Homework #1 for protok11
Peter Boström <pbos@kth.se>
2011-09-04

1. IPv4 Addressing

a) Propose the best fit netmask (i.e. resulting in as few host addresses as possible) for a network with 15 hosts in it!

This network will need room for 15 hosts, however as two addresses of every network larger than /31 are reserved for broadcast and to identify the network itself, using a /28 network would yield only 14 addresses for use. Therefore a /27 network is required, which yields 32-2=30 addresses for use by hosts in the network.

A /27 network has the netmask: 255.255.255.224

b) How many hosts can you have in a /26 network?

A /26 network contains 64 addresses of which 2 are reserved for broadcast and to identify the network itself. 62 of these addresses can be used by hosts in the network.

c) Split up the network 192.168.48.0/23 into four equally sized /25 networks!

192.168.48.0/25
192.168.48.128/25
192.168.49.0/25
192.168.49.128/25

d) What is the directed broadcast address of the network 121.2.20.0/22?

121.2.23.255

e) What is the limited broadcast address of the network 121.2.20.0/22?

255.255.255.255

f) Use the services of IANA and a regional registry to figure out to whom the network 130.237.0.0/16 belongs. Provide the name of the organization and the AS number.

SUNET/NORDUnet, AS1653

2. Address allocation

Make a network on 12.128.0.0/17.

1) The local offices require 900 addresses each. Create a minimal block for each local office A to L. Start with the lowest address for network A.

We'll allocate 1024-blocks for each local office. That's a /22 subnet for each office.

2) There are no unnumbered point-to-point links: all Ethernet networks have IP sub-networks and all nodes (routers and hosts) have an IP address on all their network interfaces.

(M, N, O, P) need IP addresses from R1 as well.

To aggregate and include three /22 well, these'll be /20 networks. M's network will encompass A, B and C, etc.

3) The address allocation should be such that the sub-networks can be aggregated. The central router R1 should have at most eight entries in its forwarding table (plus possibly a default route).

```
    M: 12.128.0.0/20
       A: 12.128.0.0/22
       B: 12.128.4.0/22
       C: 12.128.8.0/22

    N: 12.128.16.0/20
       D: 12.128.16.0/22
       E: 12.128.20.0/22
       F: 12.128.24.0/22

    O: 12.128.32.0/20
       G: 12.128.32.0/22
       H: 12.128.36.0/22
       I: 12.128.40.0/22

    P: 12.128.48.0/23
       J: 12.128.48.0/22
       K: 12.128.52.0/22
       L: 12.128.56.0/22


    Q: 12.128.64.0/30
       M: 12.128.64.1
       Computer: 12.128.64.2
    R: 12.128.64.4/30
       N: 12.128.64.5
       Computer: 12.128.64.6
    S: 12.128.64.8/30
       O: 12.128.64.9
       Computer: 12.128.64.10
    T: 12.128.64.12/30
       P: 12.128.64.13
       Computer: 12.128.64.14

    The routing table for R1 will include M, N, O, P and the directly
    connected computers.

    Routing table (Destination, Next hop, Flags, Interface):
    12.128.64.0/30, -, U, m0
    12.128.64.4/30, -, U, m1
    12.128.64.8/30, -, U, m2
    12.128.64.12/30, -, U, m3
    12.128.0.0/20, 12.128.64.1, UG, m0
    12.128.16.0/20, 12.128.64.5, UG, m1
    12.128.32.0/20, 12.128.64.9, UG, m2
    12.128.48.0/20, 12.128.64.13, UG, m3

3. IPv4 forwarding

    a) 167.14.81.9

    No prefix matches 167. so default route, next hop is 93.171.89.1

    b) 192.121.17.13

    Matches 192.121.17.13/32, next hop is 111.92.92.127.

    c) 7.152.35.14

    This fucker is down, either fall back to default and send to 93.171.89.1
    or drop that shit.

    d) 111.92.66.178

    Matches 111.92.64.0/18, direct delivery.
```

e) 133.10.165.254

Matches 133.10.165.0/24, next hop is 62.88.93.33.

4. IPv4 options and IPv6 extension headers

Connected with IPv4 and IPv6.
Path MTU on both networks: 100,000 octets.
Would like to send a datagram of 120,000 octets.
(Check RFC2460 and RFC2675 to see how fragmentation and jumbo payloads can/can not coexist in IPv6.)

a) Assume you choose the IPv4 network. Do the IP datagrams need to carry IPv4 options? If yes, how many?

The MF (more fragments) flag for all but the last packet and a non-zero fragmentation offset for all but the first packet. Fragmentation specifies an 8-octet offset from the original payload. The maximum offset we can have is $(2^{13}-1)*8 = 65528$ octets. Max size for one packet is $2^{16}-1 = 65535$ octets, so the largest possible ipv4 datagram can contain is $65528+65535 = 131063$ octets. This is enough to store our datagram. How lucky!

The largest possible IPv4 packet fits within a transmission, so we'll have two packets sent. The last packet will have a large fragmentation offset, and the first one will have the MF flag set and a fragmentation offset of zero.

This does not require additional options, so we can have 20-octet headers, which gives us a 40-octet "header cost".

b) Assume you choose the IPv6 network. Do the datagrams need to carry IPv6 extension headers? If yes, how many? How many options would they carry?

Jumbograms will not help, we can split into a fragmented package just the same way as IPv4 and still fit the load. We can't send a fragmented jumbogram, which in this case wouldn't even help, and as the datagram is larger than the MTU, a jumbogram with it would not fit. So we need 2 packets here as well. And it just about fits, analogous to how IPv4 worked out.

To use fragmentation in IPv6, a fragmentation header needs to be included which includes offset, a M flag (more fragments flag, again) and also an identifier, as these could be sent asyncroneously. This is an 8-octet header required in both two packets, which gives a total header size of 48 octets, or 96 for both.

c) Which network should you choose to minimize the total number of octets sent in datagram headers?

IPv4, (40<96) though it isn't a large cost either way.