Homework 1 :

EP2120 InternetWorking

Pierre FLEITZ

1. IPv4 Addressing (30/100) :

a) -> The best fit netmask for a network with 255 hosts is : **255.255.254.0**
Because you gonna need 1 address for the network, one broadcast address and then 255 hosts address, so 257 adresses, therefore 255.255.254.0 is the best fit netmask for this network.

b) -> The maximum number of host we can have in a /14 network is : (2^ (32-14)) – 2 = **262142**

c) -> We have the network 93.113.245.224/27 in four equally sized /29 networks :

1) 93.113.245.224/29
2) 93.113.245.232/29
3) 93.113.245.240/29
4) 93.113.245.248/29

d) -> The directed broadcast address of the network 84.129.75.113/24 is : **84.129.75.255**

e) -> The directed broadcast address of the network 84.129.75.113/24 is : **255.255.255.255**

f) -> According to IANA services IP network 195.74.160.0/19 belongs to ESA (European Space Agency) and their AS number is : 1Z8519

2. Address allocation (30/100) :

➔ **Network A :** We gonna need 500 hosts, so network A is going to be a /23 network.
@NetworkA = 72.33.144.0/23 -- **@BroadcastA = 72.33.145.255/23**
We definitely have enough adresses for 500 hosts, and 10 free addresses.

➔ **Network B :** Just like for NetworkA we gonna need 500 hosts :
@NetworkB = 72.33.146.0/23 -- @BroadcastB = 72.33.147.255/23

➔ **Network F :** This time we need to be able to reach 3 places (NetworkA, NetworkB, Switch R2), therefore we gonna need 3 hosts minimum then the network F is going to be a /29 network.
@NetworkF = 72.33.148.0/29 -- @BroadcastF = 72.33.148.8/29

➔ **Network C :** Here we gonna need 300 hosts, so network B is going to be a /23 network.
@NetworkC = 72.33.150.0/23 -- @BroadcastC = 72.33.151.255/23

➔ **Network D :** Same than for networkC
@NetworkD = 72.33.152.0/23 -- @BroadcastD = 72.33.153.255/23

➔ **Network E :** Same again.
@NetworkE = 72.33.154.0/23 -- @BroadcastE = 72.33.155.255/23

➔ **Network G :** This network needs to be able to talk to 4 differents networks, therefore it needs at least 4 hosts address. Then it's gonna be a /29 network.
@NetworkG = 72.33.156.0/29 -- @BroadcastG = 72.33.156.8/29

➔ **Network H :** This network needs to be able to talk to 3 differents networks (F through R2, G through R3 and ext. Network through R1) therefore it needs at least 3 hosts address. Then it's gonna be a /29 network.
@NetworkH = 72.33.158.0/29 -- @BroadcastH = 72.33.158.8/29

➔ **Conclusion :**
What we did is that we split the network in 2 /21 blocks :
- block 1 = 72.33.144.0/21 reachable through R2
- block 2 = 72.33.150.0/21 reachable through R3

➔ **Then the routing table of R1 is :**

Destination	Next hop	Flags	Interface
72.33.144.0/21	R2	UG	m1
72.33.150.0/21	R3	UG	m2
EXT	-	U	m0

3. IPv4 forwarding (20/100) :

a) -> Next hop : 249.21.118.102 – interface : m2

b) –> Next hop : 7.255.85.122 – interface : m0

c) -> Next hop : None – interface : m2

d) -> Next hop : 7.255.85.90 – interface : m0

e) -> Next hop : None – interface : m0

4. IPv4 options (20/100) :

a) -> They are simple options. The **No-options** option can be used to align the next option on a 16-bit ou 32-bit boundary ; on the other hand **End-of-options** option is used for padding at the end of the option field, but it only can be the last option !
Wich means that if more than 1 byte is needed to align the option field, some No-options options **must** be used, followed by an End-of-options option.

b) -> It can "only" list up 9 route IP addresses because the maximum size of the header is 60 bytes, wich must include 20 bytes for the base header.
This implies that only 40 bytes are left over the option part.

c) -> **Strict Source Route** and **Loose Source Route** options are options that are copied in every fragments. Simply because for the example of the Strict Source Route, this option is used to route the IP packet based on information supplied by the source, so every fragments needs to keep this option.

d) -> The advantages of IPv6 extension header :
- The header size is fixed, so the IP Header Length is useless and the calculations are easier than in IPv4.
- A lot of useless options have been deleted, added only when needed.
- The options are supported outside the header, so easily adaptable protocols (modular upgradability).