Shivam Patel

COMP 343

Dordal

Assignment 5- Chap 7: 2,6,7,8 ; Chap 9: 1,3

2. How can a receiving host tell if an arriving IPv4 packet is unfragmented? Hint: such

a packet will be both the “first fragment” and the “last fragment”.

If the arriving packet does not have a matching IDENT field with other incoming packets and the same source and destination address the receiver will not have any fragments to assemble. Also if it is the “first fragment” the fragment offset equals zero and if it’s the last fragment is 0 then with both these criteria true the packet will be unfragmented.

6. For each IPv4 network prefix given (with length), identify which of the subsequent

IPv4 addresses are part of the same subnet.

1. **10.0.130.0/23**: 10.0.130.23, 10.0.129.1, 10.0.131.12, 10.0.132.7

130 --> 1000001|0

**130 --> 1000001|0 This Matches**

129 --> 1000000|1 does NOT match

**131 --> 1000001|1 This Matches**

132 --> 1000010|0 does NOT match

1. **10.0.132.0/22**: 10.0.130.23, 10.0.135.1, 10.0.134.12, 10.0.136.7

132 --> 100001|00

130 --> 100000|10 does NOT match

**135 --> 100001|11 This Matches**

**134 --> 100001|10 This Matches**

136 --> 100010|00 does NOT match

1. **10.0.64.0/18**: 10.0.65.13, 10.0.32.4, 10.0.127.3, 10.0.128.4

64 --> 01|00 0000

**65 --> 01|00 0001 This Matches**

32 --> 00|10 0000 does NOT match

**127 --> 01|11 1111 This Matches**

128 --> 10|00 0000 does NOT match

1. **10.0.168.0/21**: 10.0.166.1, 10.0.170.3, 10.0.174.5, 10.0.177.7

168 --> 10101|000

166 --> 10100|110 does NOT match

**170 --> 10101|010 This Matches**

**174 --> 10101|110 This Matches**

177 --> 10110|001 does NOT match

1. **10.0.0.64/26**: 10.0.0.125, 10.0.0.66, 10.0.0.130, 10.0.0.62

64 --> 01|00 0000

**125 --> 01|11 1101 This Matches**

**66 --> 01|00 0010 This Matches**

130 --> 10|00 0010 does NOT match

62 --> 00|11 1110 does NOT match

7. Suppose that the subnet bits below for the following five subnets A-E all come

from the beginning of the fourth byte of the IPv4 address; that is, these are s

subnets of a /24 block.

* A: 00
* B: 01
* C: 110
* D: 111
* E: 1010

1. What are the sizes of each subnet, and the corresponding decimal ranges? Count the addresses with host bits all 0’s or with host bits all 1’s as part of the subnet.

A: 00\_ \_ \_ \_ \_ \_ first two bytes of fourth byte

This is a /26:

Decimal Range: 0 to 63

Total IP’s = 64

B: 01\_ \_ \_ \_ \_ \_ first two bytes of fourth byte

This is a /26:

Decimal Range: 64-127

Total IP’s = 64

C: 110\_ \_ \_ \_ \_ first three bytes of fourth byte

This is a /27

Decimal Range: 192-223

Total IP’s = 32

D: 111\_ \_ \_ \_ \_ \_ first three bytes of fourth byte

This is a /27

Decimal Range: 224-255

Total IP’s = 32

E: 1010 \_ \_ \_ \_ first four bytes of fourth byte

This is a /28

Decimal Range: 160-175

Total IP’s = 16

1. How many IPv4 addresses in the class-C block do not belong to any of the subnets A, B, C, D and E?

C Block is size /24 --> 256 Total IP’s

256 – 64 – 64 – 32 – 32 – 16 = **48 IPv4 Addresses do not belong to any subnets A,B,C,D, and E.**

8. In [7.8   Address Resolution Protocol: ARP](http://intronetworks.cs.luc.edu/current/html/ipv4.html#address-resolution-protocol) it was stated that, in newer

implementations, “repeat ARP queries about a timed out entry are first sent

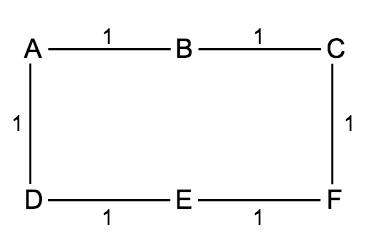
unicast”, in order to reduce broadcast traffic. Why is this unicast approach likely to

succeed most of the time? What would have to happen to create a situation where

the unicast query fails, but a follow up broadcast succeeds?

Because most of the time sites just don’t change their physical address. So If a node is just turned off, then there is a time out, causing the unicast to fail ultimately resulting in the broadcast to fail. But if a node changes its physical address the unicast query will fail but the broadcast will succeed.

**CHAP 9**

1. ****Suppose the network is as follows, where distance-vector routing update is used. Each link has cost 1, and each router has entries in its forwarding table only for its immediate neighbors (so A’s table contains ⟨B,B,1⟩, ⟨D,D,1⟩ and B’s table contains ⟨A,A,1⟩, ⟨C,C,1⟩).
2. Suppose each node creates a report from its initial configuration and sends that to each of its neighbors. What will each node’s forwarding table be after this set of exchanges? The exchanges, in other words, are all conducted simultaneously and in parallel; no report contains new information learned by a router as part of this process.

A: <B,B,1> <D,D,1> <C,B,2> <E,D,2>

B: <A,A,1> <C,C,1> <D,A,2> <F,C,2>

C: <B,B,1> <F,F,1> <A,B,2> <E,F,2>

D: <A,A,1> <E,E,1> <B,A,2> <F,E,2>

E: <D,D,1> <F,F,1> <A,D,2> <C,F,2>

F: <E,E,1> <C,C,1> <D,E,2> <B,C,2>

1. What will each node’s table be after the simultaneous-and-parallel exchange process of part (a) is repeated a second time?

**(HINT:** you do not have to go through each exchange in detail; the only

information added by an exchange is additional *reachability* information.

New Information

A: <B,B,1> <D,D,1> <C,B,2> <E,D,2> <F,B,3> <F,D,3>

B: <A,A,1> <C,C,1> <D,A,2> <F,C,2> <E,A,3> <E,C,3>

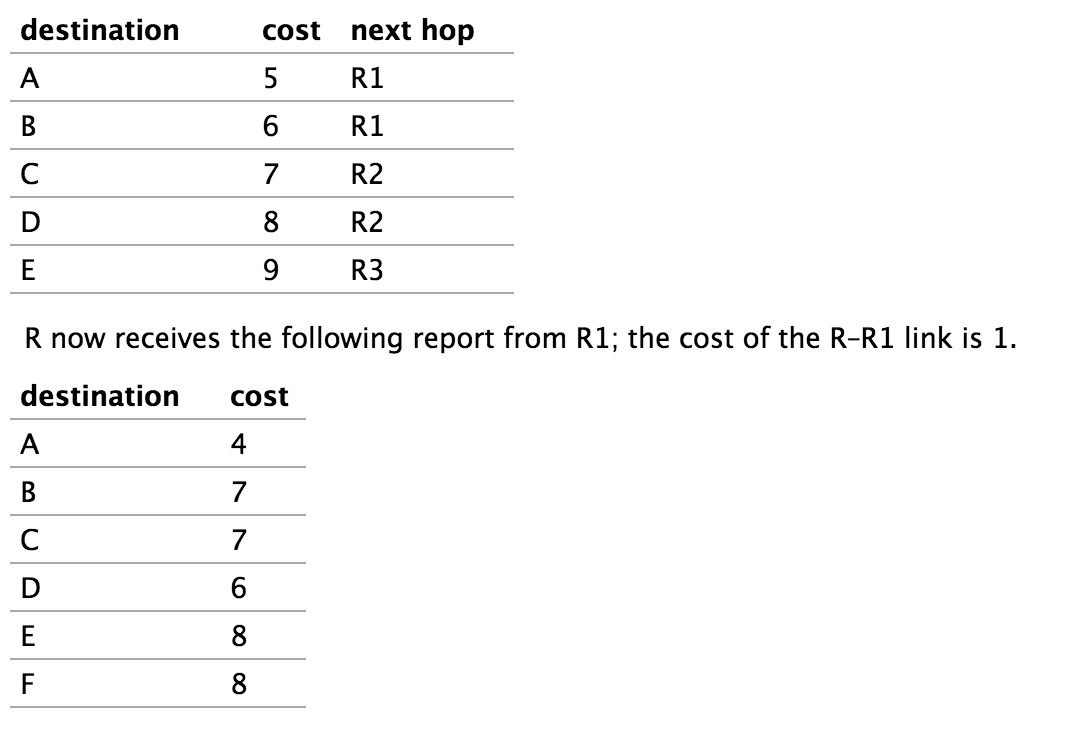
C: <B,B,1> <F,F,1> <A,B,2> <E,F,2> <D,B,3> <D,F,3>

D: <A,A,1> <E,E,1> <B,A,2> <F,E,2> <C,A,3> <C,E,3>

E: <D,D,1> <F,F,1> <A,D,2> <C,F,2> <B,D,3> <B,F,3>

F: <E,E,1> <C,C,1> <D,E,2> <B,C,2> <A,E,3> <A,C,3>

3. Suppose a router R has the following distance-vector table:



Give R’s updated table after it processes R1’s report. For each entry that changes, give a brief explanation, in the style of [*9.1.5   Example 4*](http://intronetworks.cs.luc.edu/1/html/routing.html#dv-example-4).

|  |  |  |  |
| --- | --- | --- | --- |
| Destination | Next hop | Cost | Reason |
| A | R1 | 5 | No Change; R probably sent this report before |
| B | R1 | 6 | Lower Cost Route via R1 |
| C | R2 | 8 | Next\_Hop Increase |
| D | R1 | 7 | Lower-Cost Route via R1 |
| E | R1 | 8 | No change; R’s cost via R1 is tied with R’s cost via R3 |
| F | R1 | 9 | New Destination |