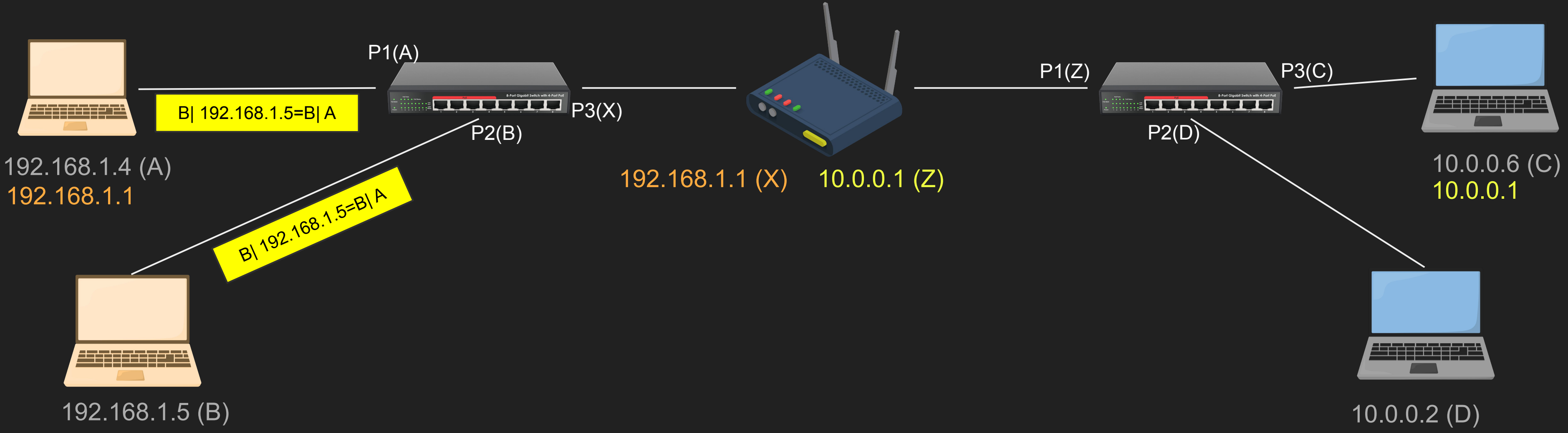
Link networking example

- ARP gets to the switches, sends it to all ports
- The router itself is a device, and it is not 192.168.1.5 so it doesn't response
- The ARP request is NOT sent through the other network (saving bandwidth)



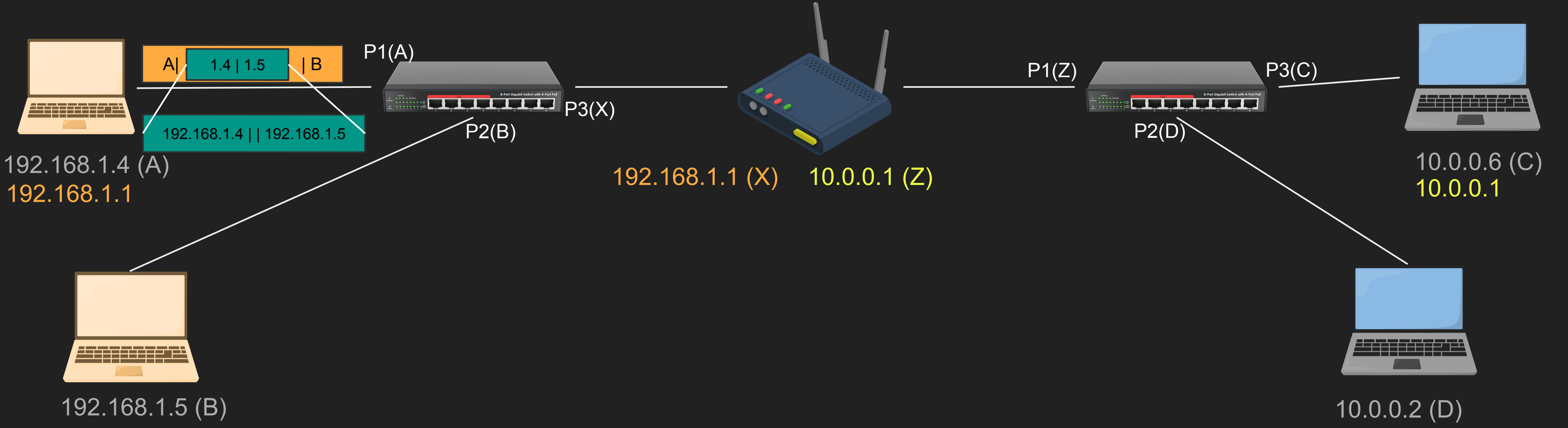
Link networking example

- B replies back, switch sends the message only to A (why?)



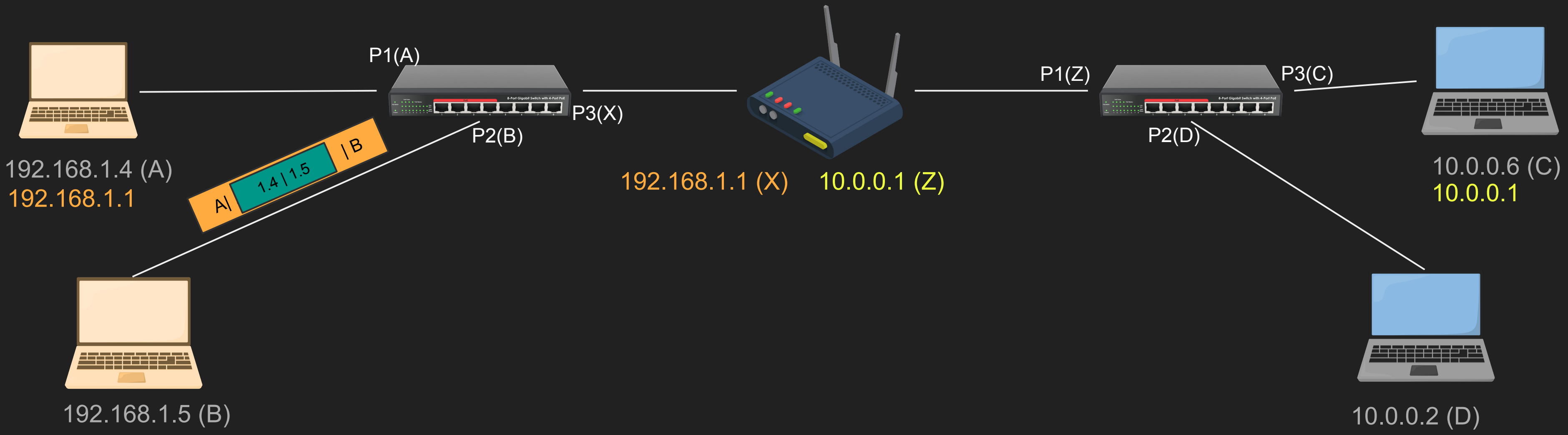
Link networking example

- Now that we know the direct link address we send an IP packet
- Source IP: 192.168.1.4, Destination IP 192.168.1.5
- Encapsulated in a Data Link frame
- Source MAC: A, Destination MAC: B



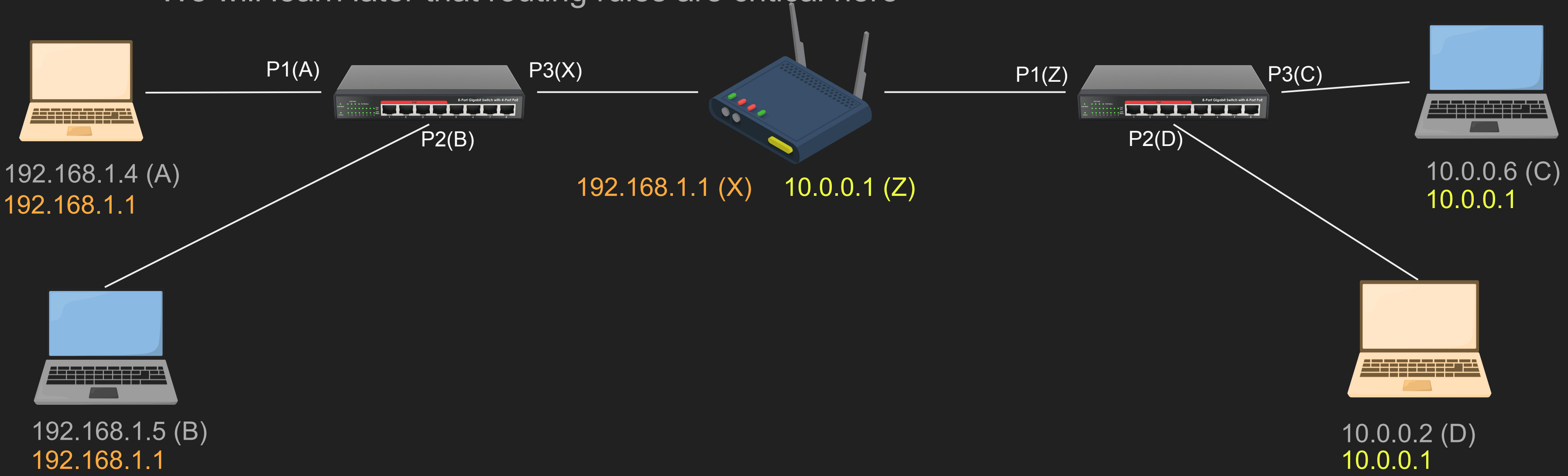
Link networking example

- The frame goes to the switch, switch writes it to P2 only!
- Local link delivery



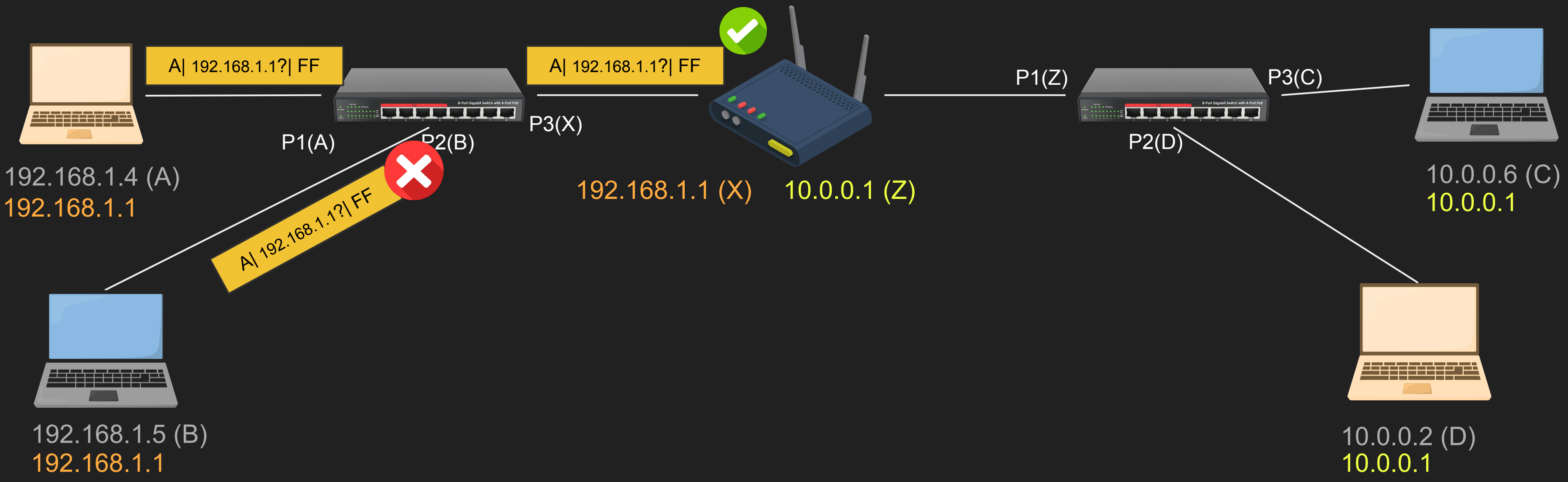
Inter-network communications

- 192.168.1.4 wants to send a message to 10.0.0.2
- Can’t do ARP, because it is not in the same network. (No direct link)
    - 192.168.1.0 != 10.0.0.0
- Only way is the gateway (default)*
    - We will learn later that routing rules are critical here



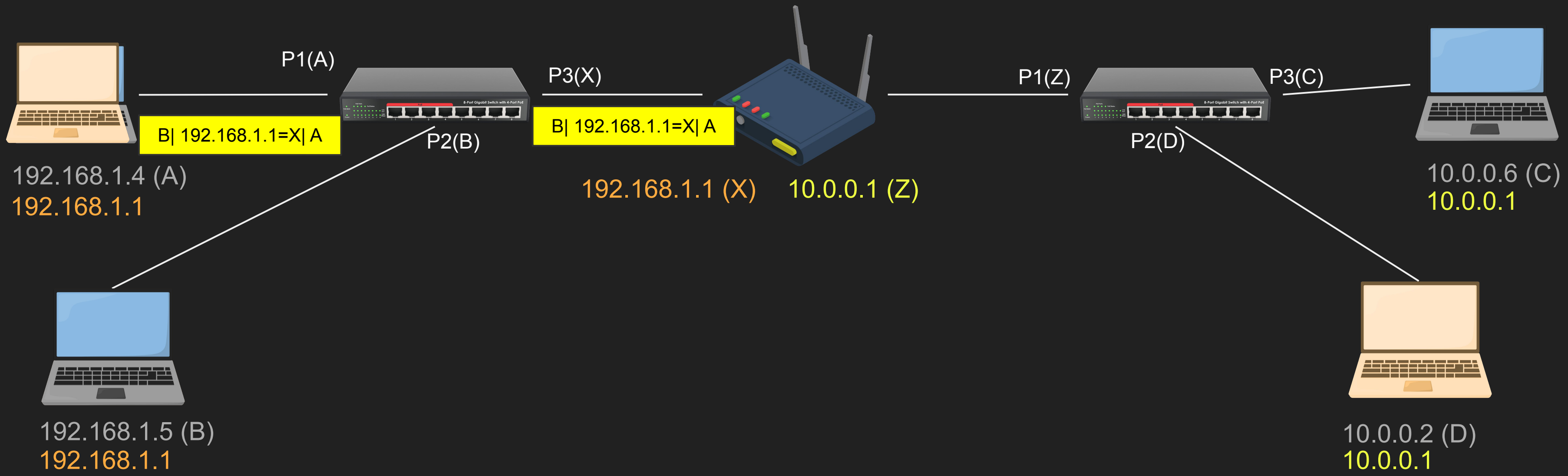
Inter-network communications

- 192.168.1.5 needs to talk to the gateway which is 192.168.1.1
- 192.168.1.5 and 192.168.1.1 in the same network
- We do ARP to get the gateway’s MAC



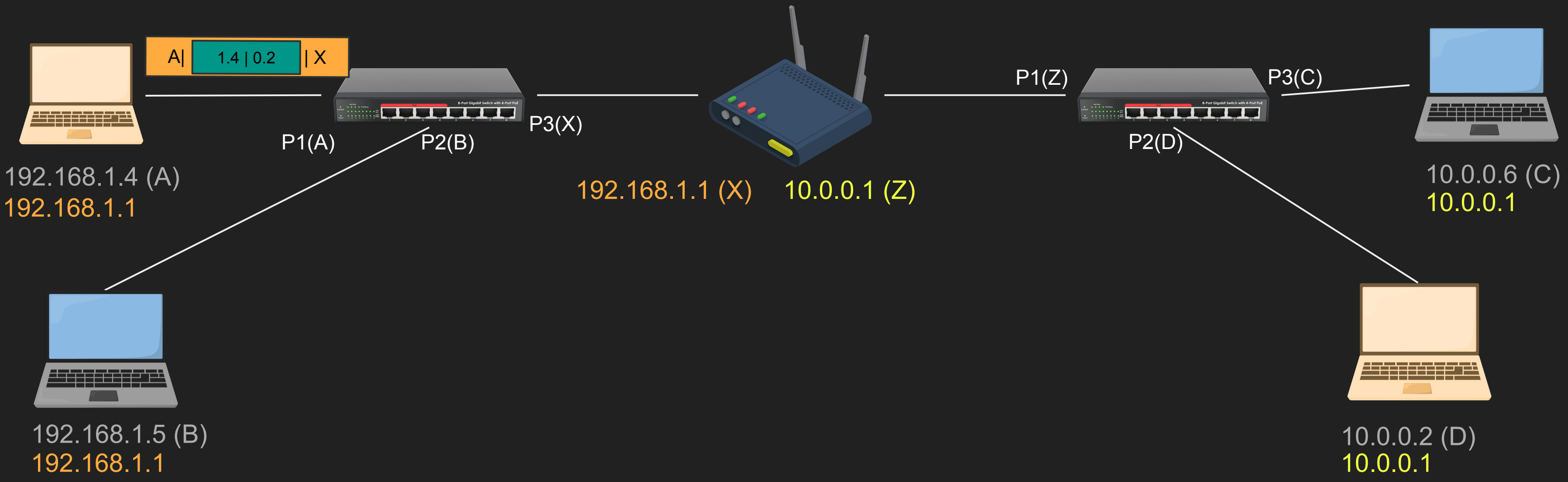
Inter-network communications

- The gateway reply that 192.168.1.1 = X
- A now knows that X = 192.168.1.1 and prepares to send the data frame



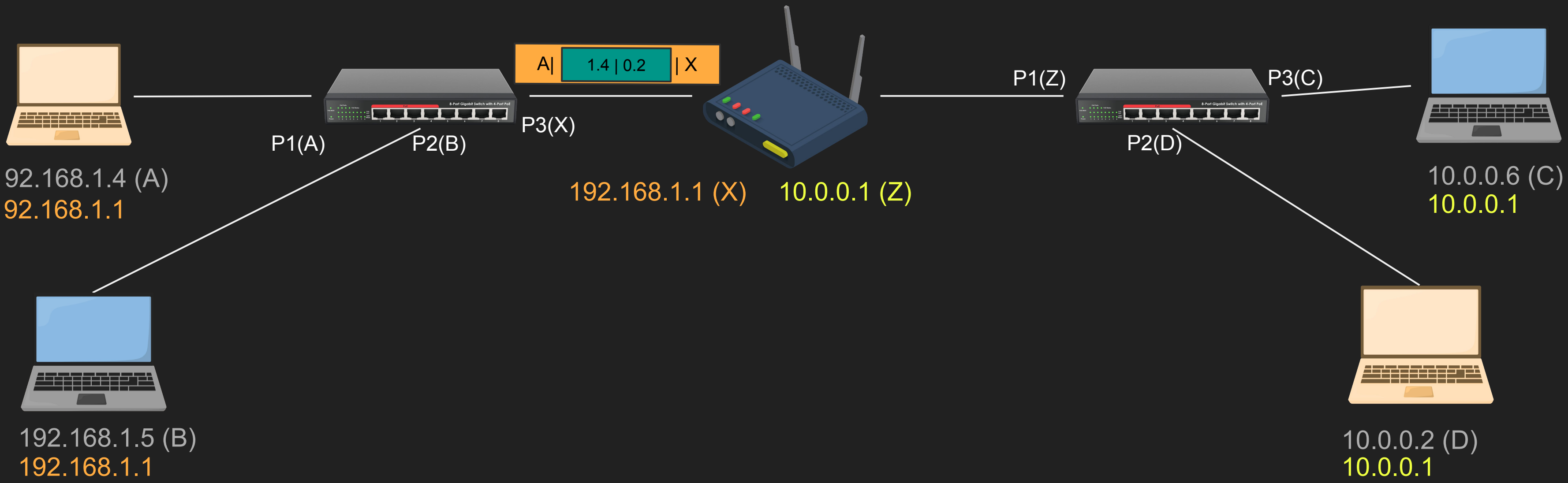
Inter-network communications

- The IP Packet is Destination 10.0.0.2
- Source IP is 192.168.1.4
- Frame: Source MAC : A
- Frame: Dest MAC : X



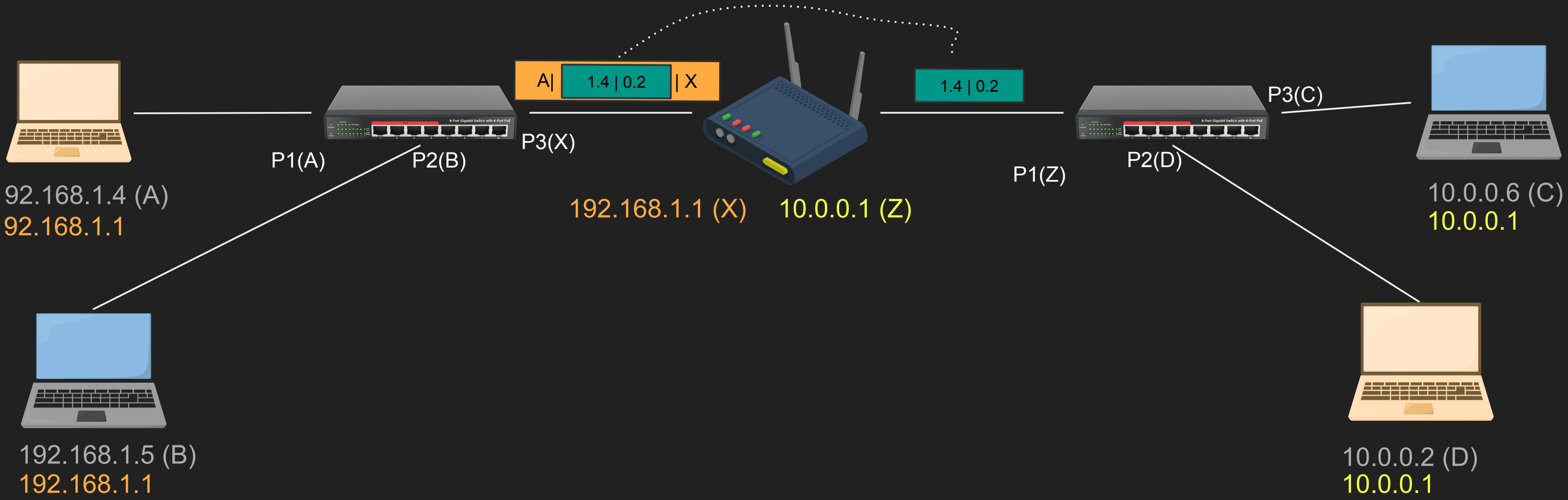
Gateway gets the frame

- The frame is destined to gateway (X)
- Gateway cracks the frame, sees the IP packet
- The destination IP is NOT same as his
- IP 10.0.0.2 is not 10.0.0.1 nor 192.168.1.1, IP Forwarding kicks in..



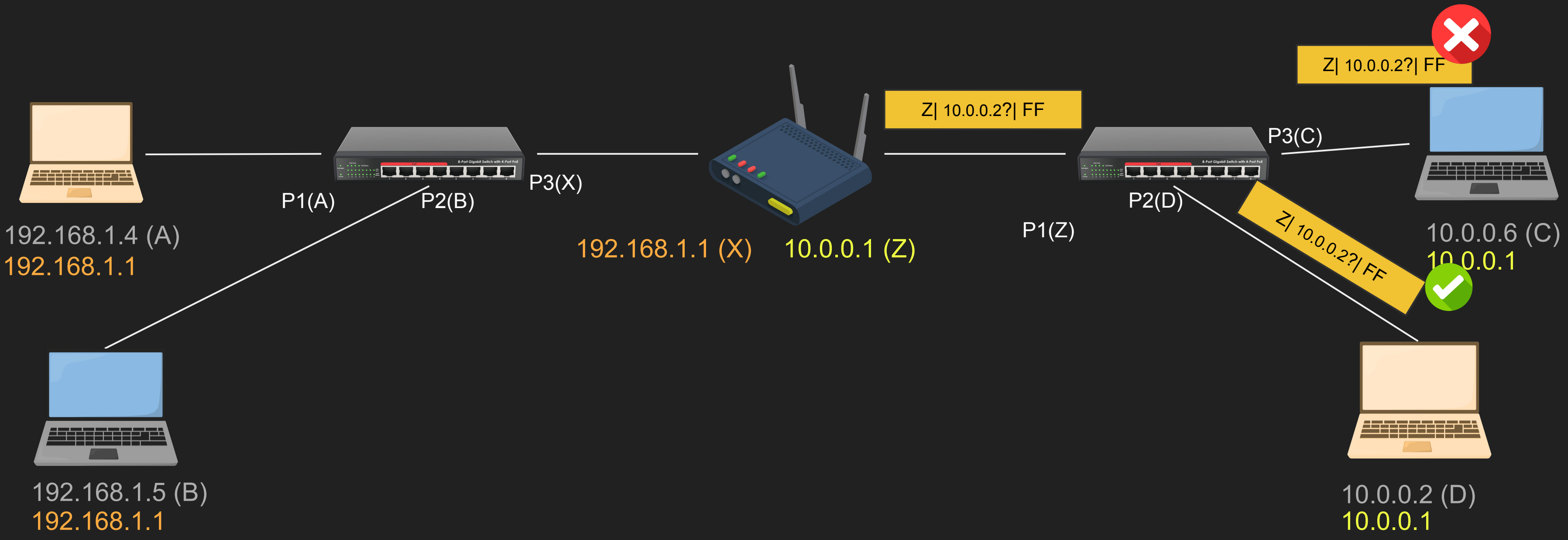
IP Forwarding

- In normal situations the packet is dropped (nothing is there
- The OS kernel has a feature to enable IP Forwarding,
- I.e. if you receive a frame that is for you but the IP is not for you
- You can forward the IP packet to other interfaces



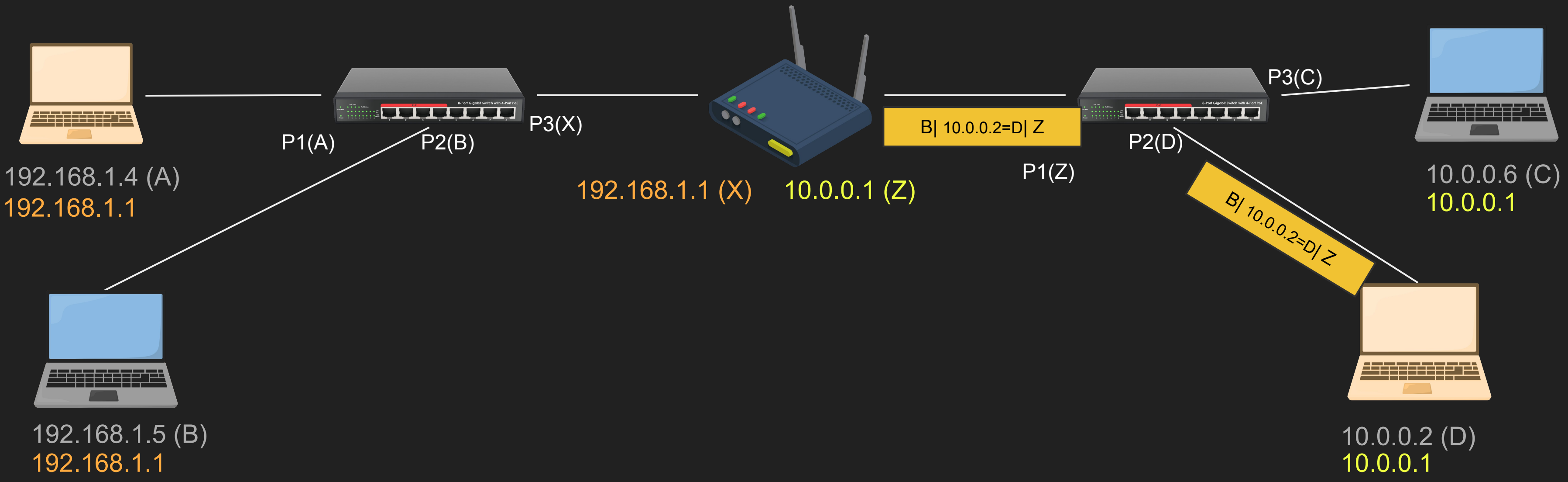
ARP on the other network

- The gateway now has to do an ARP on the other network
- Who has 10.0.0.2 ? Note that Z is our MAC now.
- We get D!



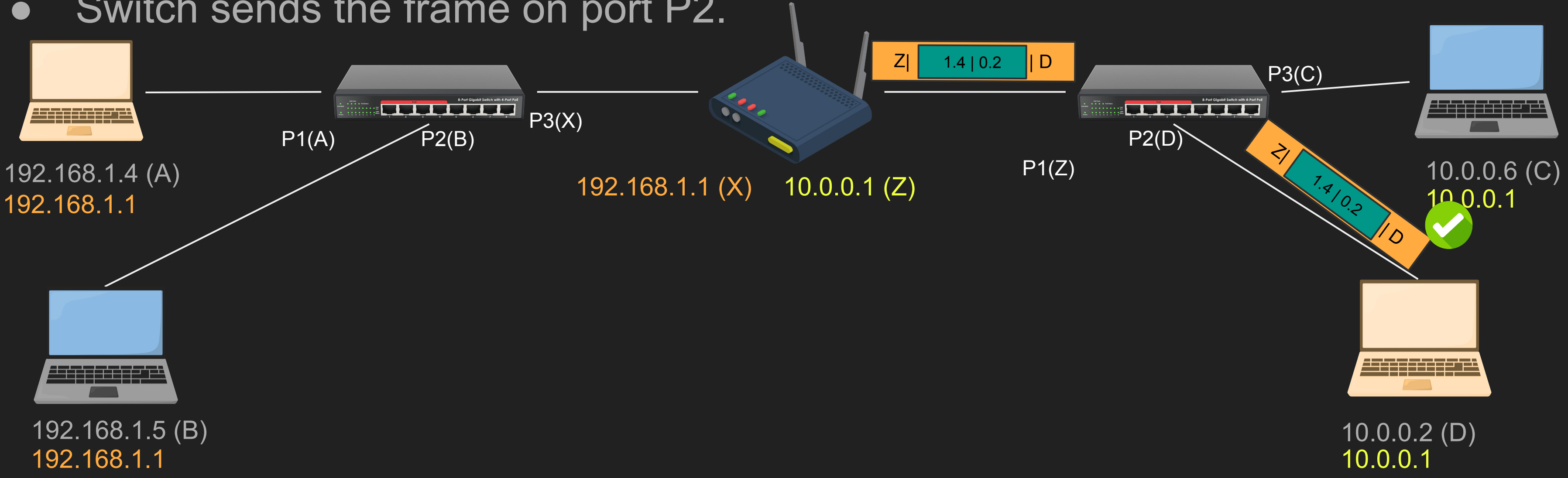
ARP on the other network

- D replies back to the ARP
- Goes through port P1 on switch
- Gateway sends the packet.



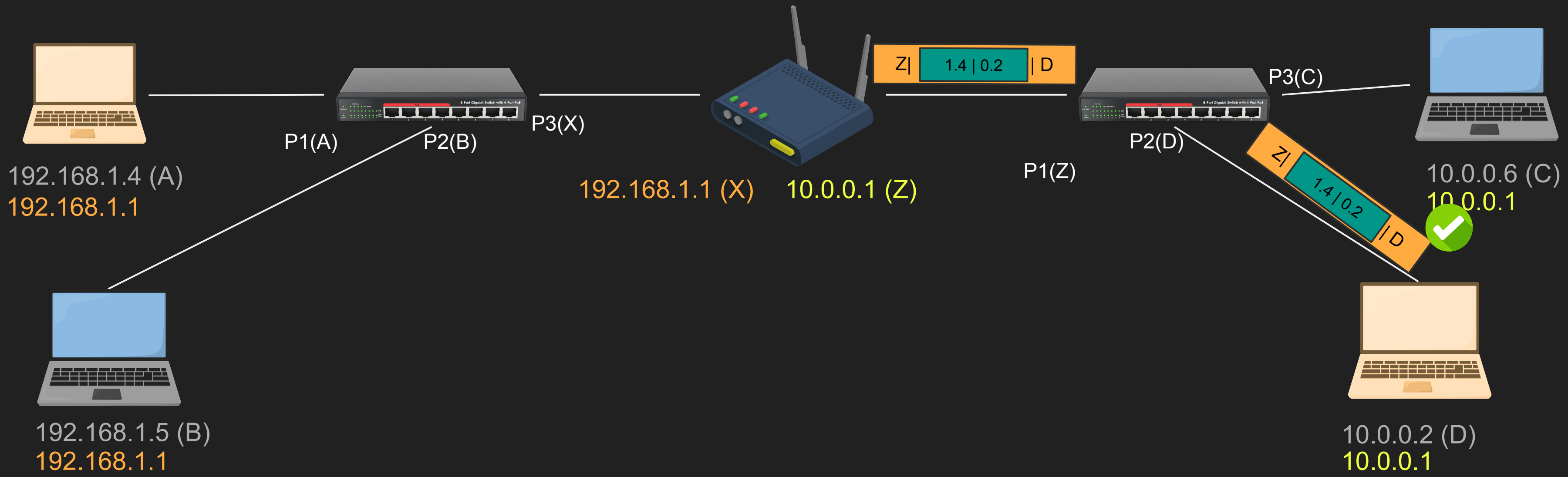
Gateway forwards the packet

- D replies back to the ARP
- Goes through port P1 on switch
- Gateway sends the packet in a frame destined for D
- Gateway keeps the source ip (unless NAT)
- Switch sends the frame on port P2.



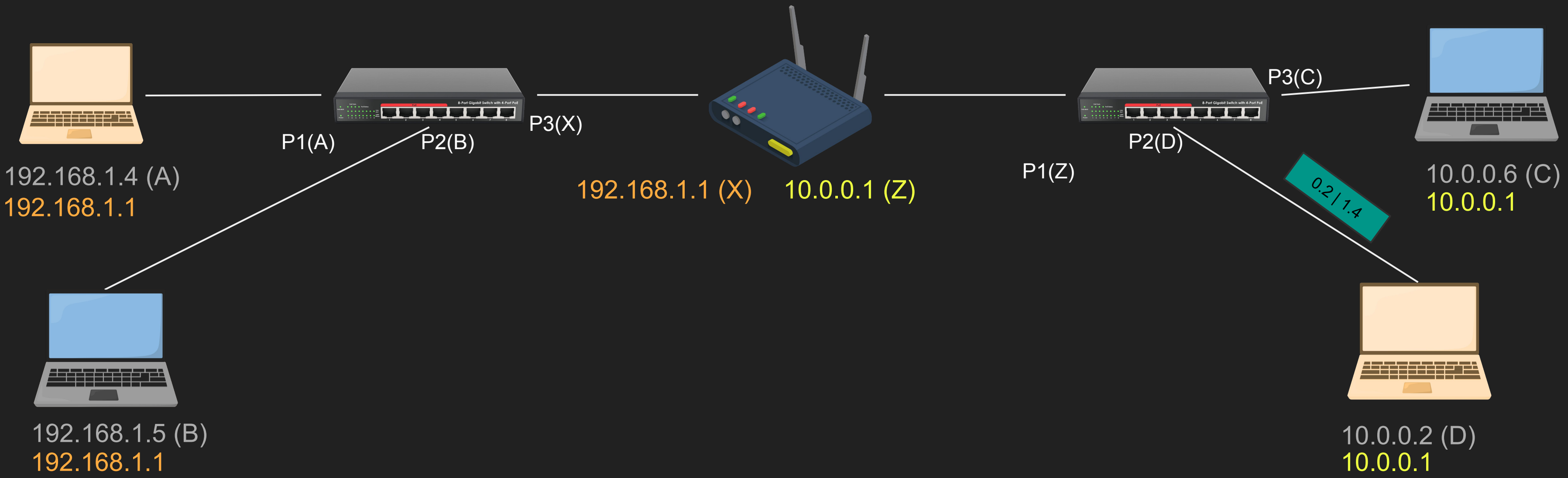
Machine receives it

- 10.0.0.2 receives the frame, it is destined for D so it accepts it.
- Cracks open the IP packet, the IP is also destined to it 10.0.0.2
- Kernel delivers it to the application.



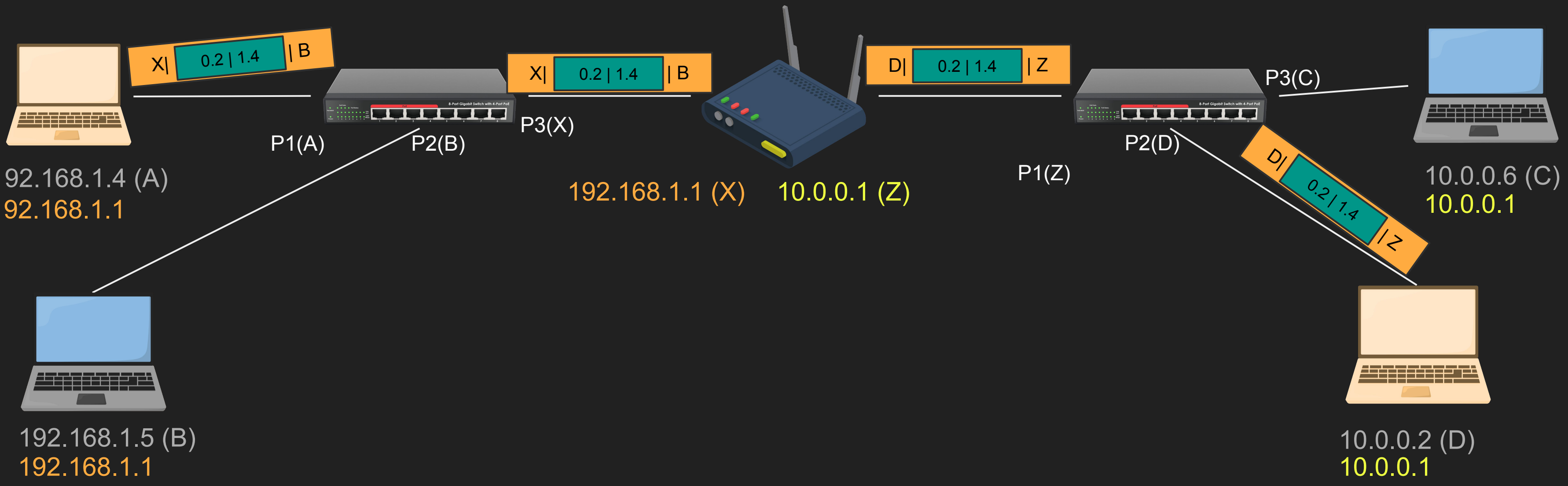
Inter-network reply

- Let us walk through a reply?
- It's just an IP packet with destination as 192.168.1.4
- Same exact story, 192.168.1.4 is not in my network, send it to the gateway



Inter-network reply

- Let us walk through a reply?
- It's just an IP packet with destination as 192.168.1.4
- Same exact story, 192.168.1.4 is not in my network, send it to the gateway

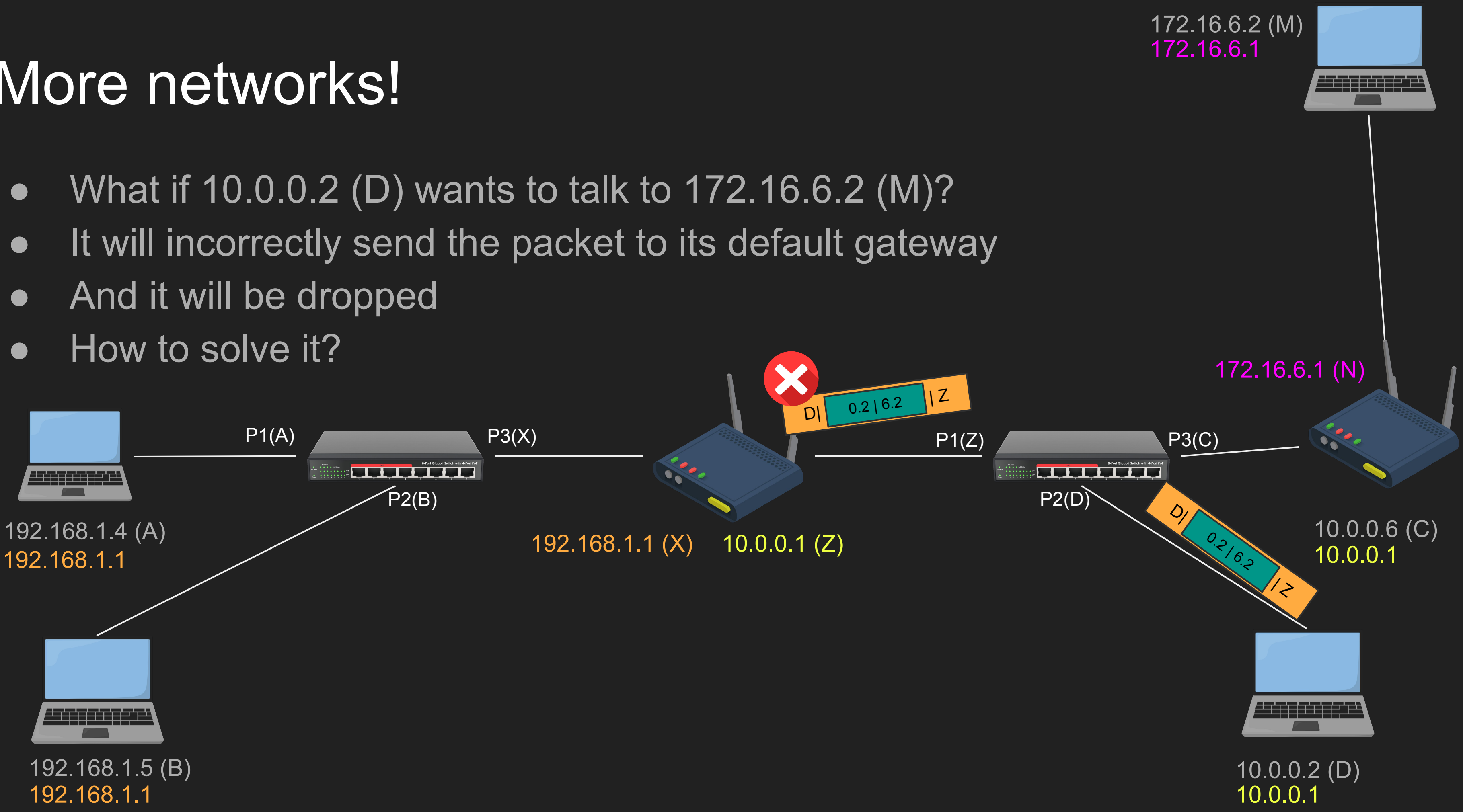


Multiple gateways and paths?

- You can have multiple gateways
- With one default gateway doesn’t always work
- We need more granularity
- Meet the routing table

More networks!

- What if 10.0.0.2 (D) wants to talk to 172.16.6.2 (M)?
- It will incorrectly send the packet to its default gateway
- And it will be dropped
- How to solve it?

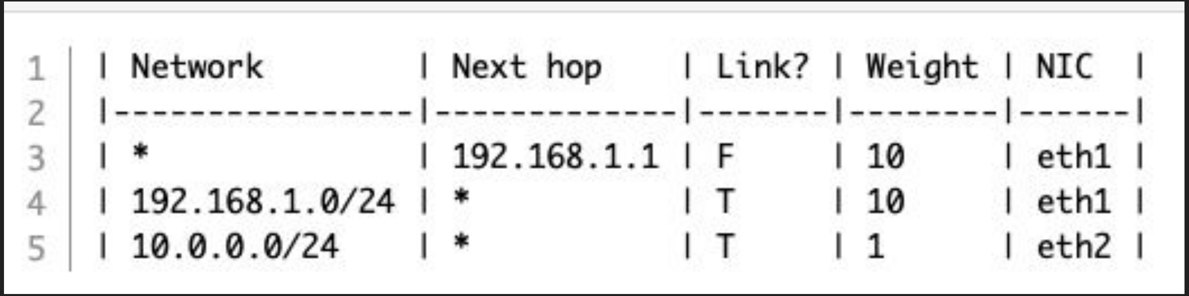


Routing Table

- Where do you want go? (Network)
- What is the nextHop (gateway)?
- Is it direct link or not?
- Weight (metric)
- Which interface to go through

Examples

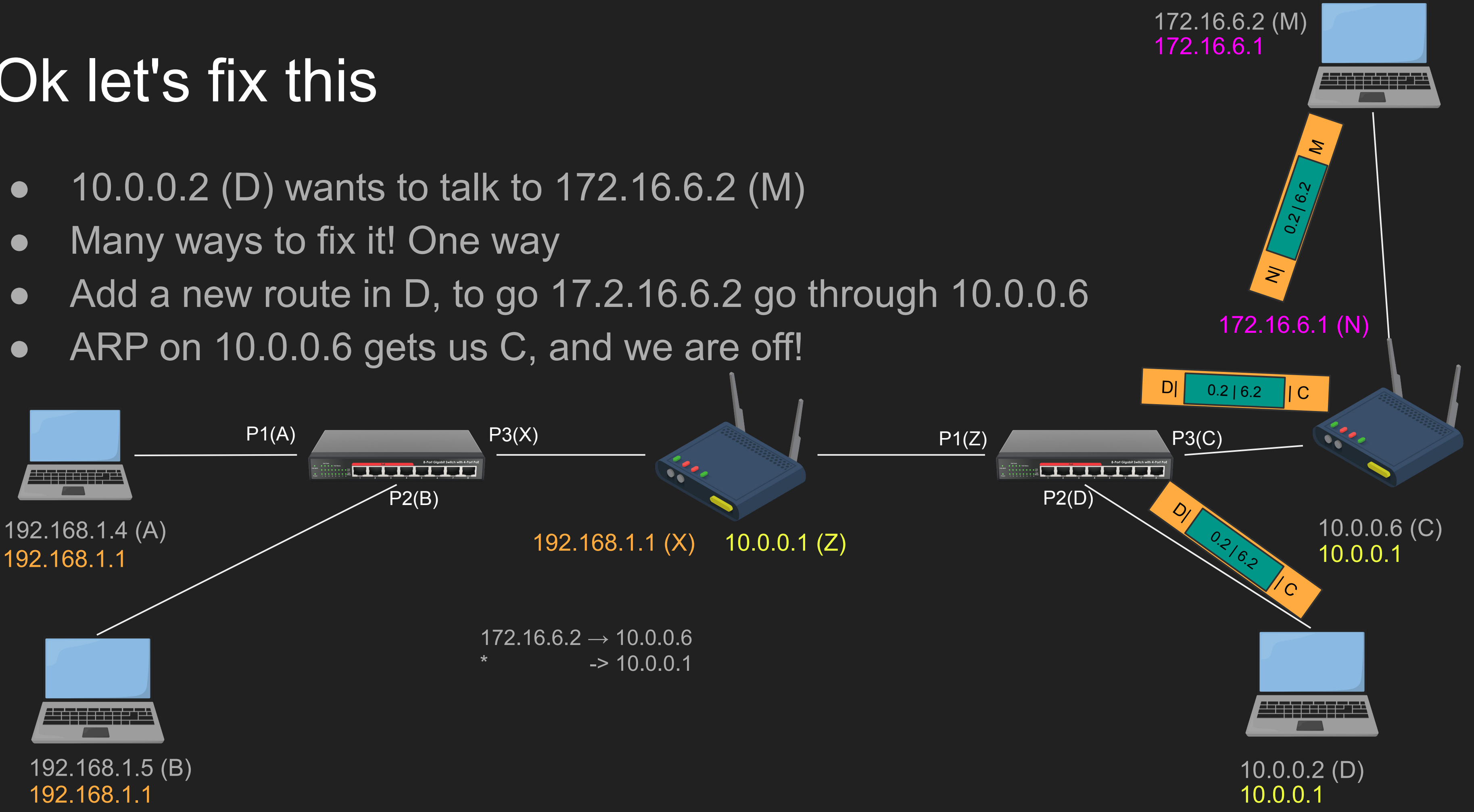
- Line 3: If you are going to any network
  - Send the frame gateway 192.168.1.1 (Default)
- Line 4: If you want to go to a machine on 192.168.1.0/24
  - => You can reach it directly through eth1 just ARP!
- Line 5: If you want to go to a machine on 10.0.0.0/24
  - => You can reach it directly through eth2 just ARP!



```
1 | Network        | Next hop    | Link? | Weight | NIC  |
2 |----------------|-------------|-------|--------|------|
3 | *              | 192.168.1.1 | F     | 10     | eth1 |
4 | 192.168.1.0/24 | *           | T     | 10     | eth1 |
5 | 10.0.0.0/24    | *           | T     | 1      | eth2 |
```

Ok let's fix this

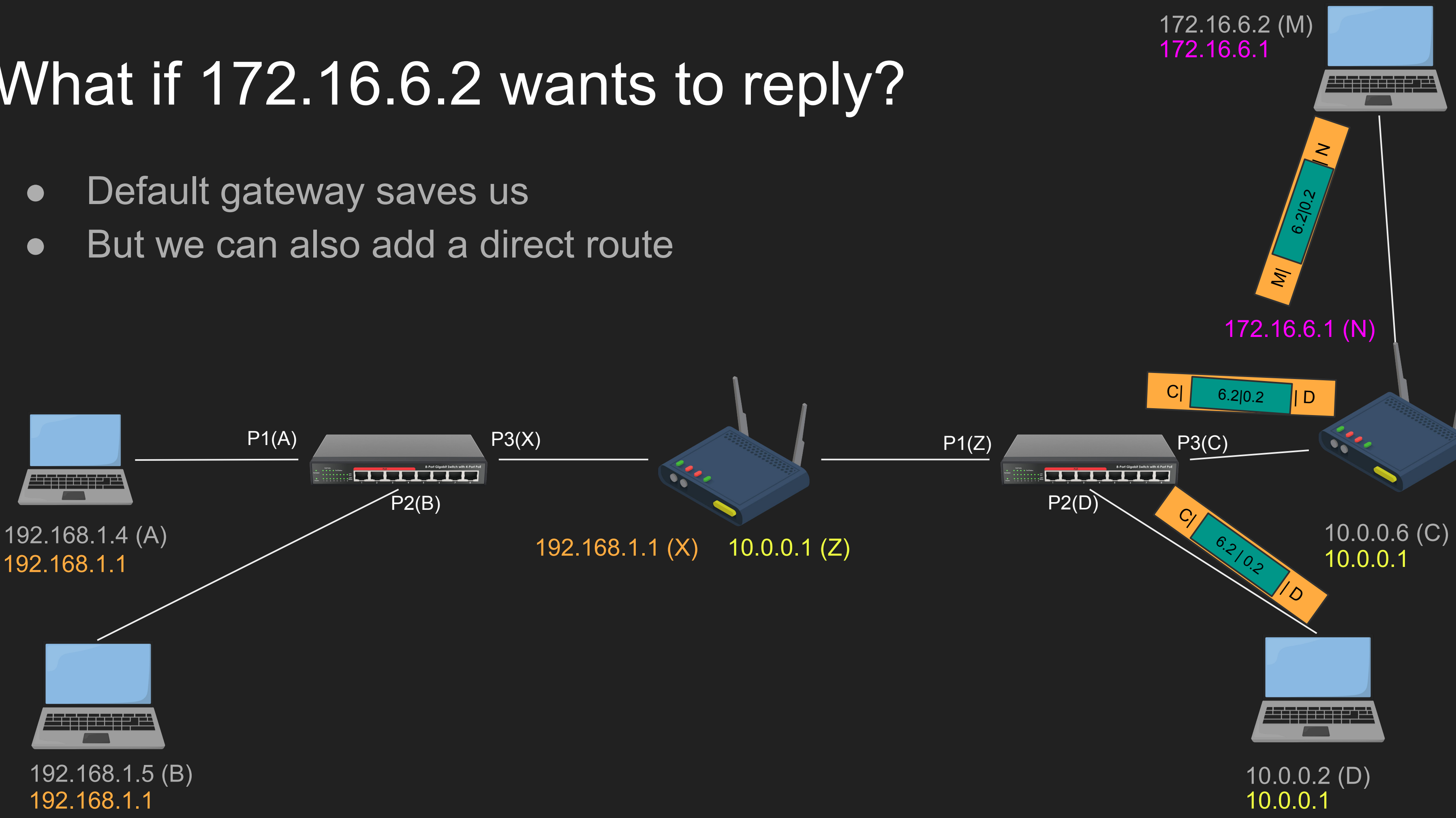
- 10.0.0.2 (D) wants to talk to 172.16.6.2 (M)
- Many ways to fix it! One way
- Add a new route in D, to go 17.2.16.6.2 go through 10.0.0.6
- ARP on 10.0.0.6 gets us C, and we are off!



172.16.6.2 → 10.0.0.6
* -> 10.0.0.1

What if 172.16.6.2 wants to reply?

- Default gateway saves us
- But we can also add a direct route



Who updates the Routing table

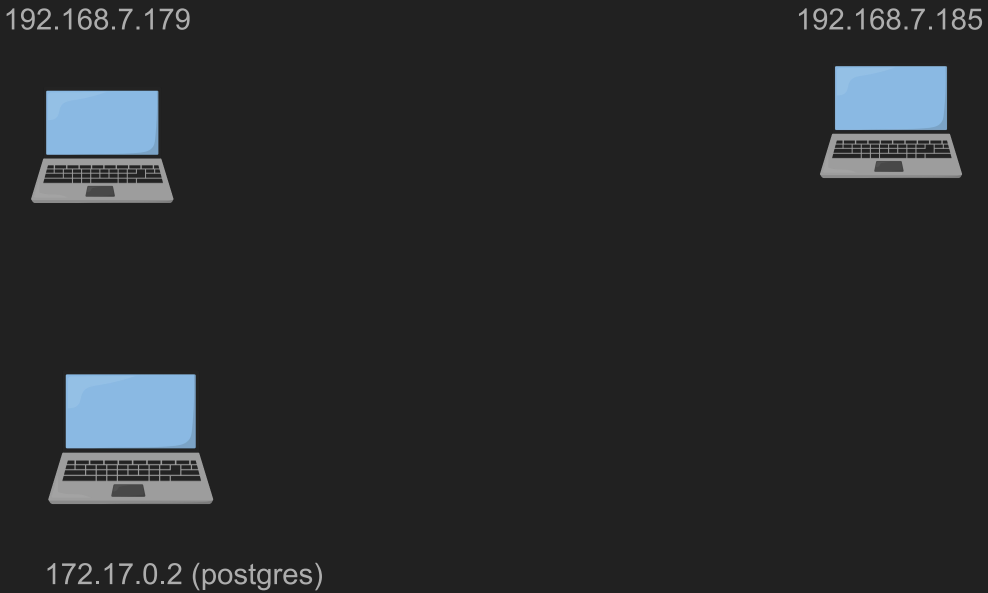
- Kernel, DHCP, other protocols like OSPF, BGP or even manually
- Kernel -> Direct network links and other administrative
- DHCP -> automatically detect and assign IPs and gateways
- OSPF -> multiple paths! Which one is the shortest?
- BGP -> On the Internet between ISPs
- Everything pours down eventually to the routing table

Demo

- List routes
  - On Windows -> route print
  - On Linux -> Ip route show
  - On mac -> netstat -rn
- Add a route
  - If you want to go google go through wifi

Summary

- Each device has a unique link address (Mac address) per network
- Devices can talk each other via data link and mac addresses
- Direct links can’t scale for large number of devices
- IP Address was invented
- Concept of a network was born
- Routing is the key for IP and networks



192.168.7.179

192.168.7.185

172.17.0.2 (postgres)