

Deliverable: Assignment #1
Title: CS436: Distributed Computer Systems
Offering: Winter 2011

WWW: http://bit.ly/uw_446-11 [The web page will be periodically updated]
Twitter: @cs436 (<http://twitter.com/cs436>) [Important updates broadcast here]

Lectures: Tuesday & Thursday 0830 - 0950 MC 4064

Instructor: Dr. Reid Holmes; DC 3351. Office hours by appointment. rth.cs436@gmail.com
TAs: Ali Abedi; DC 3549. Office hours by appointment. a2abedi@cs.uwaterloo.ca

Due: 0800 on Jan 27 via email to rth.cs436@gmail.com

Notes:

This assignment can be completed individually or in a team of up to two people. Only one copy of the assignment needs to be submitted, but make sure that **both** team members names are on the assignment page and any source code files. You should submit your answers to the questions as a PDF called <first-last>[_first-last]_a1.pdf; your source code files / scripts / README should be submitted as a zip file called <first-last>[_first-last]_a1.zip (if you are performing the assignment alone obviously the second first-last is not required). Email the assignment by 0800 on Jan 27 to rth.cs436@gmail.com. Late assignments will not be accepted.

Questions:

1) Assume the following code:

```
0000000000  
0000011111  
0011100011  
0011111100  
1100100101  
1100111010  
1111000110  
1111011001
```

- a) (1 point) What is the minimum Hamming distance of this code?
- b) (1 point) What maximum bit errors can be detected?
- c) (1 point) What maximum bit errors can be corrected? Why?

2) (1 point) Why do routers require more computational capacity than switches?

3) Assume a network that uses a "token passing" medium access control protocol with the following parameters:

- 10 stations
- ring length: 500 m
- frame size: 1000 bytes (assume all 1000 bytes are useful data)
- transmission rate: 100 Mbit/s
- propagation speed: 200,000,000 m/s
- per-station token delay (token forwarding, frame forwarding): 0.000001 s
- token size is negligible
- After sending a frame, the sending station waits for the frame to travel around the ring. It then releases the token to give other stations the possibility to send. The station can send next when the token arrives again.

a) (2 points) What is the overall efficiency, if only a single station has data to send?

b) (2 points) What is the overall efficiency, if all stations have data to send?

4) (2 points) Assume an Ethernet network with the following parameters:

- maximum distance between nodes: 1000 m
- transmission rate: 10 Mbit/s
- propagation speed: 200,000,000 m/s

a) Compute the minimum frame size for CSMA/CD to guarantee that all collisions can be detected by a sender.

5) (10 points) Implement the Dijkstra algorithm in your favourite programming language. Be sure to include the source code along with a suitable Makefile/script, etc. (depending on your programming language) as a zip.

Make sure that your submission works in the student.cs environment and include a file named README that explains how to compile (if necessary) and invoke the program. Your program must adhere to the following interface:

```
./dijkstra <origin> <file>
```

<origin> is the number of the node at which the algorithm is executed. <file> is the name of an input file that lists multiple network links (one per line) according to the following format:

```
<node1> <node2> <cost>
```

Each line describes a symmetric link between <node1> and <node2> with cost <cost>. Nodes are identified by numbers. The program must produce output on standard output of the form:

```
<total cost> <origin> <node1> <node2> ... <destination>
```

That is, each line describes the shortest path to an destination the path's total cost. The order of the output lines does not matter.

Your program should perform basic error checking, i.e., it should not just crash when receiving faulty input, but either try to perform the computation or produce an error message and quit. As stated, you can consult 3rd-party resources ([example](#)) for this assignment, but you must implement your own solution. You can use any basic library classes that are included with the language you are using for your assignment but must not use libraries provided by others.

Sample input/output files have been provided; note that the line breaks may not be properly represented by some file editors.