Computer Networks 159.334

Answers for Assignment No. 1 – Semester 2, 2010

Professor Richard Harris

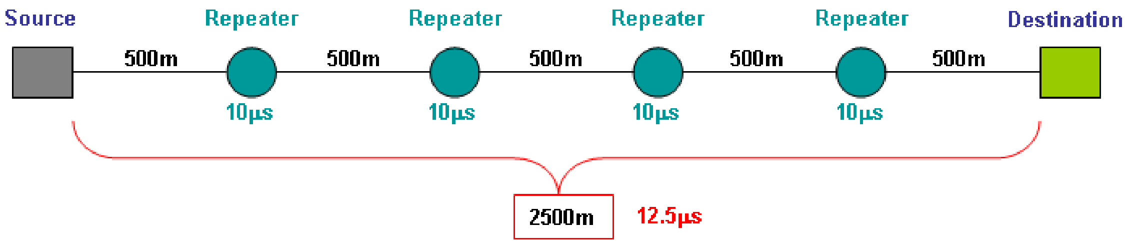
Problems

Question 1

1.1 Suppose the length of a 10Base5 cable is 2500 metres. If the speed of propagation in a thick coaxial cable is 200,000,000 m/s how long does it take for a bit to travel from the beginning to the end of the network? Assume that there is a 10μs delay in the equipment.

Answer:

10Base5 has a maximum length of 500 metres so repeaters should be inserted into the cable in order to ensure transmission is possible over the full length of the 2500m cable. 4 repeaters are required as shown below:



Recommended answer is **4 x 10μs + 2500/200,000,000 = 52.5μs**, but other answers will be accepted with suitable arguments.

1.2 The data rate of 10Base5 is 10Mbps. How long does it take to create the smallest frame? Show your calculations.

Answer:

The smallest frame is 64 bytes or 512 bits. With a data rate of 10 Mbps, we have

$$T_{fr} = (512 \text{ bits}) / (10 \text{ Mbps}) = \mathbf{51.2\ \mu s}$$

This means that the time required to send the smallest frame is the same at the maximum time required to detect the collision.

Question 2

Asynchronous Transfer Mode Packets have an 8 bit CRC for the information contained in their headers. The header has the following 6 fields:

- First 4 bits: Generic Flow Control (GFC) field
- Next 8 bits: Virtual Path Identifier (VPI) field
- Next 16 bits: Virtual Circuit Identifier (VCI)field
- Next 3 bits: Type field
- Next 1 bit: CLP field
- Next 8 bits: CRC field

2.1 The CRC is calculated using the generator polynomial $x^8 + x^2 + x + 1$. Find the CRC bits if the GFC, VPI, Type and CLP fields are all zero and the VCI field is given by 00000000 00001111. Assume that the GFC bits correspond to the highest order bits in the polynomial.

Answer:

GFC	VPI	VCI	TYPE	CLP
000	**00000000**	**00000000 00001111**	**000**	**0**

Polynomial for this case is thus: $\mathbf{x^7+x^6+x^5+x^4}$

We multiply by x^8 to shift the polynomial for getting the new polynomial for division by the generator polynomial. This gives: $\mathbf{x^{15}+x^{14}+x^{13}+x^{12}}$

Dividing by the generator polynomial we obtain the CRC bits as: **11011110**

$$\begin{array}{r}
x^7+x^6+x^5+x^4+x^1 \\
x^8+x^2+x^1+1\,\overline{)\,x^{15}+x^{14}+x^{13}+x^{12}} \\
\underline{x^{15}+x^9+x^8+x^7} \\
x^{14}+x^{13}+x^{12}+x^9+x^8+x^7 \\
\underline{x^{14}+x^8+x^7+x^6} \\
x^{13}+x^{12}+x^9+x^6 \\
\underline{x^{13}+x^7+x^6+x^5} \\
x^{12}+x^9+x^7+x^5 \\
\underline{x^{12}+x^6+x^5+x^4} \\
x^9+x^7+x^6+x^4 \\
\underline{x^9+x^3+x^2+x^1} \\
x^7+x^6+x^4+x^3+x^2+x^1
\end{array}$$

2.2 Read Forouzan (Fourth Edition) Chapter 10.4 on Cyclic codes and then explain whether the code described in question **2.1** above can detect single bit errors.

Answer:

Yes, the generator polynomial $x^8 + x^2 + x + 1$ can detect single bit errors since there is more than one term in this polynomial and the coefficient of x^0 is '1'. See page 294 of the Forouzan textbook.

Question 3

3.1 Given the dataword 1010011110 and the divisor 10111

a) Show the generation of the codeword at the sender site using binary division

b) Show the checking of the codeword at the receiver site assuming no error has occurred.

c) What is the syndrome at the receiver end if the dataword has an error in the 5th bit position counting from the right? Namely: dataword 1010001110 is received.

Answer:

a) Binary division case

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M=	1	0	1	0	0	1	1	1	1	0				
G=														
10111	1	0	1	0	0	1	1	1	1	0	0	0	0	0
	1	0	1	1	1									
		0	0	1	1	1								
		0	0	0	0	0								
			0	1	1	1	1							
			0	0	0	0	0							
				1	1	1	1	1						
				1	0	1	1	1						
					1	0	0	0	1					
					1	0	1	1	1					
						0	1	1	0	0				
						0	0	0	0	0				
							1	1	0	0	0			
							1	0	1	1	1			
								1	1	1	1	0		
								1	0	1	1	1		
									1	0	0	1	0	
									1	0	1	1	1	
										0	1	0	1	0
											0	0	0	0
											1	0	1	0
T=	1	0	1	0	0	1	1	1	1	0	1	0	1	0
			Original Message								CRC Checksum			

CRC Checksum was **1010**

Codeword was **10100111101010**

b) Receiver using binary division:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M=	1	0	1	0	0	1	1	1	1	0				
G=														
10111	1	0	1	0	0	1	1	1	1	0	1	0	1	0
	1	0	1	1	1									
		0	0	1	1	1								
		0	0	0	0	0								
			0	1	1	1	1							
			0	0	0	0	0							
				1	1	1	1	1						
				1	0	1	1	1						
					1	0	0	0	1					
					1	0	1	1	1					
						0	1	1	0	0				
						0	0	0	0	0				
							1	1	0	0	1			
							1	0	1	1	1			
								1	1	1	0	0		
								1	0	1	1	1		
									1	0	1	1	1	
									1	0	1	1	1	
										0	0	0	0	0
											0	0	0	0
											0	0	0	0

Remainder was **0000** as required.

c) Suppose that we received the corrupted dataword with the old CRC value as follows: **10100011101010.** We can determine the syndrome in this case as:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M=	1	0	1	0	0	1	1	1	1	0				
G=														
10111	1	0	1	0	0	0	1	1	1	0	1	0	1	0
	1	0	1	1	1									
		0	0	1	1	0								
		0	0	0	0	0								
			0	1	1	0	1							
			0	0	0	0	0							
				1	1	0	1	1						
				1	0	1	1	1						
					1	1	0	0	1					
					1	0	1	1	1					
						1	1	1	0	0				
						1	0	1	1	1				
							1	0	1	1	1			
							1	0	1	1	1			
								0	0	0	0	0		
								0	0	0	0	0		
									0	0	0	0	1	
									0	0	0	0	0	
										0	0	0	1	0
										0	0	0	0	0
											0	0	1	0

Non-zero CRC is obtained. **CRC = 0010**

3.2 Repeat Question **3.1** by using polynomials instead of binary arithmetic.

Answer:

The polynomial for the divisor is: $x^4 + x^2 + x + 1$

The polynomial for the dataword is: $x^9 + x^7 + x^4 + x^3 + x^2 + x^1$

a) In order to perform the operations required when using the polynomial form we must first multiply the dataword by x^4 (equivalent to adding the 4 zeros in the binary division method. This yields the new polynomial as: $x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5$ we now proceed to divide this by the divisor obtained above:

$$x^4 + x^2 + x + 1 \,\overline{)\, x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5}$$

Following the standard procedure, we obtain the remainder as $x^3 + x^1$, so that the bit pattern for this will be **1010** which agrees with our previous binary division result.

$$
\begin{array}{r}
x^9 + x^6 + x^5 + x^3 + x^2 + x^1 \\
x^4 + x^2 + x^1 + 1 \,\overline{)\, x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5}
\end{array}
$$

$$\underline{x^{13} + x^{11} + x^{10} + x^9}$$
$$x^{10} + x^9 + x^8 + x^7 + x^6 + x^5$$
$$\underline{x^{10} + x^8 + x^7 + x^6}$$
$$x^9 + x^5$$
$$\underline{x^9 + x^7 + x^6 + x^5}$$
$$x^7 + x^6$$
$$\underline{x^7 + x^5 + x^4 + x^3}$$
$$x^6 + x^5 + x^4 + x^3$$
$$\underline{x^6 + x^4 + x^3 + x^2}$$
$$x^5 + x^2$$
$$\underline{x^5 + x^3 + x^2 + x^1}$$
$$x^3 + x^1$$

b) When we repeat the process for the complete message with CRC attached, the required polynomial will be **$x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5 + x^3 + x^1$**

Thus we can then perform the required division once again and we find a **zero** remainder which verifies the required result.

$$
\begin{array}{r}
x^9 + x^6 + x^5 + x^3 + x^2 + x^1 \\
x^4 + x^2 + x^1 + 1 \,\overline{)\, x^{13} + x^{11} + x^8 + x^7 + x^6 + x^5 + x^3 + x^1}
\end{array}
$$

$$\underline{x^{13} + x^{11} + x^{10} + x^9}$$
$$x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^3 + x^1$$
$$\underline{x^{10} + x^8 + x^7 + x^6}$$
$$x^9 + x^5 + x^3 + x^1$$
$$\underline{x^9 + x^7 + x^6 + x^5}$$
$$x^7 + x^6 + x^3 + x^1$$
$$\underline{x^7 + x^5 + x^4 + x^3}$$
$$x^6 + x^5 + x^4 + x^1$$
$$\underline{x^6 + x^4 + x^3 + x^2}$$
$$x^5 + x^3 + x^2 + x^1$$
$$\underline{x^5 + x^3 + x^2 + x^1}$$
$$0$$

c) Taking the dataword that has an error in the appropriate 5th position, we obtain the new polynomial to be considered as: **$x^{13} + x^{11} + x^7 + x^6 + x^5 + x^3 + x^1$**

Thus we can then perform the required division once again and we find a non-zero remainder of **x^1** which verifies the required result and matches our conclusion from question 3.1 which found the bit pattern **0010.**

$$
\begin{array}{r}
x^9 + x^6 + x^5 + x^4 + x^3 \\
x^4 + x^2 + x^1 + 1 \overline{) x^{13} + x^{11} + x^7 + x^6 + x^5 + x^3 + x^1}
\end{array}
$$

$$\underline{x^{13} + x^{11} + x^{10} + x^9}$$

$$x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 + x^1$$

$$\underline{x^{10} + x^8 + x^7 + x^6}$$

$$x^9 + x^8 + x^5 + x^3 + x^1$$

$$\underline{x^9 + x^7 + x^6 + x^5}$$

$$x^8 + x^7 + x^6 + x^3 + x^1$$

$$\underline{x^8 + x^6 + x^5 + x^4}$$

$$x^7 + x^5 + x^4 + x^3 + x^1$$

$$\underline{x^7 + x^5 + x^4 + x^3}$$

$$0 + 0 + x^1 + 0$$

Question 4

Consider the following segment of the Internet that consists of 7 nodes and 11 links and answer the questions below:

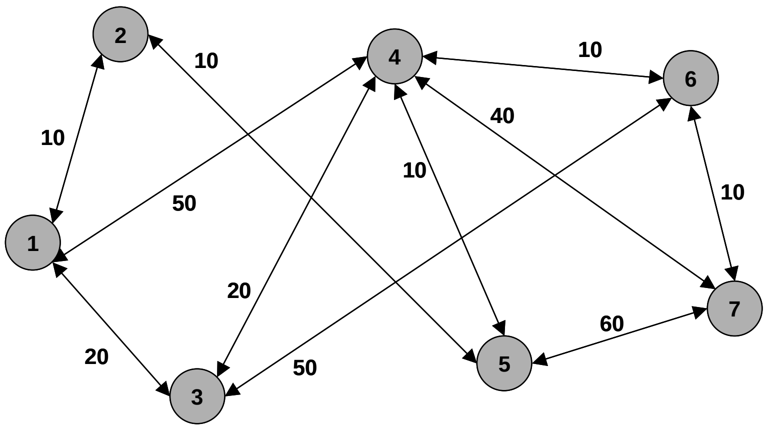


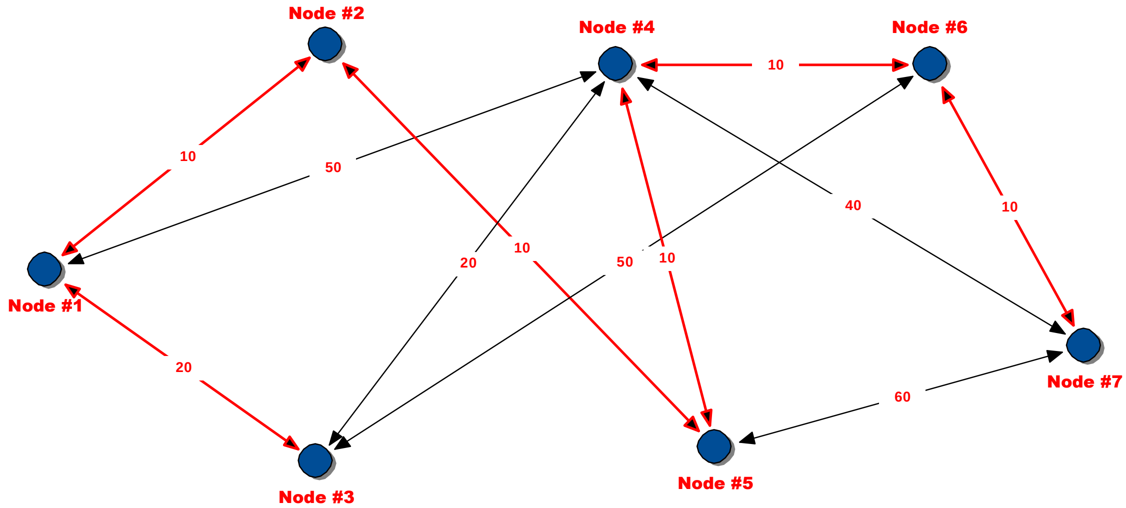
Figure 1: Sample Network from the Internet

The figures on the links represent the delay (in milliseconds) for traffic using that link.

4.1 Determine the shortest route tree based on the home node "1", and connecting to all other nodes, using the Dijkstra algorithm. *Carefully* draw <u>one</u> copy of this network into your answer book. Show your working by placing appropriate labels on the nodes of this network. ***(Do not draw multiple copies of the network for your answer – one will be sufficient.)***

Answer:

A tree is required for this answer.



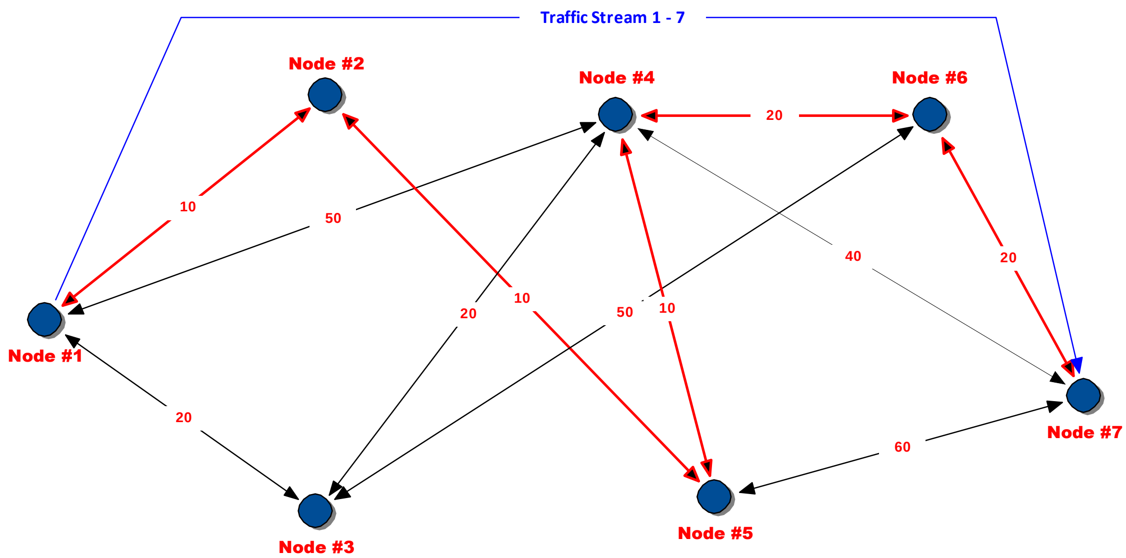
Note that for the shortest path between nodes 1 and 7, the cost is 50ms. The tree shows the path as traversing the links 1 – 2, 2 – 5, 5 – 4, 4 – 6, and 6 – 7.

4.2 Draw another copy of the network into your answer book. How does the shortest route change between nodes 1 and 7 if the delays on links 4 – 6 and 6 – 7 are both increased from 10ms to 20ms?

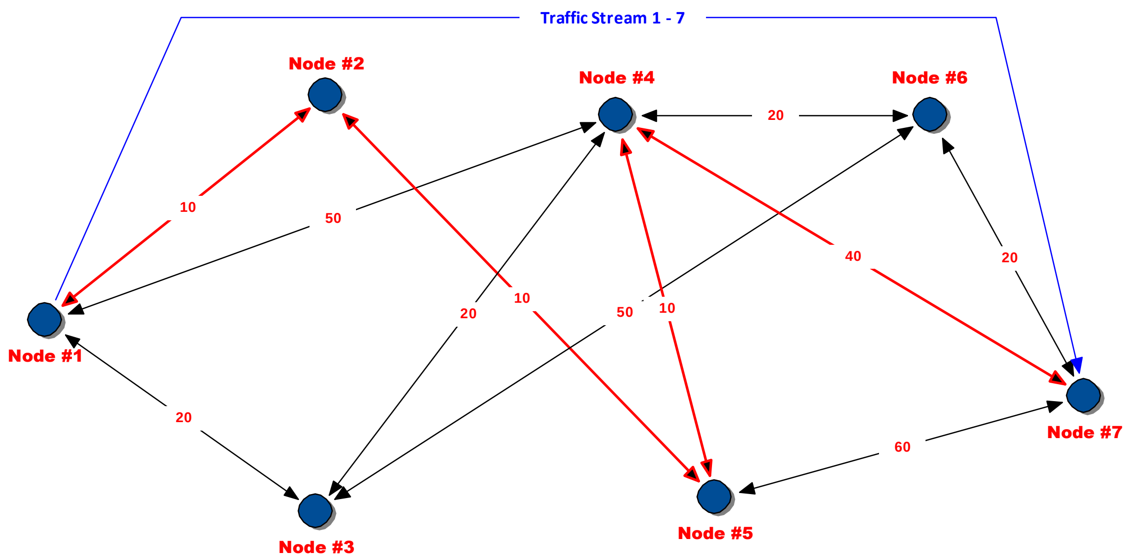
Answer:

Two answers are possible here:

1) The path is actually still the same, it just has a higher cost: Cost = 70ms.

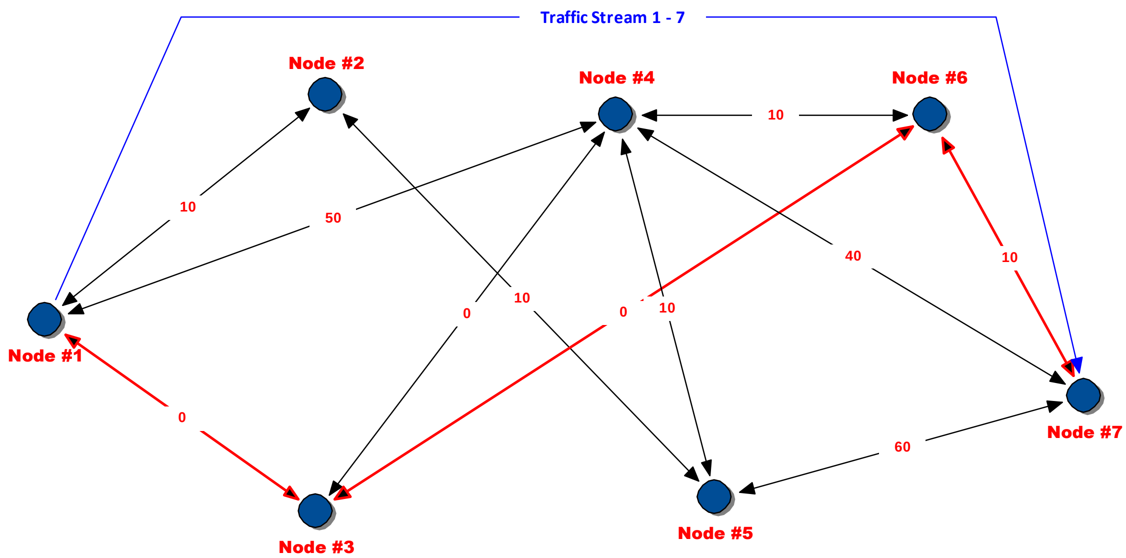


2) The path changes to using the 4 – 7 link instead of the 4 – 6 and 6 – 7 pair. The cost is 70ms.



4.3 If all the delays into and out of node 3 are misreported as zero, what effect will this misinformation have on traffic using the network? ***(Do not compute the new shortest route on a new diagram – only a descriptive answer is required.)***

Answer:



A totally new path is created if these links are misreported as having zero costs (delays). The new path would be via node #3 since it now acts as a sponge for traffic in the network. We can expect congestion to occur at this node as more traffic will divert towards it from elsewhere in the network.

4.4 Suppose that we need to determine a backup route between node 1 and node 7 in the network, and that it has been suggested that this route could be the *second shortest route* between these two nodes.

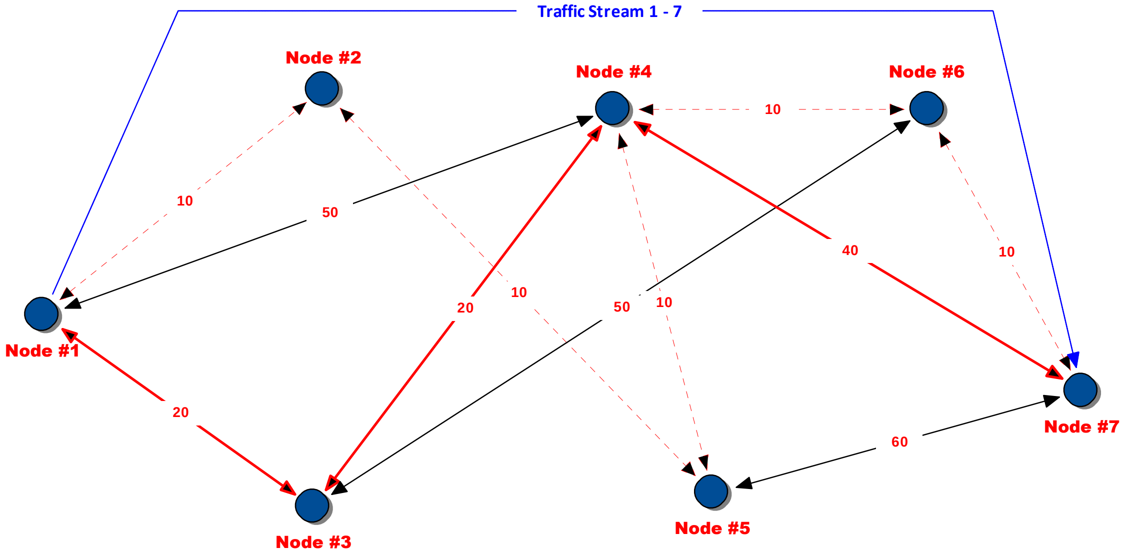
a) Suggest a way that could be used to find this second shortest route and use your method to locate this route for the network given in Figure 1 above.
b) What is the delay on this second shortest route?
c) Are there any links that are common to the shortest and second shortest route from node 1 to node 7? (If so, which links?)
d) How many links occur in **both** the shortest and second shortest routes?

Answers:

a) Many different answers are possible. Two answers that are quite logical and feasible are:

- Remove all links from the original shortest path or set their costs to infinity. Recalculate the shortest path. This actually achieves the second shortest path that is link disjoint from the original path.
- Systematically remove one link at a time from the original shortest path leaving the others available. Compute the shortest path again and then restore the link and move onto the next one. After you have computed all of the shortest paths with just one link missing, compare them and take the shortest available path. This can approach the required ideal. More sophisticated approaches can be used to get the true second shortest path, eg. Chen's algorithm.

b) Taking the first approach gives the ***link disjoint*** second shortest path as shown by the highlighted unbroken lines on the figure below:



Minimum cost of disjoint path: Node #1 to Node #7 is = 80ms.

Path:

Node #1 → (Lk_Node #1_Node #3) → Node #3 → (Lk_Node #3_Node #4) → Node #4 → (Lk_Node #4_Node #7) → Node #7

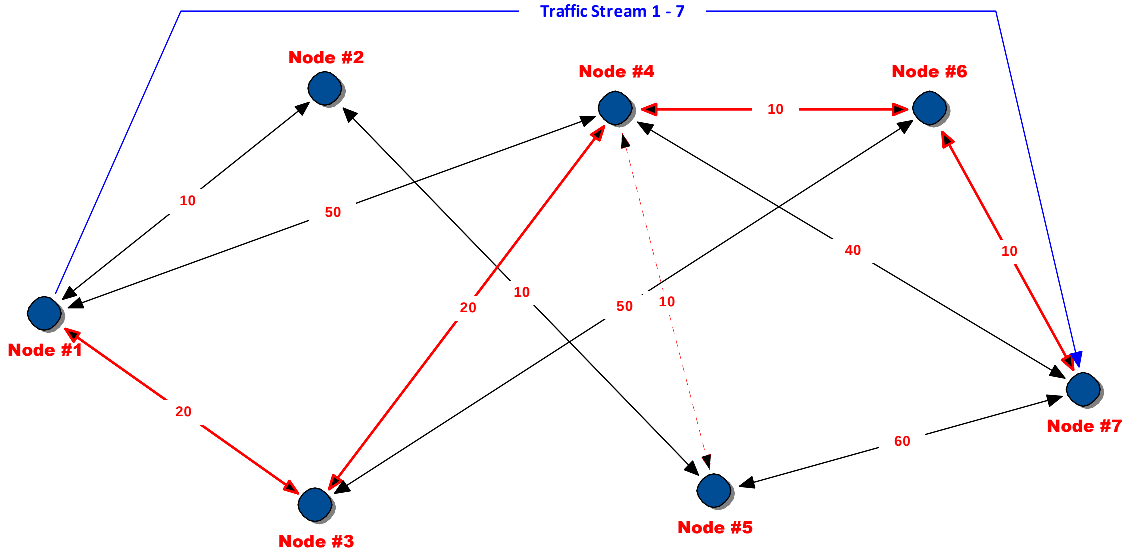
Taking the second approach produces the following set of possible paths:
Link Node #1 to Node #2 absent, cost = 60ms
Link Node #2 to Node #5 absent, cost = 60ms
Link Node #5 to Node #4 absent, cost = 60ms
Link Node #4 to Node #6 absent, cost = 70ms
Link Node #6 to Node #7 absent, cost = 70ms

Thus there are three candidate situations leading to a cost of 60ms that would qualify as being the second shortest path without considering disjointness. The path found for each of these three cases is:

Node #1 → (Lk_Node #1_Node #3) → Node #3 → (Lk_Node #3_Node #4) →Node #4 → (Lk_Node #4_Node #6) →Node #6 → (Lk_Node #6_Node #7) →Node #7

Thus, we would conclude that this is likely to be the second shortest path.

Minimum cost of this "second choice" path: Node #1 to Node #7 is = 60ms.



c) Depending on the answer that you select, there are no links in common with the original shortest path if we choose the disjoint one (by definition!) If we choose the second approach, the common links are: 4 – 6 and 6 – 7.

d) No links in the disjoint case and 2 links in the second case.

4.5 How does the version of the Dijkstra algorithm that you have used in part 4.1 of this question differ from the version of this algorithm used in the Internet routing protocol known as "OSPF" (Open Shortest Path First)?

Answer:

The original Dijkstra algorithm was designed and conceived as a centralised algorithm, OSPF has translated this into a distributed algorithm using the flooding algorithm.

Question 5

5.1 (a) What is the purpose of a subnet mask?

(b) Is the subnet mask 255.255.0.255 valid for a Class A address? Explain.

Answers:

(a) The subnet mask enables us to subdivide addresses to achieve more useful mixes of host and subnets for a given range of addresses.

(b) The subnet mask given is represented in binary notation as:

11111111 11111111 00000000 11111111

A valid subnet mask must consist of contiguous ‘1’s and hence this proposed mask does not satisfy this requirement since the ‘1’s are not contiguous and contain a gap.

5.2 Consider the following internet address: **136.27.32.104**

a) Convert this address into Binary format.

b) Convert this address into Hex format.

c) What class does this internet address represent?

d) If we apply a subnet mask of **FFFFFE00,** obtain the relevant network, subnet and host addresses for the given internet address.

Answers:

a) In Binary format the address is

136.27.32.104 = **10001000 00011011 00100000 01101000**

b) In Hex format the address is

136.27.32.104 = **88.1B.20.68**

c) It is a B class address since the first two bits are **10**

d) The subnet mask is:

FFFFFE00 = **11111111 11111111 11111110 00000000 = /23**

23 bit mask = 7 bits for subnet and 9 bits for host, so we have 128 subnets and 510 hosts per subnet.

Network address only details
Hex: 881b0000
Octal: 21006600000
Decimal: 2283470848
Binary: 10001000 00011011 00000000 00000000
IP Address: 136.27.0.0

Subnet address only details
Hex: 2000
Octal: 20000
Decimal: 8192
Binary: 00000000 00000000 00100000 00000000
IP Address: 0.0.32.0
Subnet #: (16)

Host address only details
Hex: 68
Octal: 150
Decimal: 104
Binary: 00000000 00000000 00000000 01101000
IP Address: 0.0.0.104

+ + + + + + + +