Loughborough University

[NETWORK MODELLING AND PERFORMANCE]

[COP512]

Feb. 2018

Deliverables from Practical Lab work (MATLAB)

Specification:	In this section of the coursework, you need to complete 2 deliverables. Deliverable 2: You need to correctly implement Little's Result (Formula) to obtain the mean waiting time given a number of conditions and plot the resulting curves on a chart. Deliverable 3: You need to create a MATLAB based GUI, plot simulation output for a number of scenarios and comment on the prioritisation scheme used. Full details attached.
Method(s) of Presentation:	1. A report including diagrams and graphs with explanation for Deliverable 2 & 3. 2. The simulation code, results and analysis to prove that you have picked up basic MATLAB skills. 3. A hard copy of the source code (i.e. MATLAB .fig files and .m scripts). 4. A disk or CD including all source codes and report.
Assessment Guidelines:	Deliverable 2: 10%, Deliverable 3: 20%. Plagiarism will receive no mark.

MATLAB Simulation

Deliverable 2: [10%]

In this deliverable you will use MATLAB to plot a series of curves on the same chart to show the mean waiting time of a queueing system calculated using Little's Result (Formula) with a given set of conditions.

Loughborough University

Assuming:

- The system is a Discrete-time M/M/1 (Geo/Geo/1) queue with an unlimited buffer size and the service discipline is FIFO
- α is the probability of a packet arriving in any time slot
- β is the probability that a packet will depart in any time slot

Plot three separate curves on a single chart to show the value of mean waiting time W when $\alpha = 0.35$, 0.45 and 0.55, and plot β on the x axis with values from 0.7 to 0.9 in increments of 0.02.

Include a legend and add an appropriate title and axis labels. Use the following symbols for marking points on the chart:

$\alpha = 0.35$, asterisk

$\alpha = 0.45$, diamond

$\alpha = 0.55$, point

Deliverable 3: [20%]

In this deliverable you will use MATLAB to adapt an existing GUI and plot the output of a Quality of Service related simulation. It shows the effect of a priority based Quality of Service negotiation algorithm depending on the amount of network based congestion experienced.

First, download the zip archive "MATLAB Coursework" containing the simulation files from the COP512 module page on Learn. Unzip this archive and place the enclosed files within your working directory. These files contain the simulation and basic GUI that you will use.

While in the working directory containing the MATLAB coursework files, enter "courseworkGUI" at the command window. You will be presented with a basic GUI (as shown in Figure 1) containing one button. This button can be pressed to execute the simulation. The simulation will display the output for each time period in the MATLAB command window.

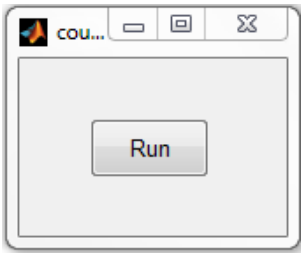

Figure 1 - Basic GUI

All functionality with regards to the simulation operation can be found within 'courseworkGUI.m'. The simulation operates as follows:

- The simulation first executes the function 'generateRandomTopology', found in the file 'courseworkGUI.m'. This creates a pseudo-randomly generated system topology composed of elements defined by the class files Node.m, Applications.m and NetworkLinks.m. The 'simulationManager' class is used to create a handle to hold all nodes within a structure.
    - System topology details (such as the number of nodes, number of applications, maximum data rates, etc.) are currently all statically configured within the 'generateRandomTopology' function.
    - Applications have a specified priority and data rate and are associated at random with the nodes.
    - Each node has one network link with which it can send data.
    - This forms a basic structure for a system where nodes contain applications that need to send data at a specified rate in order to meet their QoS. The resource limitations for sending data are defined by the maximum bandwidth of the node's network link.
- Once the topology has been created, the function 'runSimulation' is called.
    - The simulation executes for a specified number of iterations calculating the data rate required by each node given errors occurring at random on each network link. These values are placed in a 2xn matrix (where n is the number of nodes) called 'networkLinkUsage'.
    - The simulation contains the option to use a prioritisation scheme to select those applications with the highest priority such that the total bandwidth required (assuming no errors in transmission) does not exceed the maximum available. This is set by altering the 'prioritise' variable (1 for on, 0 for off) in the function 'runSimulation'.

Adapt the GUI using the MATLAB GUIDE tool to allow the user to input the following simulation parameters (with the associated variable names given) from the GUI (instead of modifying the values in class files):

- The number of nodes – ‘numOfNodes’
- The number of applications – ‘numOfApplications’
- The random seed value – ‘seed’
- The simulation time – ‘totalSimulationTime’
- Whether the simulation will perform prioritised negotiation or simply show the output of the configuration – ‘prioritise’

You now need to plot the simulation output as a graph of your choice showing the amount of network resources required as a percentage of the total available to each node at a given point in time. The graph should include the following elements:

- An indication of the network resources consumed by each node (both including and excluding the current error rate) at a given time.
- The plot should indicate when network resources required (either including, or excluding the current error rate) exceed 100% of the available resources.
- Appropriate X and Y axis labels & title
- Legend

Write the output from the simulation, ‘networkLinkUsage’, (for every node, for every time the simulation executes) to a .csv file called ‘courseworkGUIOutput.csv’.

Run the simulation with the configurations shown in Table 1 for scenarios 1-4.

Table 1: Scenario Configurations

Scenario	No. of Nodes	No. of Applications	Seed Value	Simulation Time	Prioritised Negotiation?
1	15	25	4	10	No
2	15	25	4	10	Yes
3	10	50	8	30	No
4	10	50	8	30	Yes

Compile your results into a report, including the following elements:

- A screenshot of the new GUI with annotation to show that the required elements have been included.
- Screenshots of the final plot (at time 20 or 50) for each scenario (1-4).
- Text output from the .csv files for the results of scenarios 1 and 2 formatted into a table (note that this can be split as appropriate to ensure it fits on the page, e.g. showing nodes 1-5 in one table, followed by nodes 6-10 in a separate table).

Loughborough University

- A discussion of the prioritisation scheme used in this simulation and its advantages/ limitations. Suggest one way in which the algorithm could be improved.
- An appendix containing extracts of your code for plotting the results and writing the output to a .csv file. Please add comments to this to highlight the required elements (e.g. updating results for each simulation time period).

When submitting your code be sure to include all files necessary to execute the simulation (i.e. including all files already provided to you).

The marking scheme is given as follows:

a. A GUI containing the required elements. [5%]
b. The correct plotting of the simulation output meeting the requirements. [5%]
c. Correct graphical and text-based output as required for each simulation scenario. [5%]
d. A brief discussion of the advantages/limitations of the prioritisation scheme used in this simulation. Suggest one way in which this algorithm could be improved. [5%]

Method of presentation:

1. A report including screenshots and text based simulation output with explanation for Deliverable 2 & Deliverable 3.
2. The simulation scripts, results and analysis to prove that you have picked up basic MATLAB skills.
3. A hard copy of the source code (i.e. MATLAB .fig and .m files).
4. A disk or CD including all source codes, and report.

Assessment:

Total: 30% for MATLAB. Deliverable 2: 10%, Deliverable 3: 20%.