COMP3721 – Assignment Four – Fall 2008

General Instructions

- You may work with one partner for this assignment. Your partner may be from your set or another full-time CST set.
- You and your partner may discuss any and all details of each question freely. You may also discuss questions in broad terms with others, particularly in lab, but ultimately your answers should show sufficient individuation from others' answers reflecting your work in answering the questions.
- All work submitted is subject to the standards of conduct as specified in BCIT Policy 5002.

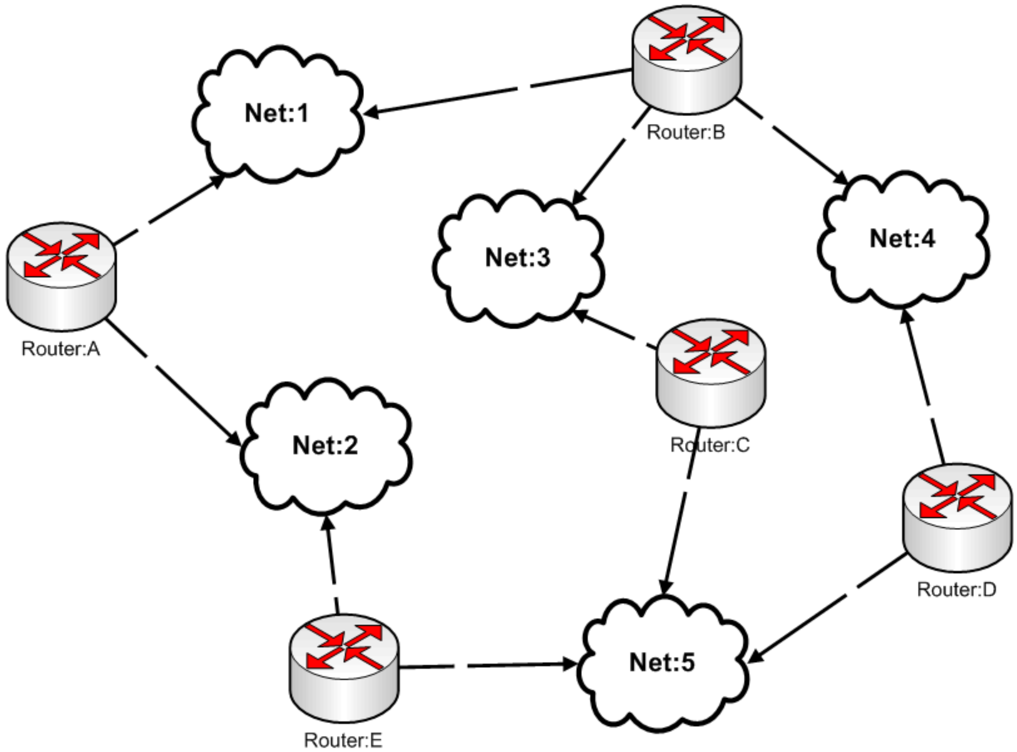
Submissions

- This assignment is due Friday, December 5, 2008 by 1600 hrs at the latest. Late assignments will not be accepted.
- Submit you assignment to your **<u>lab instructor's assignment box</u>** in the SW2 connector.
- Your submissions must include a cover page clearly specifying your name, student number and set. If working with a partner, this information should be provided for each partner.

Marking

The assignment consists of 6 questions totaling 35 marks.

1. Consider the network shown below:



Distance vector routing is used - measured delay is used rather than hop count. The measured delays from router C to routers B, D, and E are 3, 5, and 5, respectively. C receives the following three distance vectors:

To	B	E	D
Net:1	4	10	9
Net:2	15	7	10
Net:3	1	4	5
Net:4	5	7	5
Net:5	8	2	3

What is C's new routing table? Give both the outgoing network and router to use as well as the expected delay.

Adding the expected delay to B, E and D to the vector received from each router yields the following table with the least-cost path in bold:

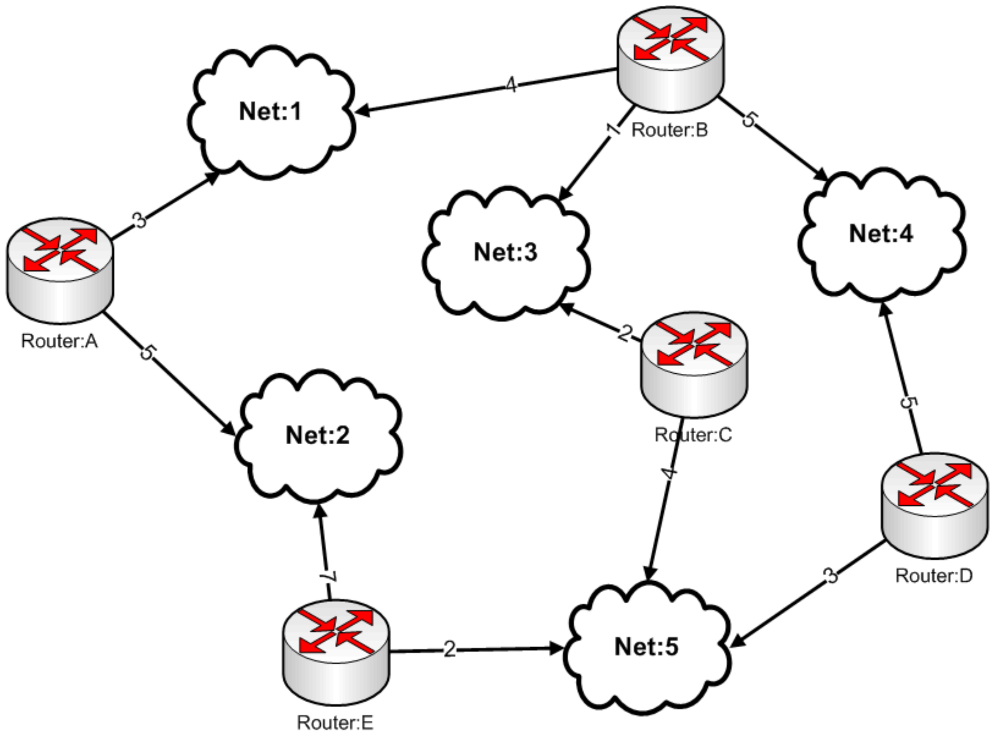
To	B'	E'	D'
Net:1	4 + 3 = 7	10 + 5 = 15	9 + 5 = 14
Net:2	15 + 3 = 18	7 + 5 = 12	10 + 5 = 15
Net:3	1 + 3 = 4	4 + 5 = 9	5 + 5 = 10
Net:4	5 + 3 = 8	7 + 5 = 12	5 + 5 = 10
Net:5	8 + 3 = 11	2 + 5 = 7	3 + 5 = 8

C's new routing table will consist of the least-cost path for each destination from the table above (excepting itself); either of the two equal cost paths to A may be used.

To	Cost	Path
Net:1	**7**	**B (Net:3)**
Net:2	**12**	**E (Net:5)**
Net:3	**3**	**-**
Net:4	**8**	**B (Net:3)**
Net:5	**5**	**-**

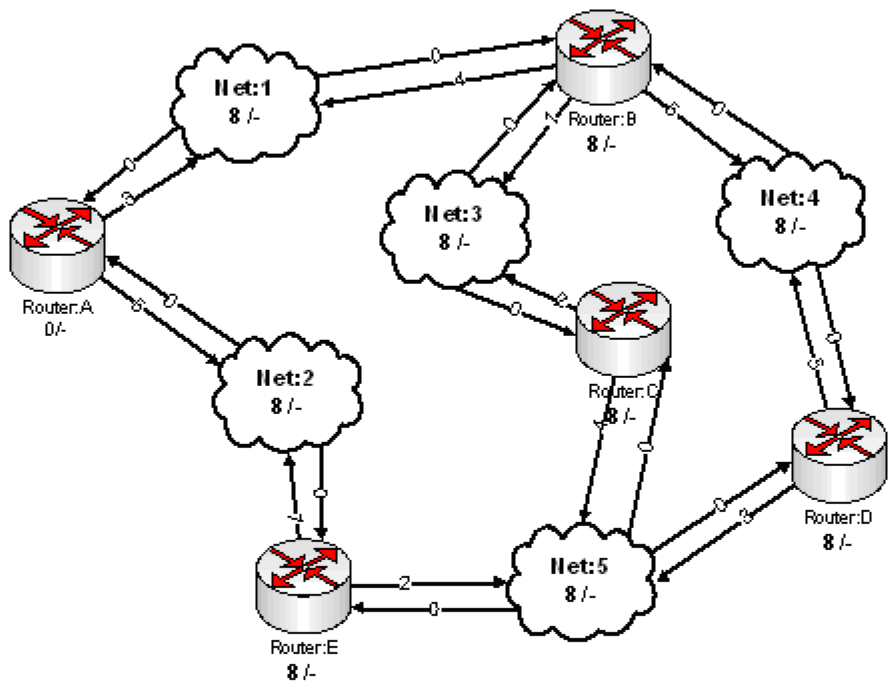
Note that this update is only one stage of a multi-stage process. Distance vector routing involves an iterative process of updates – only after several rounds of updates do all of the routers will converge to the best possible path.

2. Router A has accumulated the following view of the network through receipt of a series of link state packets:



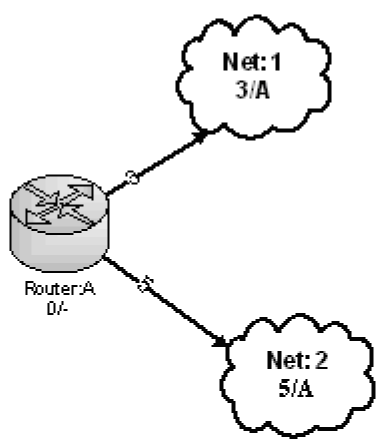
Apply Dijkstra's shortest-path algorithm to determine A's routing table. Your solution must show the intermediate steps in running Dijkstra's algorithm – the correct final routing table alone is not sufficient.

The first step is to augment the graph. Given that the routers are part of the networks they are connected to, there is no additional cost to 'get out' of a network to a router – thus the network should be supplemented with zero-weight edges to the connected routers. In addition, the Dijkstra algorithm requires each node (router or network in this case) be assigned an initial infinite distance; the exception is the source node (A in this case) which is known to be a distance of 0 from itself.

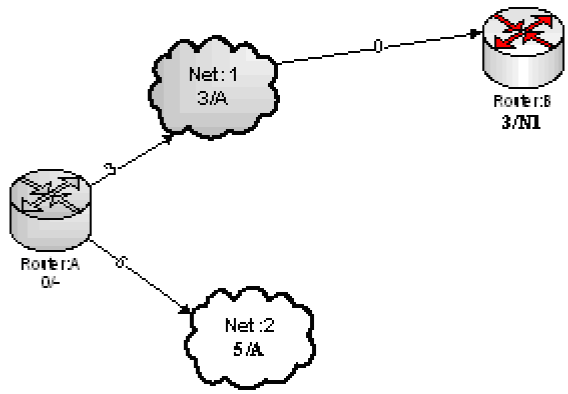


In each pass of the Dijkstra algorithm, one more node is added to the solution. For this network then, 10 passes are required for the five routers and five networks.

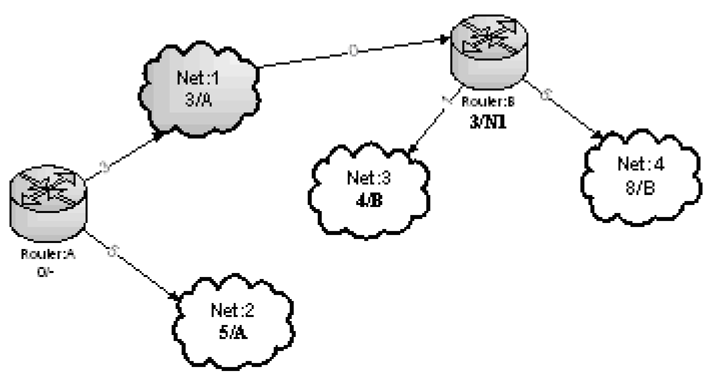
Pass	**Node**	**Distance/Preceding Node**
1	**A**	**0/-**



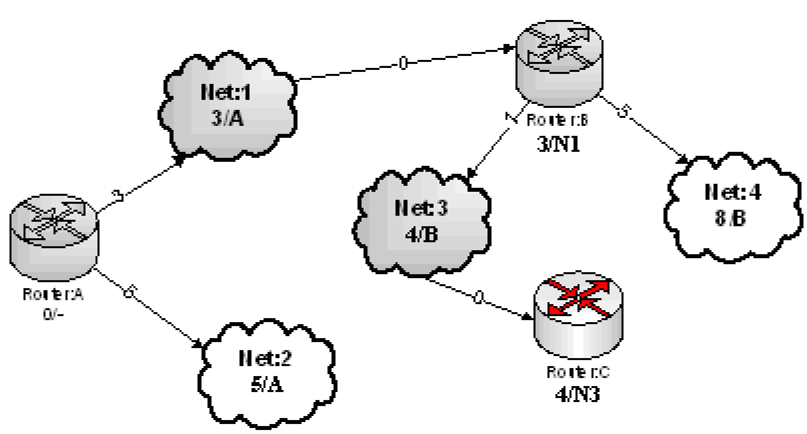
Pass	Node	Distance/Preceding Node
2	Net:1	3/A



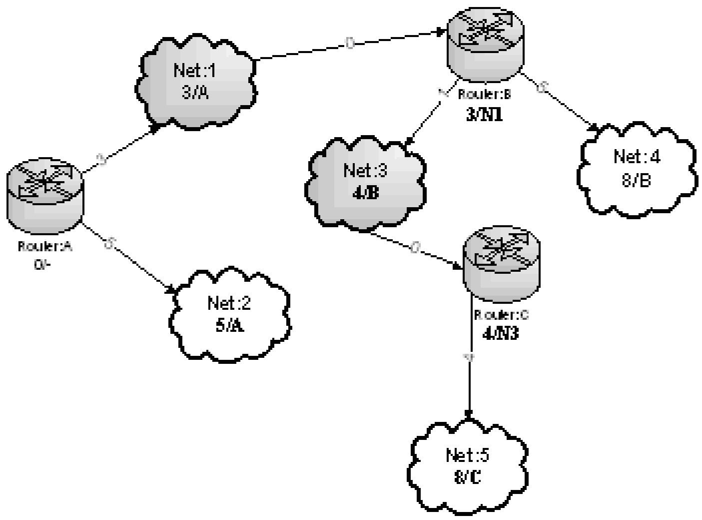
Pass	Node	Distance/Preceding Node
3	B	3/Net:1



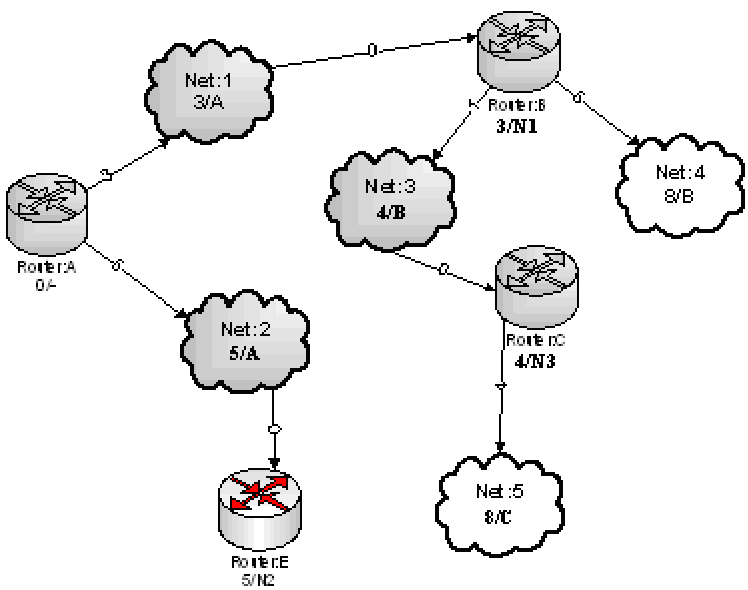
Pass	Node	Distance/Preceding Node
4	Net:3	4/B



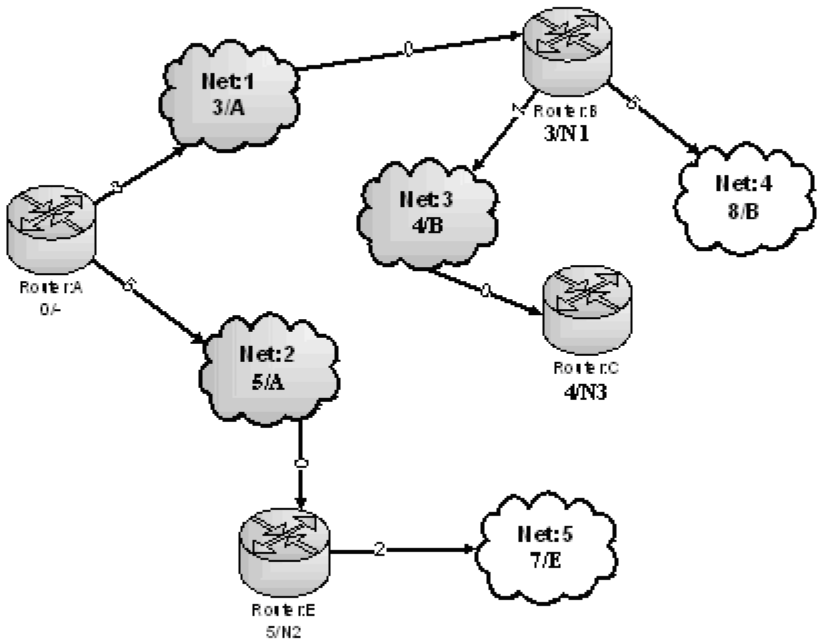
Pass	Node	Distance/Preceding Node
5	C	4/Net:3



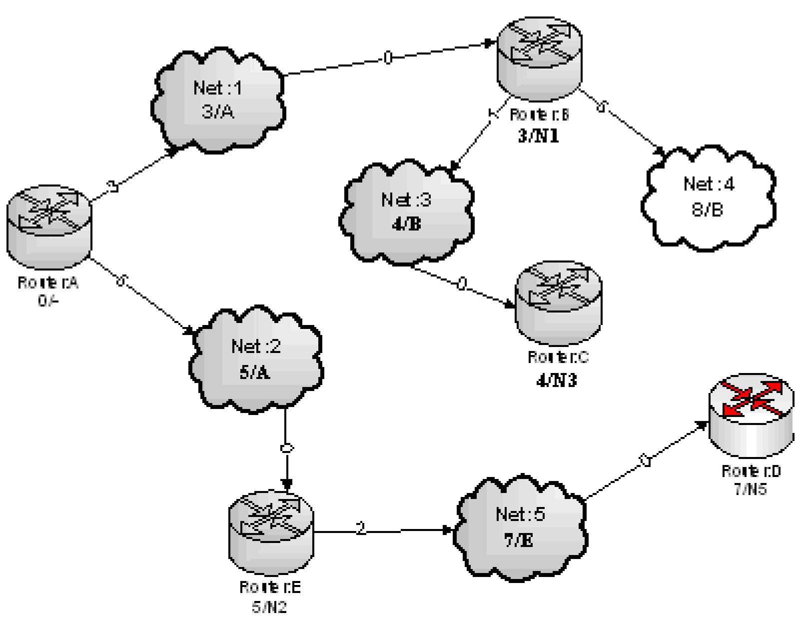
Pass	Node	Distance/Preceding Node
6	Net:2	5/A



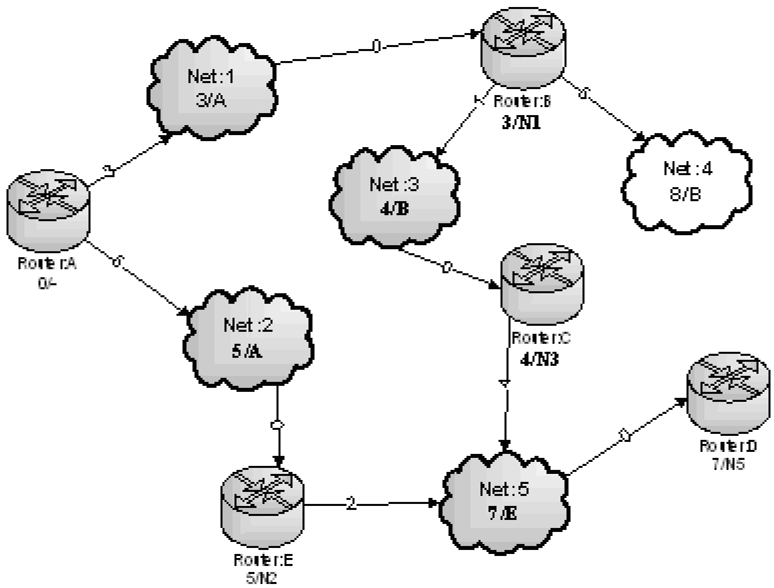
Pass	Node	Distance/Preceding Node
7	E	5/Net:2



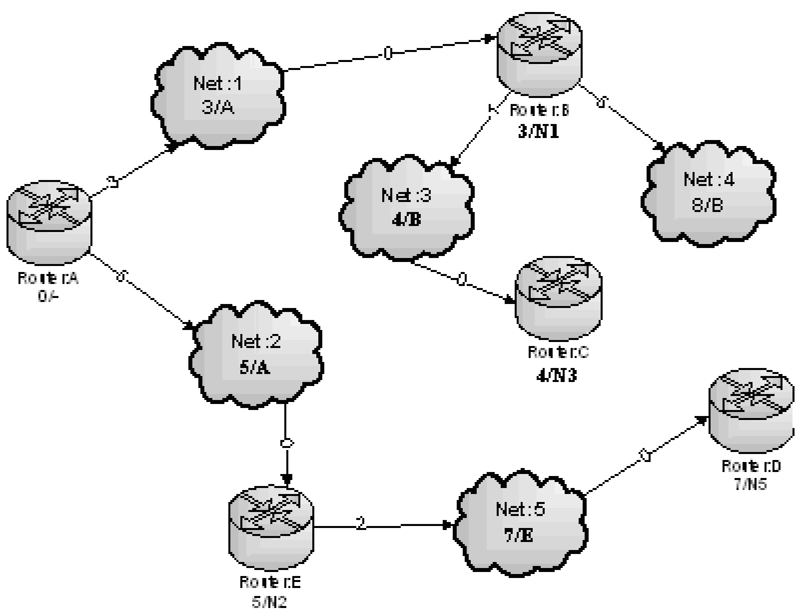
Pass	Node	Distance/Preceding Node
8	Net:5	7/E



Pass	Node	Distance/Preceding Node
9	D	7/Net:5



Pass	Node	Distance/Preceding Node
10	Net:4	8/B



We can then extract the routing table:

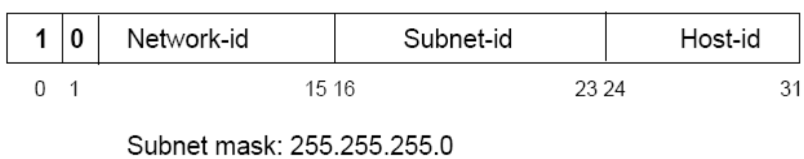
Destination	Cost	Next Hop
Net:1	3	-
Net:2	5	-
Net:3	4	B (Net:1)
Net:4	8	B (Net:1)
Net:5	7	E (Net:2)

3. A university has 150 LANs with 100 hosts in each LAN.

Solutions follow questions:

(a) Suppose the university has one Class B address. Design an appropriate subnet addressing scheme.

A Class B address has 14 bits for the network ID and 16 bits for the host ID. To design an appropriate subnet addressing scheme we need to decide how many bits to allocate to the host ID versus the subnet ID. We can choose either 7 bits or 8 bits to identify the hosts.
If we allocate 8 bits for to identify the host, as shown below, then there are sufficient subnet-id bits to cover up to 2^8=256 LANs and enough host-id bits to cover up to 256 hosts for each LAN. The subnet mask in this case is 255.255.255.0



Subnet mask: 255.255.255.0

If we allocate 7 bits for to identify the host, then there are sufficient subnet-id bits to cover up to 2^9=512 LANs and enough host-id bits to cover up to 128 hosts for each LAN. The subnet mask in this case is 255.255.255.128.

The choice between 7 or 8 bits to represent the hosts depends on which is likely to grow more, the number of subnets or the number of hosts in a LAN. Alternatively a variable-length prefix scheme using 7-bit host addresses, and grouping these form larger subnets provides greater flexibility in accommodating future changes.

(b) Design an appropriate CIDR addressing scheme.

CIDR addressing scheme involves devising a prefix length that indicates the length of the network mask. In this case, 8 bits are required to identify each LAN (since 127 < 150 < 255) and 7 bits are required to identify each host in each LAN (since 63 < 100 < 127). Therefore a CIDR address would use a 17-bit prefix, and thus have an address of the form address/17.

4. A router has the following entries in its routing table:

Address/Mask	Next Hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
Default	Router 2

For each of the following IP addresses, what does the router do if a packet with that address arrives?

a. 135.46.63.10
b. 135.46.57.14
c. 135.46.52.2
d. 192.53.40.7
e. 192.53.56.7

A: 135.46.56.0/22: ***10000111.00101110.001110****00.00000000*
B: 135.46.60.0/22: ***10000111.00101110.001111****00.00000000*
C: 192.53.40.0/23: ***11000000.00110101.0010100****0.00000000*

a. **135.46.63.10 →** ***10000111.00101110.001111****11.00001010*

Bold Matches with B → forward to Interface 1

b. **135.46.57.14 →** ***10000111.00101110.001110****01.00001110*

Bold Matches with A → forward to Interface 0

c. **135.46.52.2 →** *10000111.00101110.00110100.00000010*

No matching rule → forward to Router 2 (default)

d. **192.53.40.7 →** ***11000000.00110101.0010100****0.00000111*

Bold Matches with C → forward to Router 1

e. **192.53.56.7 →** *11000000.00110101.00111000.00000111*

No matching rule → forward to Router 2 (default)

5. Examine the following (complete) **raw fame**, which was captured using tcpdump:

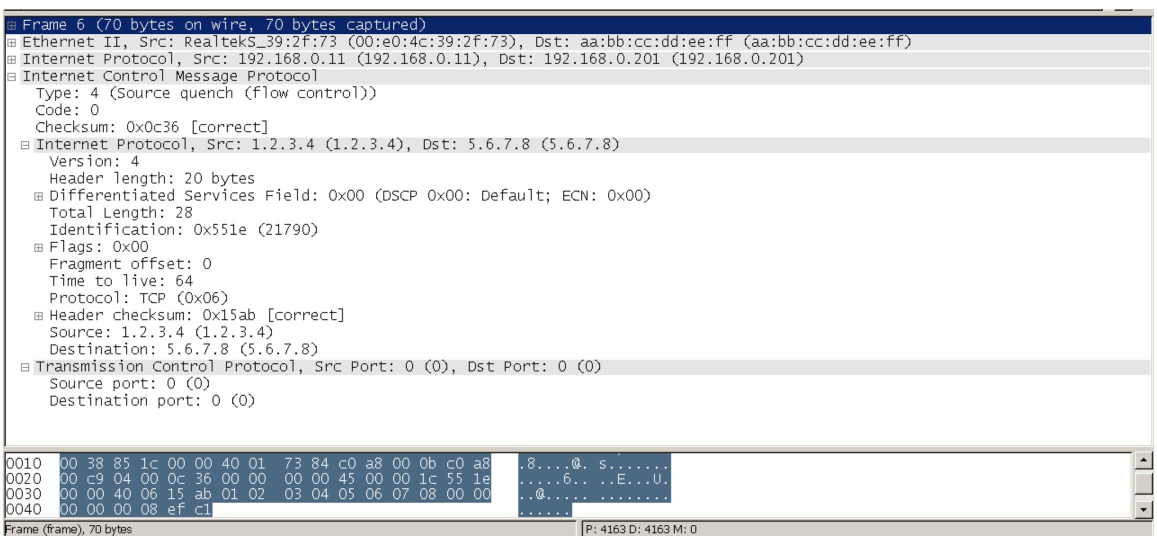
0000 aa bb cc dd ee ff 00 e0 4c 39 2f 73 08 00 45 00 ....L9/s..E.
0010 00 38 85 1c 00 00 40 01 73 84 c0 a8 00 0b c0 a8 .8....@.s.......
0020 00 c9 04 00 0c 36 00 00 00 00 45 00 00 1c 55 1e .....6....E...U.
0030 00 00 40 06 15 ab 01 02 03 04 05 06 07 08 00 00 ..@............
0040 00 00 00 08 ef c1

Analyze the frame in detail and provide the following information:

(a). All the protocol types carried in the frame

The following diagram shows all of the headers contained in this frame:

- Ethernet
- IP
- ICMP (together with the original IP and TCP headers)



(b). All of the available address information (IP addresses, ports, etc)

The following diagram shows the MAC and IP addresses associated with this frame. The original TCP header is embedded as part of the ICMP payload and shows the port numbers (see diagram in (a)).

```
⊞ Frame 6 (70 bytes on wire, 70 bytes captured)
⊟ Ethernet II, Src: RealtekS_39:2f:73 (00:e0:4c:39:2f:73), Dst: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff)
  ⊞ Destination: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff)
  ⊞ Source: RealtekS_39:2f:73 (00:e0:4c:39:2f:73)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.0.11 (192.168.0.11), Dst: 192.168.0.201 (192.168.0.201)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 56
    Identification: 0x851c (34076)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x7384 [correct]
    Source: 192.168.0.11 (192.168.0.11)
    Destination: 192.168.0.201 (192.168.0.201)
⊞ Internet Control Message Protocol

0010  00 38 85 1c 00 00 40 01  73 84 c0 a8 00 0b c0 a8   .8....@. s.......
0020  00 c9 04 00 0c 36 00 00  00 00 45 00 00 1c 55 1e   .....6.. ..E...U.
0030  00 00 40 06 15 ab 01 02  03 04 05 06 07 08 00 00   ..@..... ........
0040  00 00 00 08 ef c1                                  ......
```

(c). The **type** and **function** of this datagram.

> This is an ICMP packet; the type is 0x04, which is a source quench. The function of a source quench is to implement flow control. See the diagram in part (a).

(d). All the key control field settings such as: Flag bits, TTL, TOS, Fragmentation.

> These are all shown in the preceding two diagrams.

(e). Header lengths and the total datagram size.

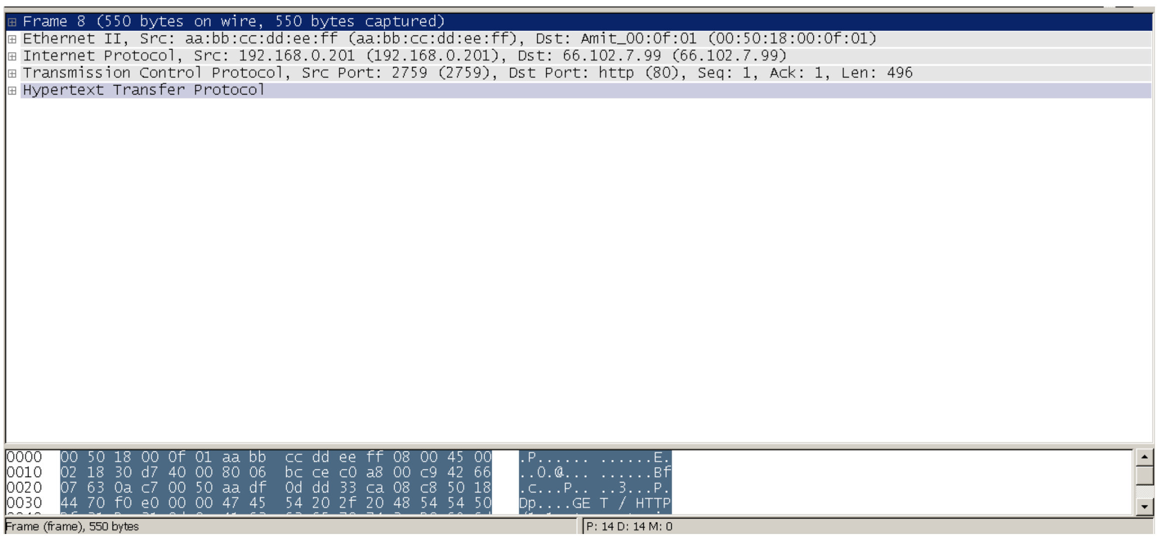
```
⊞ Frame 6 (70 bytes on wire, 70 bytes captured)
⊟ Ethernet II, Src: RealtekS_39:2f:73 (00:e0:4c:39:2f:73), Dst: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff)
  ⊞ Destination: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff)
  ⊞ Source: RealtekS_39:2f:73 (00:e0:4c:39:2f:73)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.0.11 (192.168.0.11), Dst: 192.168.0.201 (192.168.0.201)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 56
    Identification: 0x851c (34076)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x7384 [correct]
    Source: 192.168.0.11 (192.168.0.11)
    Destination: 192.168.0.201 (192.168.0.201)
⊟ Internet Control Message Protocol
    Type: 4 (Source quench (flow control))
    Code: 0
    Checksum: 0x0c36 [correct]
  ⊞ Internet Protocol, Src: 1.2.3.4 (1.2.3.4), Dst: 5.6.7.8 (5.6.7.8)
  ⊞ Transmission Control Protocol, Src Port: 0 (0), Dst Port: 0 (0)
```

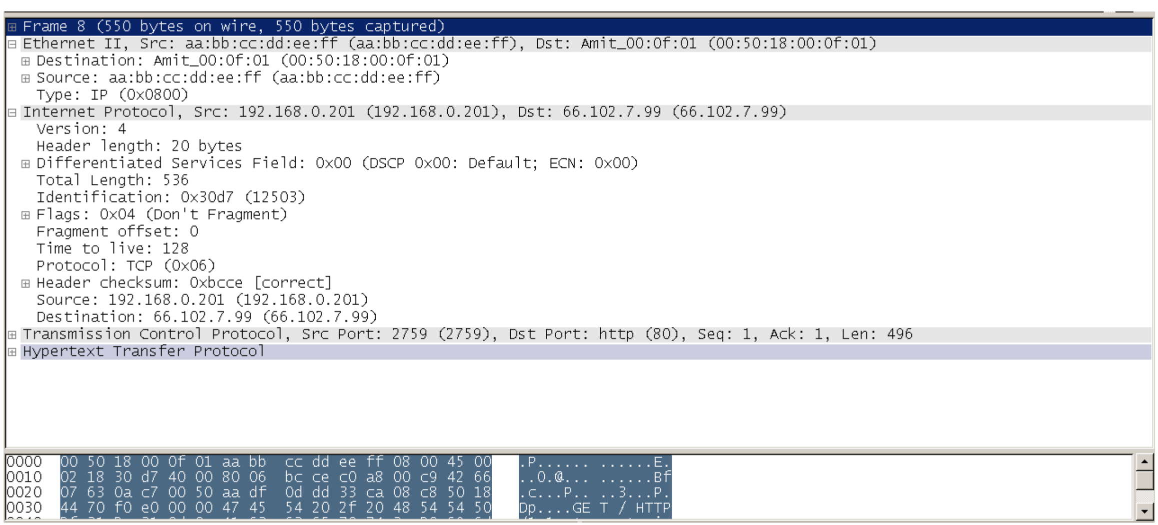
6. Examine the following (complete) **raw fame**, which was captured using tcpdump. Provide an analysis similar to that in question #8 above.

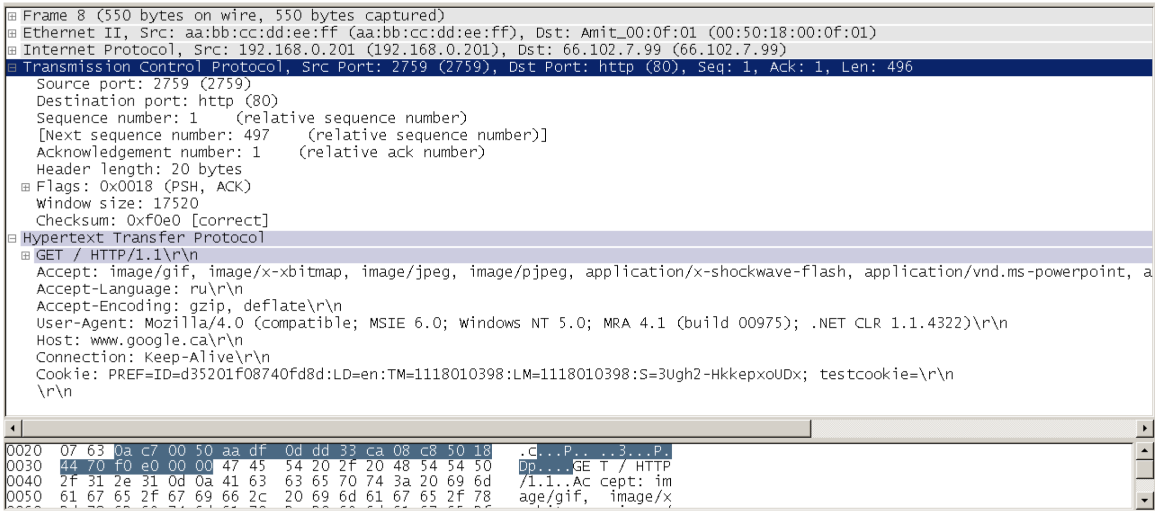
```
0000   00 50 18 00 0f 01 aa bb cc dd ee ff 08 00 45 00       .P...........E.
0010   02 18 30 d7 40 00 80 06 bc ce c0 a8 00 c9 42 66       ..0.@.........Bf
0020   07 63 0a c7 00 50 aa df 0d dd 33 ca 08 c8 50 18       .c...P....3...P.
0030   44 70 f0 e0 00 00 47 45 54 20 2f 20 48 54 54 50       Dp....GET / HTTP
0040   2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d        /1.1..Accept: im
0050   61 67 65 2f 67 69 66 2c 20 69 6d 61 67 65 2f 78       age/gif, image/x
0060   2d 78 62 69 74 6d 61 70 2c 20 69 6d 61 67 65 2f       -xbitmap, image/
0070   6a 70 65 67 2c 20 69 6d 61 67 65 2f 70 6a 70 65       jpeg, image/pjpe
0080   67 2c 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78       g, application/x
0090   2d 73 68 6f 63 6b 77 61 76 65 2d 66 6c 61 73 68       -shockwave-flash
00a0   2c 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 76 6e       , application/vn
00b0   64 2e 6d 73 2d 70 6f 77 65 72 70 6f 69 6e 74 2c       d.ms-powerpoint,
00c0   20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 76 6e 64       application/vnd
00d0   2e 6d 73 2d 65 78 63 65 6c 2c 20 61 70 70 6c 69       .ms-excel, appli
00e0   63 61 74 69 6f 6e 2f 6d 73 77 6f 72 64 2c 20 2a       cation/msword, *
00f0   2f 2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75       /*..Accept-Langu
0100   61 67 65 3a 20 72 75 0d 0a 41 63 63 65 70 74 2d       age: ru..Accept-
0110   45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20       Encoding: gzip,
0120   64 65 66 6c 61 74 65 0d 0a 55 73 65 72 2d 41 67       deflate..User-Ag
0130   65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 34 2e 30       ent: Mozilla/4.0
0140   20 28 63 6f 6d 70 61 74 69 62 6c 65 3b 20 4d 53       (compatible; MS
0150   49 45 20 36 2e 30 3b 20 57 69 6e 64 6f 77 73 20       IE 6.0; Windows
0160   4e 54 20 35 2e 30 3b 20 4d 52 41 20 34 2e 31 20       NT 5.0; MRA 4.1
0170   28 62 75 69 6c 64 20 30 30 39 37 35 29 3b 20 2e       (build 00975); .
0180   4e 45 54 20 43 4c 52 20 31 2e 31 2e 34 33 32 32       NET CLR 1.1.4322
0190   29 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 67 6f 6f  )    ..Host: www.goo
01a0   67 6c 65 2e 63 61 0d 0a 43 6f 6e 6e 65 63 74 69       gle.ca..Connecti
01b0   6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 0d 0a       on: Keep-Alive..
01c0   43 6f 6f 6b 69 65 3a 20 50 52 45 46 3d 49 44 3d       Cookie: PREF=ID=
01d0   64 33 35 32 30 31 66 30 38 37 34 30 66 64 38 64       d35201f08740fd8d
01e0   3a 4c 44 3d 65 6e 3a 54 4d 3d 31 31 31 38 30 31       :LD=en:TM=111801
01f0   30 33 39 38 3a 4c 4d 3d 31 31 31 38 30 31 30 33       0398:LM=11180103
0200   39 38 3a 53 3d 33 55 67 68 32 2d 48 6b 6b 65 70       98:S=3Ugh2-Hkkep
0210   78 6f 55 44 78 3b 20 74 65 73 74 63 6f 6f 6b 69       xoUDx; testcooki
0220   65 3d 0d 0a 0d 0a                                               e=....
```

This is simply a TCP packet that performs an HTTP GET request from a client to a web server. The string **GET / HTTP/1.1** in the payload **portion** indicates this. This is a request for the root page from http://www.google.ca. The PSH bit is set in this packet – this indicates to receiving stack that the data should be pushed up to the application layer.

The following diagrams show all the relevant protocol details:






Frame 8 (550 bytes on wire, 550 bytes captured)
Ethernet II, Src: aa:bb:cc:dd:ee:ff (aa:bb:cc:dd:ee:ff), Dst: Amit_00:0f:01 (00:50:18:00:0f:01)
Internet Protocol, Src: 192.168.0.201 (192.168.0.201), Dst: 66.102.7.99 (66.102.7.99)
Transmission Control Protocol, Src Port: 2759 (2759), Dst Port: http (80), Seq: 1, Ack: 1, Len: 496
Source port: 2759 (2759)
Destination port: http (80)
Sequence number: 1 (relative sequence number)
[Next sequence number: 497 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (PSH, ACK)
Window size: 17520
Checksum: 0xf0e0 [correct]
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-powerpoint, a
Accept-Language: ru\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; MRA 4.1 (build 00975); .NET CLR 1.1.4322)\r\n
Host: www.google.ca\r\n
Connection: Keep-Alive\r\n
Cookie: PREF=ID=d35201f08740fd8d:LD=en:TM=1118010398:LM=1118010398:S=3Ugh2-HkkepxoUDx; testcookie=\r\n
\r\n
0020 07 63 0a c7 00 50 aa df 0d dd 33 ca 08 c8 50 18 .c...P.. ..3...P.
0030 44 70 f0 e0 00 00 47 45 54 20 2f 20 48 54 54 50 Dp....GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
0050 61 67 65 2f 67 69 66 2c 20 69 6d 61 67 65 2f 78 age/gif, image/x