COMPSCI 314 S2C Assignment 3
2010

Department of Computer Science
The University of Auckland

*This assignment contributes **5%** of your overall course mark. Submit your assignment in **PDF** format to the **Assignment Drop Box**. Include all **workings** and **explanations**. Marks will be deducted for ambiguous solutions. Zero marks are awarded if the answers contain no explanation. Also, refer to the Departmental Policy on Cheating on Assignments.*

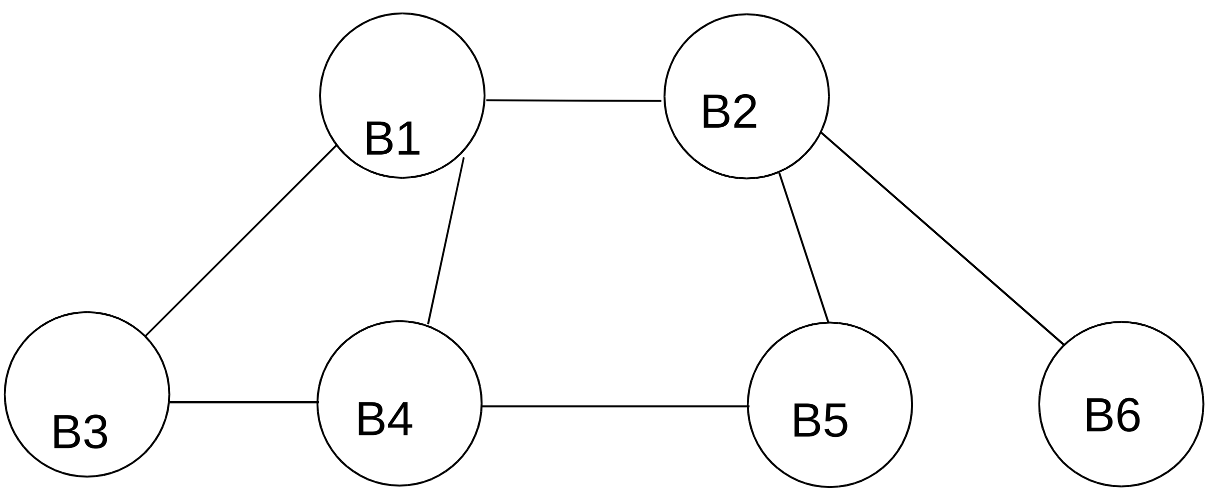
Assignment Drop Box (https://adb.ec.auckland.ac.nz/adb/).
Departmental Policy on Cheating on Assignments
(http://www.cs.auckland.ac.nz/administration/policies/CheatingPolicy.php)

[Note to Markers: the order in which the links are listed doesn't matter]

[Total: 50 marks]

Q1. Bridging [20 marks]

a) Consider the following network of LAN bridges. In practice today, these would be multiport switches, with many computers connected to each switch. Only the links *between* switches are shown.



Assume B1 is elected as the root bridge. Assuming the link metrics are all the same, list the links in a valid spanning tree (e.g. B1-B2, etc.). Which link or links will not be used? **[5 marks]**

B1-B2, B1-B3, B1-B4, B2-B5, B2-B6 (B3-B4 and B4-B5 not used)

or
B1-B2, B1-B3, B1-B4, B2-B6, B4-B5 (B3-B4 and B2-B5 not used)
[Markers: either answer is valid]

b) There are 7 links in the above network. Which of them are "redundant"? (A link is redundant if breaking it would still allow every node to communicate with all the others.) **[5 marks]**

They are all redundant except B2-B6

c) Explain how the bridging and spanning tree algorithms respond if links B1-B4 and B2-B5 both break. What links will be in the final spanning tree? **[5 marks]**

After a timeout, the spanning tree algorithm starts again and will also cause the bridging algorithm to restart at the flooding stage. The final spanning tree will be

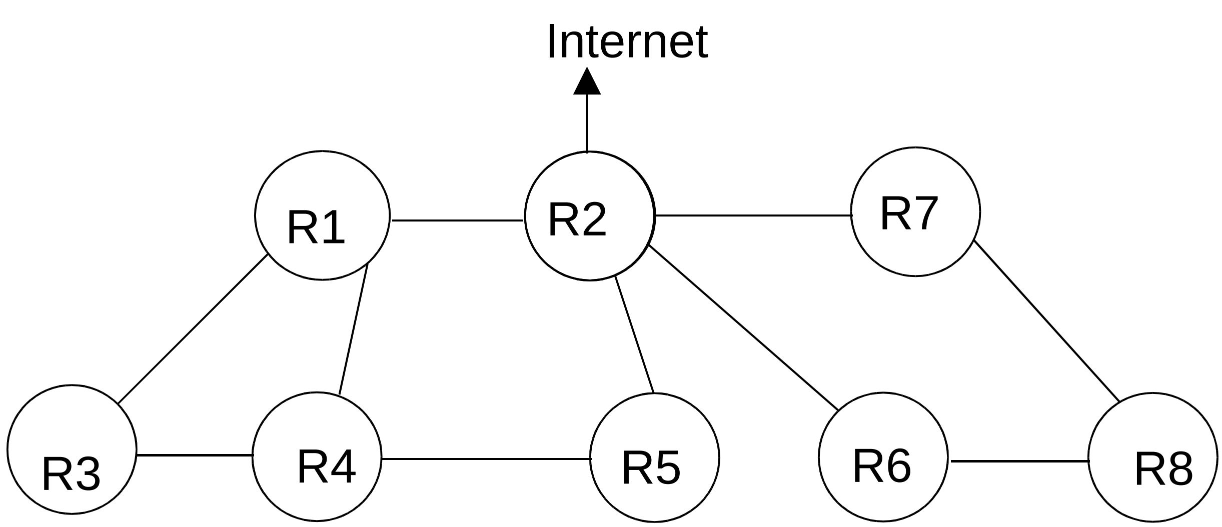
B1-B2, B1-B3, B2-B6, B3-B4, B4-B5

d) Can you improve the reliability of the original network by moving one end of one link? Explain your answer. **[5 marks]**

Move B2-B5 to become B6-B5. That makes all links in the network redundant and now a failure of B2-B6 doesn't cut off B6.

Q2. Routing [30 marks]

a) Consider the following network of routers, where all the paths have equal weight (1).



Calculate and list the shortest distances from **R2** to each other R, as the Dijkstra (shortest path first) algorithm would find them. **[5 marks]**

R2-R1=1 R2-R6=1
R2-R3=2 R2-R7=1
R2-R4=2 R2-R8=2
R2-R5=1

b) Did you find any equal choices (with two shortest paths of the same length)? If so, which ones? **[5 marks]**

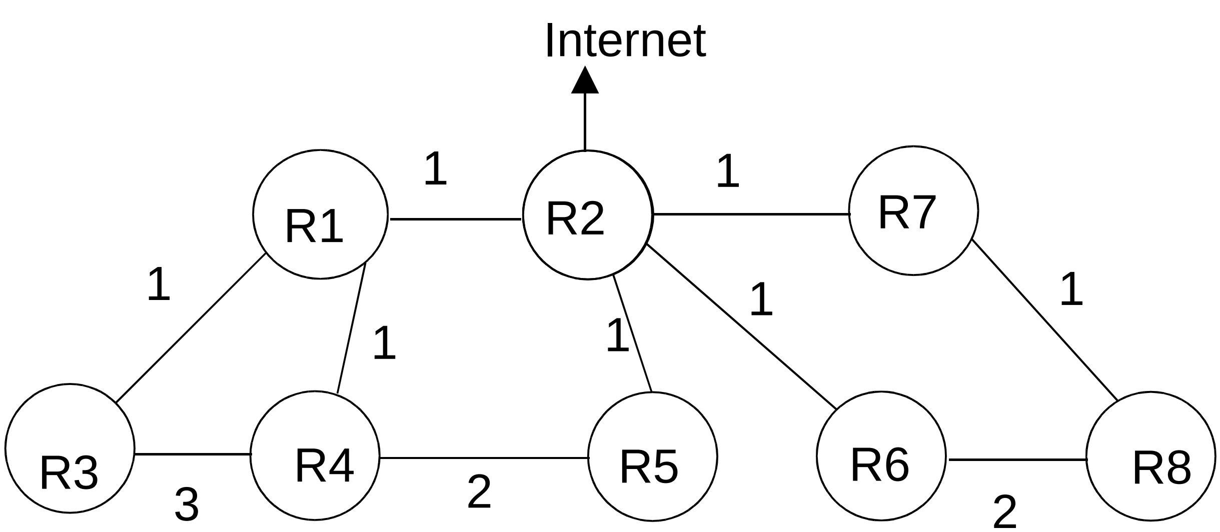
Yes
R2-R1-R4 and R2-R5-R4
R2-R6-R8 and R2-R7-R8

c) Recalculate the shortest distances if the links between R2-R1 and R2-R6 stop working. Which paths have definitely changed? Which ones might have changed, and why? **[10 marks]**

R2-R1=3+ R2-R6=3+
R2-R3=3+ R2-R7=1
R2-R4=2* R2-R8=2*
R2-R5=1 [3 marks]

+ These three paths definitely get longer. [3 marks]
* These two paths might change, to bypass the faulty links. It depends which choice the Dijkstra algorithm made between the answers to part b). [4 marks]

d) Repeat part a) but with different weights on some links, as shown below. How do the results compare to part a)? **[5 marks]**



R2-R1=1 R2-R6=1
R2-R3=2 R2-R7=1
R2-R4=2 R2-R8=2
R2-R5=1

There's no change in the shortest distances.

e) What is different now for parts b) and c)? **[5 marks]**

The paths that were equal in part b) are no longer equal; paths R2-R1-R4 and R2-R7-R8 are shortest.

In part c), R2-R4 definitely changes to the path R2-R5-R4 but R2-R8 definitely doesn't change.
