Subneting

Example

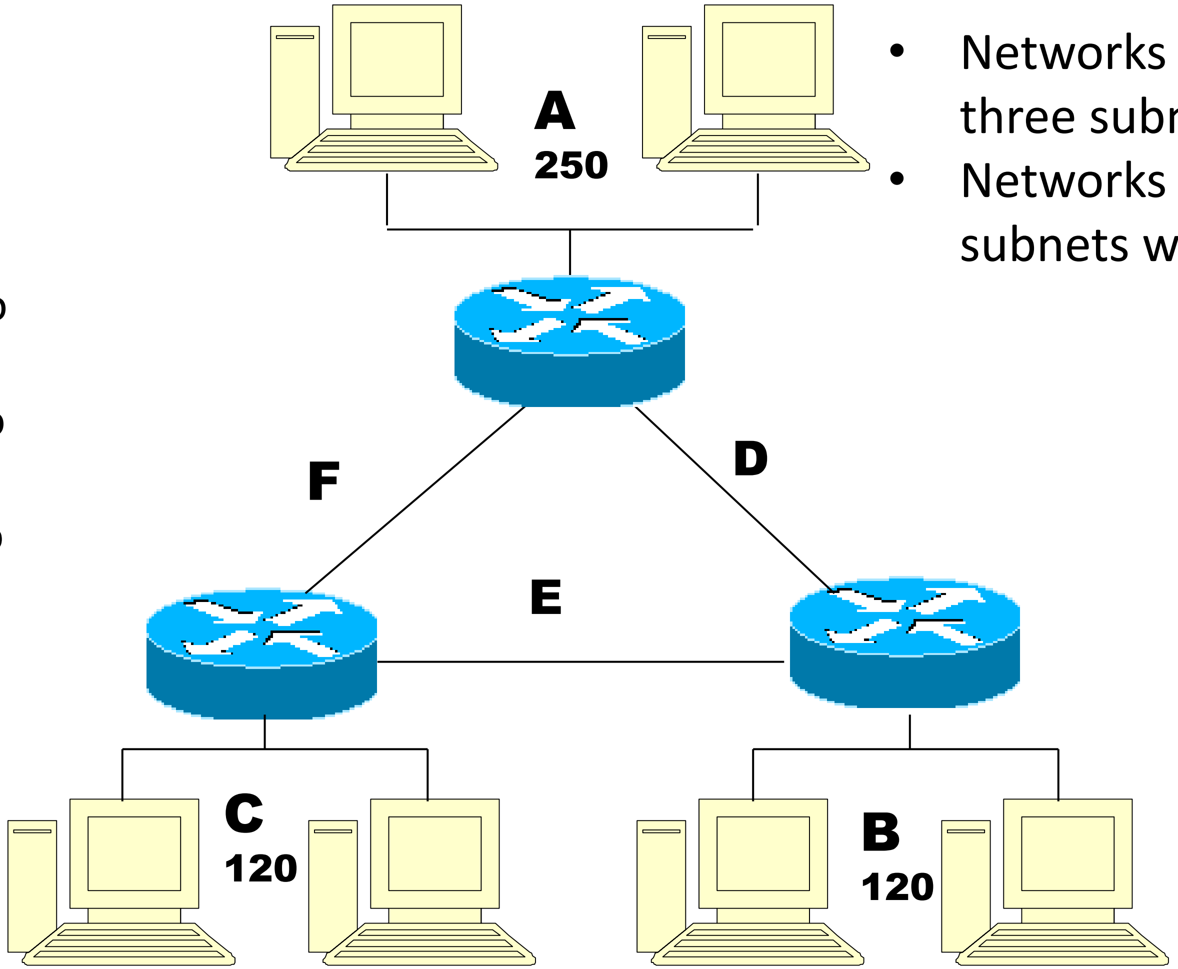
Consider the following topology:

1. There is a pull of addresses: **222.55.254/23;**

This unique network can host 2^9-2 =510 nodes
The request is to create 6 sub-networks (A…F) connected by 3 routers

- Assign network addresses to each of these six subnets, with the following constraints:
  - **Subnet A** should have enough addresses to support **250** interfaces;
  - **Subnet B** should have enough addresses to support **120** interfaces and
  - **Subnet C** should have enough addresses to support **120** interfaces.
  - **Subnets D,E and F** should each be able to support **2** interfaces.

  For each subnet, the assignment should take the form **a.b.c.d/x**
- Provide forwarding tables (using longest prefix matching) for each of the three routers



- Networks A, B and C three subnets with hosts.
- Networks D, E and F the subnets without hosts.

Solution:

1) We have to assign IP addresses for 6 subnets from 222.55.254/23.

222 . 55 . 254 .

11011110 00110111 1111111* *********

The /23 bits long subnet mask allows us to use:

32-23 = 9 bits for our networks and hosts (the asterisks).

This means we can have a maximum of **2^9 = 512** addresses.

Let's make a simple calculation:

- **250** addresses for subnet A,
- **120** addresses for B,
- **120** addresses for C,
- **2** addresses for D,
- **2** addresses for E and
- **2** addresses for E.

This totals **496** addresses, thus **512** should be enough.

subnet A (250 hosts)

- Let's start with ***subnet A***, which should support **250** addresses.
  For this we need 8 bits in the host part ($2^8 = 256$).
  We have two choices:

11011110 00110111 11111110 ********

or

11011110 00110111 11111111 ********

- Let's choose the second one: 11011110 00110111 11111111 ********
  That is **222.55.255/24.**
- Note that we have already used half of the address space, which leaves us with only 256 addresses **for the rest of the subnets**.

Subnet B (120 hosts)

Subnet B has **120** addresses; thus we need 7 bits (**2^7 = 128).** The addresses we haven't used so far are:

11011110 00110111 11111110 *********

Again we have two choices:

11011110 00110111 11111110 0*******

or

11011110 00110111 11111110 1*******

Let's take the first one: **11011110 00110111 11111110 0*******

That is **222.55.254.0/25.**

Now we are left with a quarter of the address space, which leaves us with only **128** addresses for the rest of the subnets.

Subnet C (120 hosts)

Subnet C again has **120** addresses, so we need 7 bits (**2^7 = 128**).

The addresses we haven't used so far are:

1011110 00110111 11111110 1*******

So there is only one choice: to use this one.

So for subnet C we will allocate

1011110 00110111 11111110 1*******

That is **222.55.254.128/25**

Now we have used the entire address space.

Apparently, we have reached a dead end, as we have no space for subnets D, E, F.

However, we still have room in the address space allocated for networks B and C. They only use **120** addresses each, but we have allocated **128** addresses.

Consequently, we can fit **8 x 2 = 16** addresses in the remaining space.

As networks D, E, F have only 2 addresses each, we are out of trouble.

Preparing address space for D, E and F

Let's separate these addresses from B.

8 addresses means 3 bits (2^3 = 8). Let's use the last 3 bits from the address spaces:

11011110 00110111 11111110 0*******

We have one decision to make: where to locate these space, or what to put instead of the rest of the asterisks. Let's choose 0000. Thus we will have:

11011110 00110111 11111110 00000***

We have managed to subtract **222.55.254.0/29** from the address space initially allocated to subnet B.

Subnets D (2 hosts) , D (2 hosts), D (2 hosts)

To sum up, we have so far:

Subnet A: 222.55.255/24 (256 addresses)

Subnet B: 222.55.254.0/25 - 222.55.254.0/29 (128-8 = 120 addresses)

Subnet C: 222.55.254.128/25 (128 addresses)

For each of the subnets, we only need 2 addresses, or 1 bit.

We can thus take the freed address space 222.55.254.0/29 and divide it among these subnets:

Initial :	11011110 00110111 11111110 00000***
Subnet D:	11011110 00110111 11111110 00000000*
Subnet E:	11011110 00110111 11111110 00000001*
Subnet F:	11011110 00110111 11111110 0000001**

Please note that for subnet F we have allocated 4 addresses, although we only needed 2. We could have chosen 11011110 00110111 11111110 0000010*, as well.

This way we could have saved 2 addresses for future use.

Subnet D, E and F

Thus for the last 3 subnets we have allocated addresses as follows:

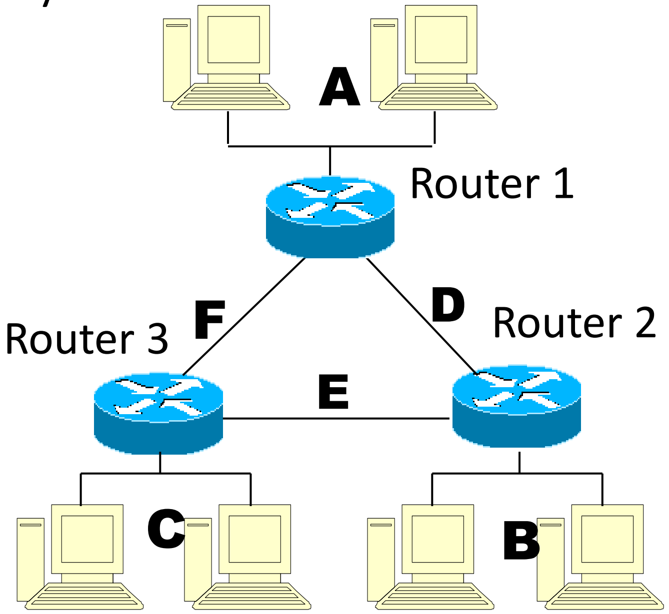
Subnet D: 222.55.254.0/31 (2 addresses)

Subnet E: 222.55.254.2/31 (2 addresses)

Subnet F: 222.55.254.4/30 (4 addresses) provide forwarding tables (using longest prefix matching) for each of the three routers

Subnet A:	11011110 00110111 11111111 ********	Subnet D:	11011110 00110111 11111110 0000000*
Subnet B:	11011110 00110111 11111110 0*******	Subnet E:	11011110 00110111 11111110 0000001*
Subnet C:	11011110 00110111 11111110 1*******	Subnet F:	11011110 00110111 11111110 000001**

To simplify the solution, assume that no datagrams have router interfaces as ultimate destinations.
Also, label D, E, F for the upper-right, bottom, and upper-left interior subnets, respectively.



Router 1

Longest Prefix Match	**Outgoing Interface**
11011110 00110111 11111111 ********	Subnet A (/24)
11011110 00110111 11111110 0000000*	Subnet D (/31)
11011110 00110111 11111110 000001**	Subnet F (/30)

Router 2

Longest Prefix Match	Outgoing Interface
11011110 00110111 11111110 0000000*	Subnet D (/31)
11011110 00110111 11111110 0*******	Subnet B (/25)
11011110 00110111 11111110 0000001*	Subnet E (/31)

Router 3

Longest Prefix Match	Outgoing Interface
11011110 00110111 11111110 000001**	Subnet F (/30)
11011110 00110111 11111110 0000001*	Subnet E (/31)
11011110 00110111 11111110 1*******	Subnet C (/25)

Note:

- Generally, 31 prefix are not used, because it imply just the network address and broadcast address.
- On point to point links it is possible to use.
- For details please read RFC3021 .
  - It describes using 31-bit prefixes for point-to-point links.
- This leaves 1 bit for the host-id portion of the IP address.
  - Normally a host-id of all zeros is used to represent the network or subnet, and a host-id of all ones is used to represent a directed broadcast.
- Using 31-Bit prefixes, the host-id of 0 represents one host, and a host-id of 1 represents the other host **of a point-to-point link**.
- We used this facility in our solution.