

Networking Fundamentals (II)

Subnetting

- When a packet arrives at L2 network, it can be delivered via MAC address using ARP table, but **how does the network decide whether an IP address belongs to a L2 network?**
 - Nodes on a local network (and only those) have IP addresses within a specific range.
 - For example, Colby has its own IP address ranges. All nodes connect to Colby network has IP addresses within that range.
 - In computer networks, the way to specify a range of consecutive IP addresses is via **IP Prefix**.
-

IP Prefix

- An IP prefix consists of a **subnet id** and a **prefix length (N)**
 - For example, the prefix for Colby network is **137.146.0.0/16**, the **subnet id** is **137.146.0.0** and the **prefix length** is **16**.
 - An **IP address belongs to a range** specified by a prefix **if the leftmost N bits of the IP address are the same as those in the subnet id**
 - E.g., IP address 137.146.183.19 belongs to the range 137.146.0.0/16, as the first 16 bits of the IP address are the same as those in the subnet id.
-

Subnet Mask

- Another way to specify **prefix length**.
- It is a 32 bit, IP address-like string. The **leftmost N bits** of the string are **1s**, and the remaining bits are 0s.
 - E.g., the subnet mask represents the prefix length /16 is
 - 1111 1111.1111 1111.0000 0000. 0000 0000 (in binary)
 - 255.255.0.0 (in decimal, **a typical representation**)
 - ff.ff.00.00 (in hexadecimal)
- Prefix length and subnet mask (netmask) are equivalent for all practical purposes:
 - E.g., **137.146.183.0 with net mask 255.255.255.0** is equivalent to **137.146.183.0/24**
 - But, subnet mask is easier for computers to conduct bit operations such as AND to decide whether a IP address belongs to the range or not.
- You will need to know subnet mask when you configure the network on some OS (e.g., Ubuntu).

Example for IP Address Calculations

- Given IP address **192.168.152.30** and prefix length **26**, answer the following questions.
 - Subnet ID: **192.168.152.0**
 - First host: **192.168.152.1**
 - Broadcast address: **192.168.152.63**
 - Last host: **192.168.152.62**
 - Subnet mask: **255.255.255.192**
- Subnet mask: the **leftmost 26 bits are 1s, and the rest bits are 0s**, which is 11111111.11111111.11111111.11 000000. In decimal, it's **255.255.255.192**.
- Subnet ID


```

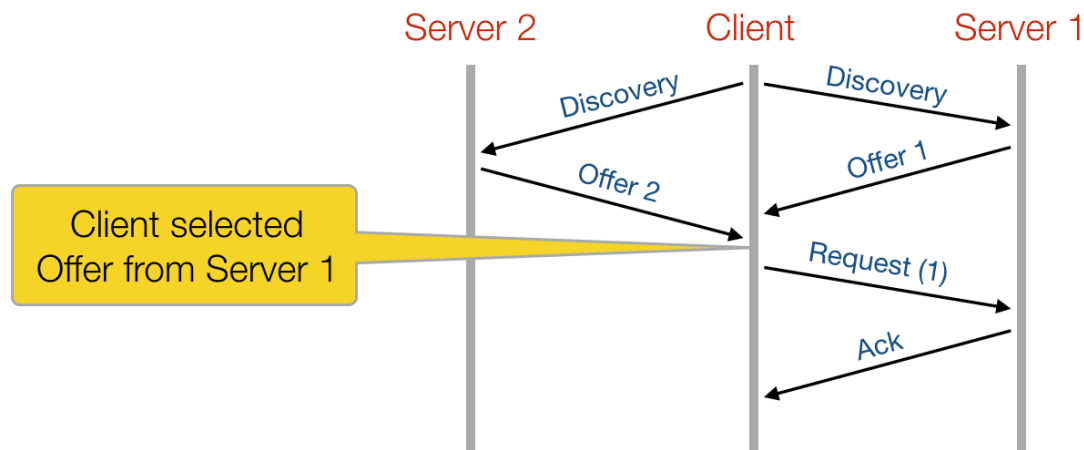
11000000.10101000.10011000.00 011110
AND 11111111.11111111.11111111.11 000000
-----
11000000.10101000.10011000.00 000000 -> 192.168.152.0

```
- Range represented by 192.168.152.0/26: start with 192.168.152.00 000000, end with 192.168.152.00 111111, which is 192.168.152.0 ~ 192.168.152.63
- **First host**: the **first address in the range that can be assigned to a host**, which is **subnet ID + 1**. In this question, it's **192.168.152.1**
- **Broadcast address**: the **maximum address in the range**, usually reserved for broadcasting in the subnet specified by the range. The maximum address in this range 192.168.152.0/26 is 11000000.10101000.10011000.00 111111 (**the rightmost 32-26 = 6 bits are all 1s, and the rest bits remain the same as what they were in subnet ID**). In decimal, the broadcast address is **192.168.152.63**.
- **Last host**: the **last address in the range that can be assigned to a host**, which is **broadcast address - 1**. In this question, it's **192.168.152.62**

DHCP

- We know that when a computer joins into a subset, it will be assigned a new IP address in the address range of that subnet without any human intervention. How does the network assign an IP address to a newly joined computer automatically?
- **Dynamic Host Configuration Protocol**
 - **application layer** protocol
 - plug-and-play or **zero-configuration** protocol, alleviate the workload of administrator
 - to obtain IP address and other configuration params (e.g., how long you can use the IP)

- one or more DHCP servers on the same subnet (to reduce congestions and improve reliability)
- utilizes IP broadcast (255.255.255.255) and UDP
- **Four steps** to acquire an IP address:
 - **Discovery**: find a DHCP server
 - **Offer**: proposed IP address and IP address lease time (several hours or days)
 - **Request**: choose from one of the offers, echo back the configuration parameters
 - **Ack**: confirm the request parameters



Network Layer Routing

- We know that when a packet gets into a L2 network, it can be delivered by using ARP and MAC address. Then, **how does a network send a packet to the subnet that contains the packet's destination IP address?**
- It uses network layer routing.
- Network layer routing is **hop-by-hop routing**. This means that you don't have to figure out the entire routing path from a source to a destination when sending the packet. You can send the packet to the next hop and ask that hop for further direction.
- A router usually connects different subnets.
- A network node maintains a **routing table** that **mapping next hops with prefixes** (aka subnets). So, it can decide which node is the next hop based on the prefixes.

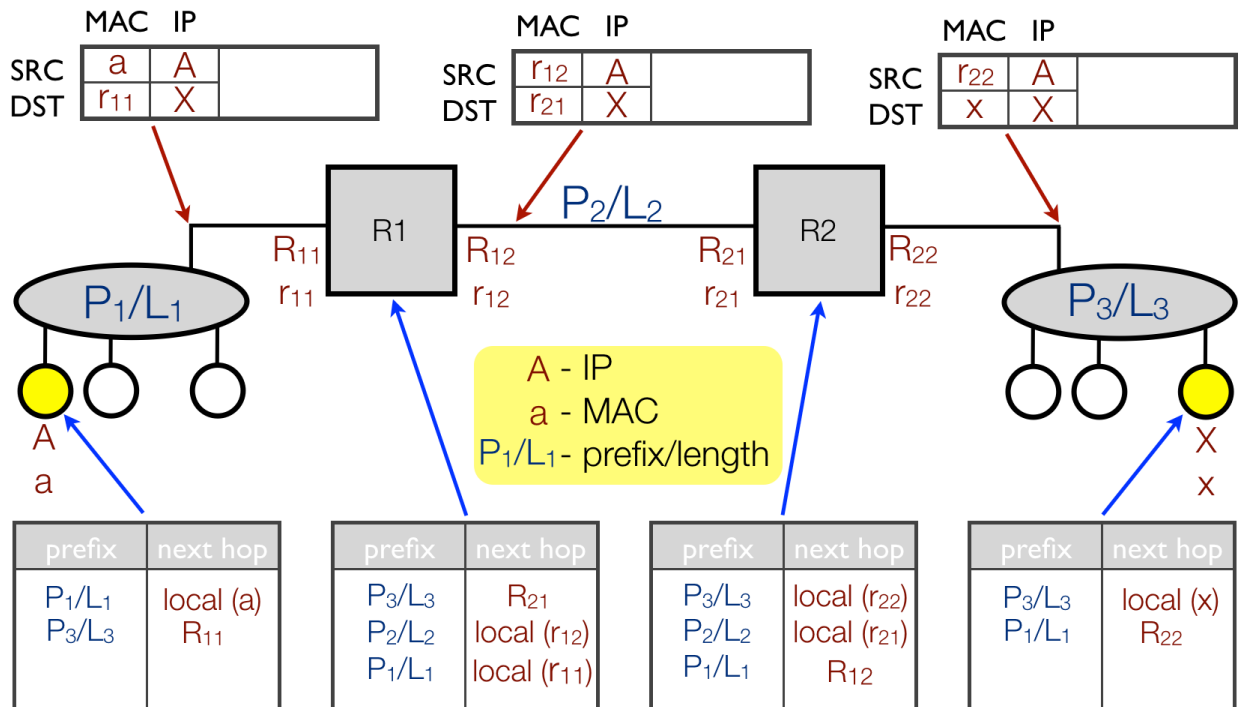
Routing Table

- What does a routing table look like? We can use the routing table on a computer for example.
 - The routing table of dori below indicates:
 - **137.146.183.0/24**: local delivery via interface eno1 (subnet for CS department)
 - use ARP to find the MAC address for the destination IP
 - **0.0.0.0/0** (everything else): send to 137.146.183.1 (gateway of CS dept.) via eno1
 - use ARP to find the MAC address for 137.146.183.1

```
dori:~> /sbin/route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          _gateway        0.0.0.0          UG    100    0      0 eno1
137.146.183.0    0.0.0.0         255.255.255.0    U      0      0      0 eno1
_gateway        0.0.0.0         255.255.255.255 UH    100    0      0 eno1
dori:~> /sbin/route -n [-n: do not resolve IP address]
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          137.146.183.1    0.0.0.0          UG    100    0      0 eno1
137.146.183.0    0.0.0.0         255.255.255.0    U      0      0      0 eno1
137.146.183.1    0.0.0.0         255.255.255.255 UH    100    0      0 eno1
```

- If the destination is in the connected subnet, use ARP to find the MAC address for the destination.
- If the destination is in another subnet, use ARP to find out the MAC address of the next hop (gateway or router). Once the packet reaches at the router then update the dest MAC.
- Please note that network routers usually have multiple interfaces, which is different from computers. Each interface connects to a different subnet and has its distinct IP and MAC.

Routing Example



- In the above example, node A in P1/L1 wants to send a packet to node X in P3/L3.
- When A sends the packet, the dest IP X is not in A's local network P1/L1. So, after using ARP if necessary, the dest MAC address of the packet will be set to r11, the MAC address of the interface of the router R1 that connects P1/L1.

- When the packet arrives at R1, R1 checks its routing table and finds packets for P3/L3 should go through R21. It then uses ARP if necessary, finds the MAC address of R21, and updates the dest MAC of the packet. Meanwhile, it also updates the src MAC to the MAC of its own interface R12. The reason is that it can provide hop-by-hop reliability if the packet is lost, which is more efficient than going back to the source and resend the packet.
- When the packet arrives at R2, R2 checks the routing table and finds that X belongs to P3/L3, its local network. It can then use ARP if necessary to find the MAC address of X, update the src and dest MAC of the packet, and sends the packet through R22.
- Please note that the src and dest IP are not changed during the entire process. The MAC addresses are updated every time when the packet is forward to the next hop.
- Please also note that it's important to [correctly configure the routing tables](#). If the first entry in R2's routing table is R12, then the packet won't be delivered to X, and it will be tossed back and forth between R1 and R2. This creates a [routing loop](#), which will leads to significant packet loss and degrade the network performance.

Disclaimer: Notes adapted from the textbook and online resources.