COMPSCI 314 S2T 2003 - Data Communication Fundamentals

Assignment 4

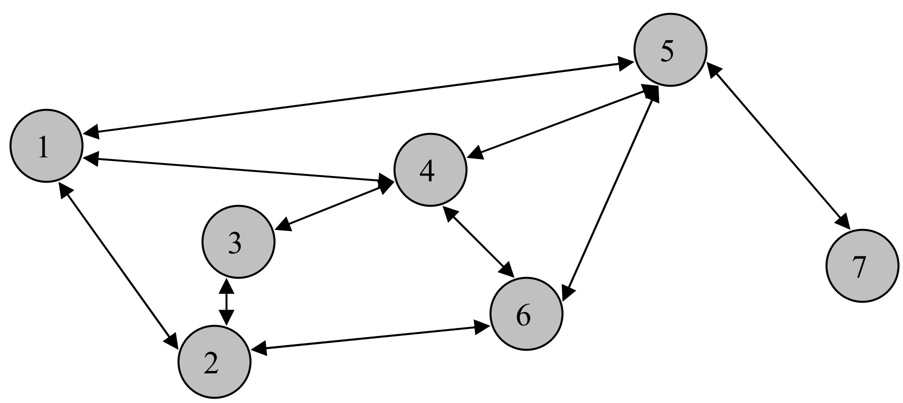
Lecturer: Ulrich Speidel (ulrich@cs.auckland.ac.nz)

Hand-in date: October 27, noon, via the Tamaki Student Resource Centre. They must be marked at the top with your name, student ID, UPI, and with "COMPSCI314S2T"

Getting help: The best source of help is monitoring and keeping your class e-mail! To avoid swamping you with messages from your classmates asking the same questions over and over again, we won't answer the same question twice. If you can't make sense of an answer I gave on class e-mail, include a copy of our response as proof that you have read it, and we'll gladly reply. Note also that the answer to your question may be in the response to another student. Yang Jia is of course also available to help you and will answer questions in his tutorials and office hours.

Problem 1 (Dijkstra's Algorithm and Routing Tables, 60 marks):

Consider the following network with seven nodes:



The double arrows between the nodes are bi-directional links. The cost of these links depends on your student ID as follows: The node number *n* indicates the *n*th digit of your student ID number. The cost of a link between two nodes is the sum of the two digits concerned.

Example: Your student ID is 8025703. The cost of the link between nodes 3 and 4 is thus:

$$cost = digit3 + digit4 = 2 + 5 = 7$$

If the two digits concerned are both 0, then the link is deemed to be non-existent or not working.

Your first task is to compute the link costs for all ten direct links. In the above example, you'd get:

from node	to node	cost		from node	to node	cost
1	2	**8**		3	4	**7**
1	4	**13**		4	5	**12**
1	5	**15**		4	6	**5**
2	3	**2**		5	6	**7**
2	6	**no link**		5	7	**10**

Enter your own link costs into the table below and copy the table to your submission:

from node	to node	cost		from node	to node	Cost
1	2			3	4	
1	4			4	5	
1	5			4	6	
2	3			5	6	
2	6			5	7	

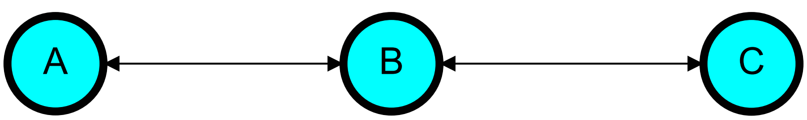
Use Dijkstra's algorithm to find the cheapest path from each node to each other node (if such a path exists). Then compute a centralized routing table. Each row in the table corresponds to the routing policy for one node, and each column to the final destination of the message. Each entry is the **number of the respective next node** along the cheapest path to the destination:

	destination node							
		1	**2**	**3**	**4**	**5**	**6**	**7**
origin node	**1**	**-**						
	2		**-**					
	3			**-**				
	4				**-**			
	5					**-**		
	6						**-**	
	7							**-**

Where no path exists, put an "x".

<u>Problem 2 (Bellman-Ford algorithm, 40 marks)</u>

Consider a very simple network that uses the Bellman-Ford algorithm to compute routing tables at each node. The network consists of three nodes A, B, and C that are connected via two bidirectional links. The topology of the network is shown below:



For simplicity, presume that the cost of each link is 1, at least as long as the link actually works. Each node can reach itself with a cost of 0. To make Bellman-Ford work, presume that each node can find the node at the other end of its link(s) automatically, e.g., via polling, and that each node periodically broadcasts to each of its immediate neighbour(s) an excerpt of its own destination table, containing the destinations it knows of and the cost involved in reaching them. Upon receiving the routing table from its neighbour, each node will update its own table with the lowest cost route.

After the first set of polls, but before the first broadcast, A knows that it can reach B via B at a cost of 1, B knows that it can reach A via A at a cost of 1 and C via C at a cost of 1. C knows that it can reach B via B at a cost of 1. Trivial so far? OK, now presume that A starts broadcasting: A tells B that it can reach B at a cost of 1. Then it is B’s turn to update its table (in this case it doesn’t change) and broadcast to A and C. Then C broadcasts to B. What is the content of the routing table at C after all three broadcasts and updates have taken place?

Now presume that the link between B and C goes down. B detects this because its poll to C remains unanswered. In the next round of broadcasts, A again broadcasts first, followed by B. What is the content of the routing table at B after the broadcasts and updates have taken place? What happens after the next round of broadcasts?

Your answer must address the three questions above and must briefly state the consequences that the behaviour of the pure Bellman-Ford algorithm has in this case.

-------------------------------------End of Assignment --