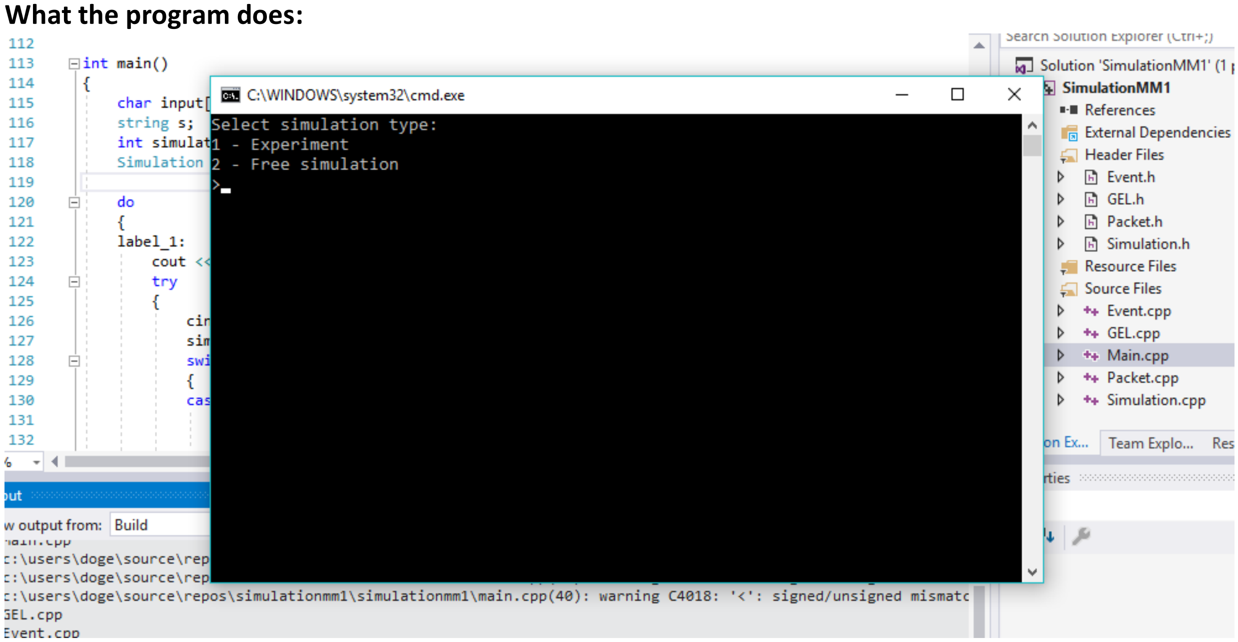
Christopher Douangnaly
SID: 998541781
ECS 152A / EEC 173A

Simulation Analysis of a Network Protocol, e.g., an IEEE 802.x Based Network

Contents

Program Compilation and Execution Instructions

- The code was not written with Linux/Ubuntu in mind and will therefore not compile on that platform.
- The code was written under Windows 10 using Visual Studio 2017 in C++. It is not guaranteed to work on any other operating system or compiler.
- **How to compile the code and run it:**
  - Method 1 (Easiest)
    - Click on \SimulationMM1\SimulationMM1.sln
    - Visual Studio should appear.
    - Build and run the program using (CTRL + F5).
  - Method 2 (Creating Project From Scratch)
    - In Visual Studio 2017: File > New > Project > **Windows Desktop Wizard**
    - **Uncheck** “Precompiled Headers” and “Security Development Lifecycle (SDL) checks”
    - Add all included header files and source files to project
    - Build and run the program using (CTRL + F5).
- **What the program does:**



  - Press 1 to simulate using the assignment’s given parameters. This gives the output to experiments #1 and #3.
  - Press 2 to simulate using any custom parameters for [lambda], [mu], [number of packets], and [buffer length].

Program Outputs

Below is a sample output of the simulation running on 1000 packets. The first half outputs the mean queue length and server utilization as a function of lambda. The second half outputs the number of dropped packets as a function of lambda.

```
Select simulation type:
1 - Experiment
2 - Free simulation
>1
Input value for packets count to simulate: 1000
.
Lambda = 0.1, mu = 1, max queue length = inf.
Experiment mean queue length = 0.1121758735654, calculated queue length =
0.111111111111111.
Experiment utilization = 9.99773463855588, calculated utulization = 10.
.
Lambda = 0.25, mu = 1, max queue length = inf.
Experiment mean queue length = 0.328448845883775, calculated queue length =
0.333333333333333.
Experiment utilization = 24.2525381020925, calculated utulization = 25.
.
Lambda = 0.4, mu = 1, max queue length = inf.
Experiment mean queue length = 0.657194496146448, calculated queue length =
0.666666666666667.
Experiment utilization = 40.3642582857015, calculated utulization = 40.
.
Lambda = 0.55, mu = 1, max queue length = inf.
Experiment mean queue length = 1.28706240579292, calculated queue length =
1.22222222222222.
Experiment utilization = 56.7555891096028, calculated utulization = 55.
.
Lambda = 0.65, mu = 1, max queue length = inf.
Experiment mean queue length = 1.84059225716492, calculated queue length =
1.85714285714286.
Experiment utilization = 62.5383968490386, calculated utulization = 65.
.
Lambda = 0.8, mu = 1, max queue length = inf.
Experiment mean queue length = 4.68315408676681, calculated queue length = 4.
Experiment utilization = 82.8668018036367, calculated utulization = 80.
.
Lambda = 0.9, mu = 1, max queue length = inf.
Experiment mean queue length = 11.6120730384528, calculated queue length = 9.
Experiment utilization = 89.828431182431, calculated utulization = 90.
.
Lambda = 0.2, mu = 1, max queue length = 1.
Number of dropped packets = 168.
.
Lambda = 0.4, mu = 1, max queue length = 1.
Number of dropped packets = 294.
.
Lambda = 0.6, mu = 1, max queue length = 1.
Number of dropped packets = 370.
.
Lambda = 0.8, mu = 1, max queue length = 1.
```

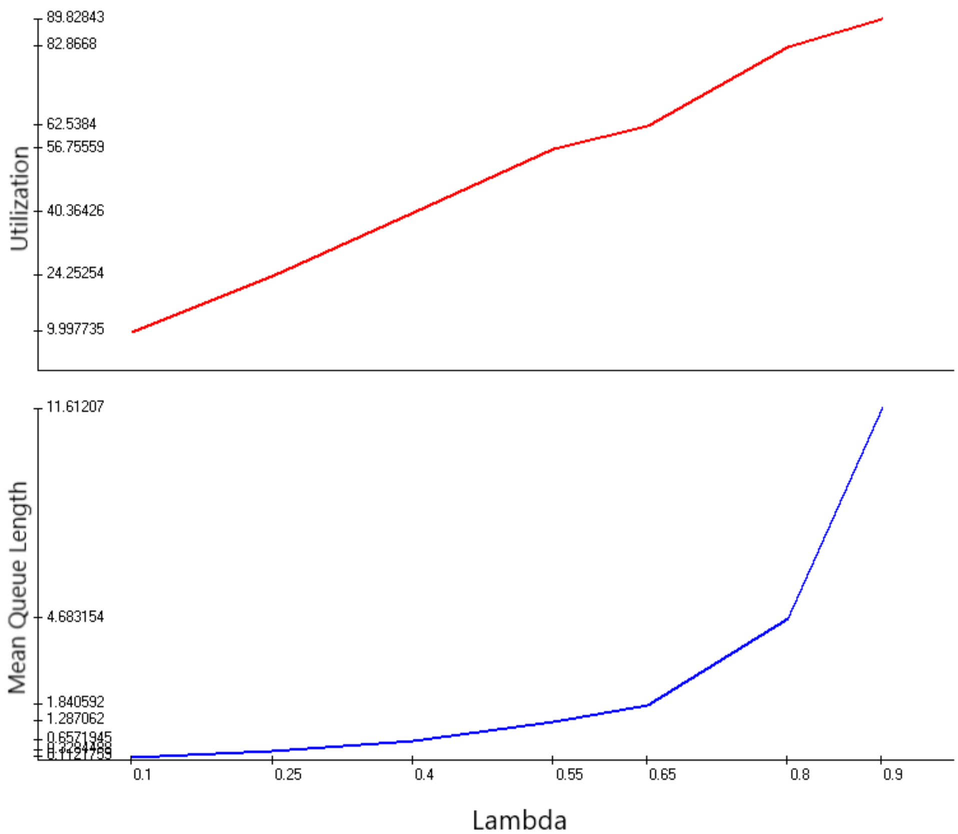
```
Number of dropped packets = 415.
.
Lambda = 0.9, mu = 1, max queue length = 1.
Number of dropped packets = 437.
.
Lambda = 0.2, mu = 1, max queue length = 20.
Number of dropped packets = 0.
.
Lambda = 0.4, mu = 1, max queue length = 20.
Number of dropped packets = 0.
.
Lambda = 0.6, mu = 1, max queue length = 20.
Number of dropped packets = 0.
.
Lambda = 0.8, mu = 1, max queue length = 20.
Number of dropped packets = 0.
.
Lambda = 0.9, mu = 1, max queue length = 20.
Number of dropped packets = 48.
.
Lambda = 0.2, mu = 1, max queue length = 50.
Number of dropped packets = 0.
.
Lambda = 0.4, mu = 1, max queue length = 50.
Number of dropped packets = 0.
.
Lambda = 0.6, mu = 1, max queue length = 50.
Number of dropped packets = 0.
.
Lambda = 0.8, mu = 1, max queue length = 50.
Number of dropped packets = 0.
.
Lambda = 0.9, mu = 1, max queue length = 50.
Number of dropped packets = 0.
.
Run new simulation [yes/no]:
```

Experiment 1 of 3: Mean Queue Length and Server Utilization

The prompt:

1. Assume that μ = 1 packet/second. Plot the queue-length and the server utilization as a function of λ for λ = 0.1, 0.25, 0.4, 0.55, 0.65, 0.80, 0.90 packets/second when the buffer size is infinite.

The graphs:



Observations:

- M/M/1 queues have the stability condition such that $utilization < 100\%$, otherwise the queue length will implode without bound. (This is probably why we weren't asked to plot when lambda = 1).
    - We can see this general correlation between the two graphs: As arrival rate approaches service rate, both utilization and mean queue length increases.
- The utilization's growth rate is approximately linear with respect to arrival rate.
- The mean queue length's growth initially lacks speed but becomes exponential as arrival rate approaches service rate.
    - At {`Lambda = 0.55, mu = 1`}, the mean queue length's growth has been building slowly.
    - Starting at {`Lambda = 0.65, mu = 1`}, as lambda begins approaching mu, the growth of the mean queue length suddenly becomes exponential.

Experiment 2 of 3: Comparing Simulation Outputs with Mathematically Expected Outputs

The Prompt:

2. Mathematically compute the mean queue lengths and the server utilization and compare with the simulation results (The mathematical formulation will be discussed in class).

How the code generates mean queue length and utilization:

```
// Get event from GEL
curr_event = gel->GetHead();
.
.
.

// Update statistics
fifo_area += (curr_event->event_time - last_event_time) * length;
last_event_time = curr_event->event_time;

.
.
.
result.mean_fifo_len = fifo_area / last_event_time;

result.utilization = 100.0 * total_service_time / last_event_time;
```

And, the equations for mean queue length and server utilization of a M/M/1 system:

$$L_q = mean\ customers\ in\ queue = mean\ queue\ length = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

$$P = server\ utilization = \frac{\lambda}{\mu}$$

Comparing the simulation outputs with mathematically expected outputs:

We will compare each simulation result (from the previous section) with the equations above. The equations will be evaluated at the appropriate values using Wolfram-Alpha (Let x = λ and y = μ). The utilization rate will be expressed as a percentage.

Simulation result:

```
Lambda = 0.1, mu = 1, max queue length = inf.
Experiment mean queue length = 0.1121758735654, calculated queue length =
0.111111111111111.
Experiment utilization = 9.99773463855588, calculated utulization = 10.
```

Wolfram-Alpha:

Input interpretation:

$$\frac{x^2}{y(y-x)} \text{ where } x = 0.1,\ y = 1$$

Result:

0.0111111

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.1,\ y = 1$$

Result:

10.

Simulation result:

```
Lambda = 0.25, mu = 1, max queue length = inf.
Experiment mean queue length = 0.328448845883775, calculated queue length =
0.333333333333333.
Experiment utilization = 24.2525381020925, calculated utulization = 25.
```

Wolfram-Alpha :

Input interpretation:

$$\frac{x^2}{y(y-x)} \text{ where } x = 0.25,\ y = 1$$

Result:

0.0833333

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.25,\ y = 1$$

Result:

25.

Simulation result:

```
Lambda = 0.4, mu = 1, max queue length = inf.
Experiment mean queue length = 0.657194496146448, calculated queue length =
0.666666666666667.
Experiment utilization = 40.3642582857015, calculated utulization = 40.
```

Wolfram-Alpha :

Input interpretation:

$$\frac{x^2}{y(y-x)} \text{ where } x = 0.4,\ y = 1$$

Result:

0.266667

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.4,\ y = 1$$

Result:

40.

Simulation result:

```
Lambda = 0.55, mu = 1, max queue length = inf.
Experiment mean queue length = 1.28706240579292, calculated queue length =
1.22222222222222.
Experiment utilization = 56.7555891096028, calculated utulization = 55.
```

Wolfram-Alpha :

Input interpretation:

$$\frac{x^2}{y\,(y-x)} \text{ where } x = 0.55,\ y = 1$$

Result:

0.672222

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.55,\ y = 1$$

Result:

55.

Simulation result:

```
Lambda = 0.65, mu = 1, max queue length = inf.
Experiment mean queue length = 1.84059225716492, calculated queue length =
1.85714285714286.
Experiment utilization = 62.5383968490386, calculated utulization = 65.
```

Wolfram-Alpha :

Input interpretation:

$$\frac{x^2}{y\,(y-x)} \text{ where } x = 0.65,\ y = 1$$

Result:

1.20714

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.65,\ y = 1$$

Result:

65.

Simulation result:

```
Lambda = 0.8, mu = 1, max queue length = inf.
Experiment mean queue length = 4.68315408676681, calculated queue length = 4.
Experiment utilization = 82.8668018036367, calculated utulization = 80.
```

Wolfram-Alpha :

Input interpretation:

$$\frac{x^2}{y\,(y-x)} \text{ where } x = 0.8,\ y = 1$$

Result:

3.2

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.8,\ y = 1$$

Result:

80.

Simulation result:

```
Lambda = 0.9, mu = 1, max queue length = inf.
Experiment mean queue length = 11.6120730384528, calculated queue length = 9.
Experiment utilization = 89.828431182431, calculated utulization = 90.
```

Wolfram-Alpha:

Input interpretation:

$$\frac{x^2}{y\,(y-x)} \text{ where } x = 0.9,\ y = 1$$

Result:

8.1

Input interpretation:

$$\frac{x}{y} \times 100 \text{ where } x = 0.9,\ y = 1$$

Result:

90.

Comparison Findings

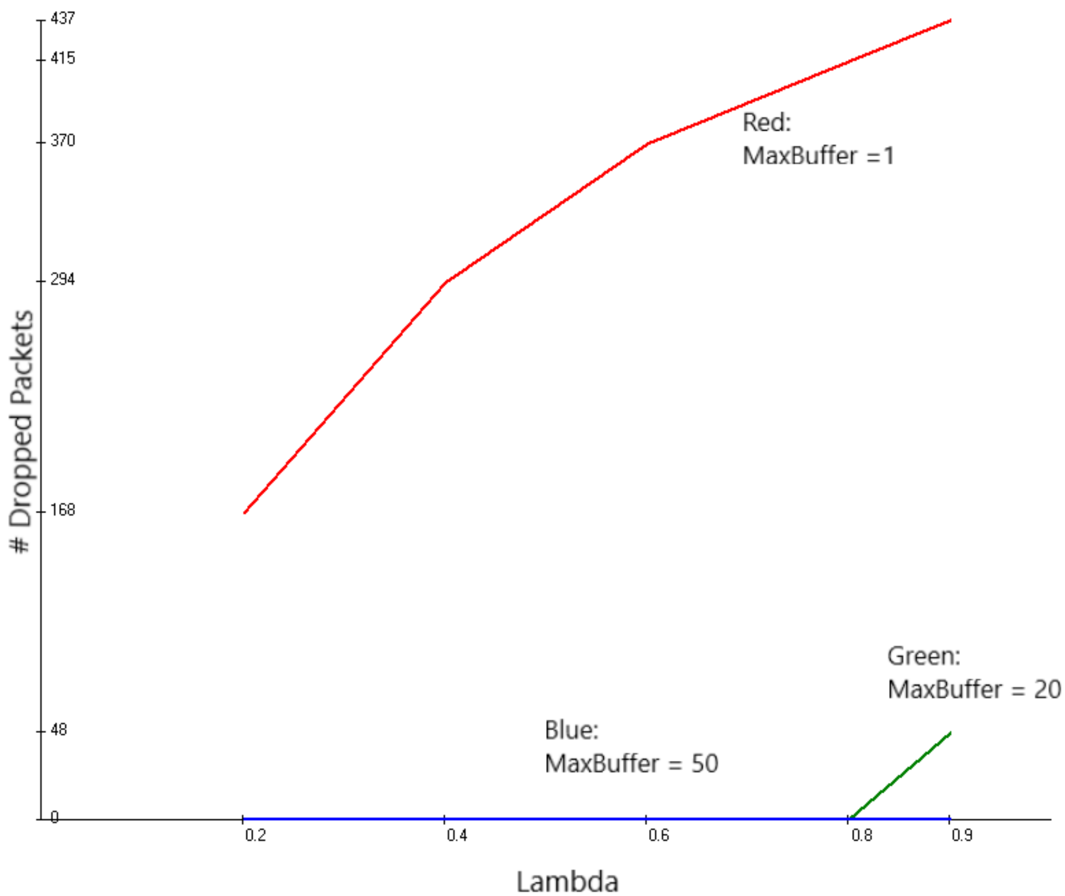
- The first simulation result completely matches with the output of the equations.
- Each of the rest of the simulation results are very close to the outputs of the equations.

Experiment 3 of 3: Number of Packets Dropped

The prompt:

3. Assume that μ = 1 packet/second. Plot the total number of dropped packets as a function of λ for λ = 0.2, 0.4, 0.6, 0.8, 0.9 packets/second for MAXBUFFER = 1, 20, and 50.

The graph:



Observations:

- Recall that the program's sample output was from a simulation on 1000 packets.
- [Red, BufferSize = 1]: With such a low buffer size, as arrival rate approaches service rate, the queue builds up quicker and quicker, resulting in a greater number of packets dropped each time.
- [Green, BufferSize = 20]: As buffer size increases, the number of dropped packets decreases. In this graph, packet drops are seen only when the arrival rate approaches 90% of the service rate, resulting in 48 packets being dropped.
- [Blue, BufferSize = 50]: At the largest buffer size available, we have zero packet drops throughout all given values of lambda.