Solution to COMP5311 Assignment 1

QING Pei, 11500811G

October 16, 2011

1 Problem 1

1.1 a

160.10.32.0/20

Convert 160.10.47.255 to binary format:
`10100000.00001010.00101111.11111111`

The last 12 bits are all 1s, therefore, it could be a subnet-directed broadcast address for a /20 subnet. /24 is impossible because otherwise there must exist a subnet mask being 00101111. But in fact there is no such subnet. The same to /28 subnets. The network mask is:
`10nnnnnn.nnnnnnnn.ssssHHHH.HHHHHHHH`

Therefore the subnet is 10100000.00001010.00100000.00000000/20, which in decimal format is 160.10.32.0/20.

1.2 b

160.10.130.16/28

160.10.130.31 in binary format is:
`10100000.00001010.10000010.00011111`

The last 5 bits are 1s. So it might belong to a /28 subnet. The mask should be:
`10nnnnnn.nnnnnnnn.ssssssss.ssssHHHH`

Therefore the subnet is 10100000.00001010.10000010.00010000/28, which in decimal format is 160.10.130.16/28.

1.3 c

No.

160.10.127.255 in binary format it:
`10100000.00001010.01111111.11111111`

The last 15 bits are 1s. If it is a subnet-directed broadcast address, there should be some subnet which is 10100000.00001010.01110000.00000000/20. That is 160.10.112.0/20 in decimal format. But in the network structure, this subnet does not exist.

2 Problem 2

2.1 a

The second fragment is 100 bytes in size.

```
| IP | data |
| 20 | 1480 |
```
The original IP packet is 20+1480 bytes. Before tunneling, this should be fragmented so that after encapsulating each fragment with an outer IP header and a tunnel header, the packet will not exceed the physical MTU.

The tunnel MTU is 1480, then the payload should be within 1460 bytes. So the original packet is broken into 20+1440 and 20+40 byte fragments:
```
| IP0 | data |
| 20 | 1440 |
```
and
```
| IP1 | rest of data |
| 20 | 40 |
```

After encapsulation, the packet would become:
```
| IP | Tunnel | IP0 | data |
| 20 | 20 | 20 | 1440 |
```
and
```
| IP | Tunnel | IP1 | rest of data |
| 20 | 20 | 20 | 40 |
```

2.2 b

The fragments will be reassembled at the destination D if no further fragmentation occurs inside the tunnel.

Since the outer IP header does not show that there is any fragmentation, each hop in the tunnel including the tunnel source will just pass the packets to the next hop until they arrive at tunnel endpoint. The tunnel endpoint will decapsulate the packets and send the packet (IP0+data) and (IP1+rest of data) to D. D will check the header and reassemble the fragments.

2.3 c

At the tunnel endpoint.

Along the tunnel, every packet's destination IP in the outer IP headers is the IP of tunnel endpoint. If there is no more tunneling inside the outer tunnel, all the packets are supposed to be sent to tunnel endpoint and that is the only hop in the tunnel ensured to take care of all the fragments. Any middle point will just pass the packet to the next hop, and it does not care whether it is a fragmented packet or not.

3 Problem 3

a The number before @ represents the payload length in that fragment.

b The number after @ represents the offset of that fragment. The offset being displayed here is already processed and the unit is **byte**.

c The + symbol means that the More-Fragment flag is true. The current fragment is not the last one.

d The number 6144 is the id of the fragmented datagram. Fragments with the same id belong to the same IP datagram before fragmentation.

e The size is $65120+398+20 = 65538$ bytes. 65120 is the offset (in bytes) of the last fragment. 398 is the payload length of the last fragment. 20 is the IP header length.

4 Problem 4

Assume there are N hosts in subnet 158.132.1.0/24 and M hosts in subnet 158.132.2.0/24.

4.1 a

158.132.1.100 and the router are attacked.

1. 158.132.1.1, pretending to be 158.132.1.100, sends a echo request message to limited broadcast address which can go through the router to subnet 158.132.2.0/24.

2. The M+N hosts in both subnets and the router reply with an echo reply message to 158.132.1.100.
3. The router has to process all the M packets from subnet 158.132.2.0/24 and sent them to 158.132.1.100.
4. 158.132.1.100 received a large number(M+N+1) of echo reply messages and becomes busy. If the number of hosts are large enough, the NIC would fail to process normal packets sent to it later.

4.2 b

158.132.2.100 and the router are attacked.

1. 158.132.1.1, pretending to be 158.132.2.100, sends a echo request message to limited broadcast address which can go through the router to subnet 158.132.2.0/24.
2. The M+N hosts in both subnets and the router reply with an echo reply message to 158.132.2.100.
3. The router has to process all the N packets from subnet 158.132.1.0/24 and sent them to 158.132.2.100.
4. 158.132.2.100 received a large number(M+N+1) of echo reply messages and becomes busy. If the number of hosts are large enough, the NIC would fail to process normal packets sent to it later.

4.3 c

Everything is fine.

1. 158.132.1.1, pretending to be 158.132.2.100, send a echo request message to a subnet-directed broadcast address 158.132.2.255.
2. All the hosts in 158.132.1.0/24 will drop the packet.
3. The router is set to drop the packet.
4. Nothing more happens. The hosts in 158.132.2.0/24 would not even know there was an attempted attack.

5 Problem 5

5.1 a

Fragment 1 total length=1500 bytes, offset=0, MF=1

Fragment 2 total length=1500 bytes, offset=185, MF=1

Fragment 3 total length=40 bytes, offset=370, MF=0

The total length of the IP packet is 3000 bytes. The header length is 20 bytes. So the payload length is 2800 bytes. The maximum payload length is $\lfloor (1500 - 20)/8 \rfloor * 8$ bytes, which is 1480 bytes. In a network with an MTU of 1500, the payload is broken into two 1480-byte and one 20-byte fragments. Each fragment will have a new header of 20 bytes.

5.2 b

The total length of the last fragment will become 140 bytes. No other changes.

The only change is the payload length. Instead of a 1480+1480+20 fragmentation, the new one would be 1480+1480+120. The first two fragments are the same, the last one has a larger payload.

5.3 c

Fragment 1 total length=1496 bytes, offset=0, MF=1

Fragment 2 total length=1496 bytes, offset=182, MF=1

Fragment 3 total length=88 bytes, offset=364, MF=0

If the header length is 40 bytes. In a network with an MTU of 1500. The maximum payload length is $\lfloor (1500 - 40)/8 \rfloor * 8$ bytes, which is 1456 bytes. Then the 2960-byte payload is broken into 1456+1456+48 bytes. A 40-byte header is generated for each fragment.