M/M/1 & VOIP QUESTIONS

Course : Computer Networks/Telecomms
Yuan Wang
A00258427

1. Aim

<u>Part one:</u>

This experiment needs to use OPNET to create an M/M/1 model, and use the Node and Project editors to construct an M/M/1 queue, collect statistics about the model, run a simulation, and analyze the results to answer the follow questions:

1. Does the average time a packet waits in the queue exceed an acceptable limit?
2. Does the queue size increase monotonically, or does it settle down at some point? If the queue size does not settle down, this might be an indication that the system (especially the server, in this case) is overloaded.

<u>Part two:</u>

This part focusing on the link utilization and analysis the relationship between server rate and delay.

<u>Part tree:</u>

This part main focusing on answer three questions:

1. Is it possible to take traces of real traffic from a live network and model it in Opnet. How?

2. How are simultaneous events handled in Opnet ?

3. Show and explain an appropriate “process mode” for a VoIP traffic source? Using a suitable diagram detail how such a model would operate.

2. Objective

<u>Part one:</u>

This experiment main objective is that we need to analyze the queue size and average delay in the network which one used M/M/1 model.

<u>Part two:</u>

The part main objective is that keep the mean size of each packet that is produced by the server does not change (equal to 9000 bits per second), then change the server rate get different link utilization and analysis how delay changes.

Part three:

This part focusing on three questions that introduced in Aim part three.

3. Introduction&Background

M/M/1 queue:

an M/M/1 queue is one in which there is one server (and one channel) and both the interarrival Time and service time are exponentially distributed (Web.mst.edu, 2018). in the queue theory, and the M/M/1 queue represents the length of the queue in the system. This model is the most basic queuing model(Sturgul, 2000), and for many of the metrics of interest in the model, an attractive research object can be obtained as a closed form expression.

VoIP Source Model:

Voice over Internet Protocol (VoIP) is a voice call technology that enables voice calls and multimedia conferences via Internet Protocol (IP), that is, via the Internet. Other informal names are IP telephony (IP telephony), internet telephony and so on.The basic principle of VoIP (Ip.eap.gr, 2018) is to compress the speech data encoding by the voice compression algorithm, then pack the voice data according to the TCP/IP standard, send the data packet to the receiving place through the IP network, and then string the voice data packets. After decompression processing, the original voice signal is restored, thereby achieving the purpose of transmitting voice by the Internet.

FSM:

Finite State Machine Contains operations for creating a state transition diagram of the finite state machine that defines the process model. It is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some external inputs; the change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the conditions for each transition(En.wikipedia.org, 2018). In our model have three different functions create state, create transition and set initial state. Create state use to creates a new state within the process model being edited. Create transition used to creates transitions between states in a process model. And initial state It is possible to draw transitions with more than one line segment.

4. Method

Part one: Create M/M/1 queue model

1) Create a new node model and named it.
2) Select **processor module** from Node edit tool buttons.
3) Right-click processor module and select edit attributes set name attribute to src. Change process model attribute to simple_source, open “packet Interarrival Time” Specification dialog box.
4) Select exponential from the Distribution name pop-up menu. Make sure that the **Mean outcome** is set to 1.0, then click OK. Change the **Packet Size** attribute so that Distribution name is exponential and **Mean outcome** is 9000 then close attributes dialog box.
5) Select **Create Queue** from Node edit tool buttons.
6) Right-click queue model select edit attributes change name to queue. Then Change the process model attribute to acb_fifo. Make sure the service_rate attribute is set to 9600 close this attributes dialog box.
7) Select **processor module** from Node edit tool buttons.

8) Right-click processor module and select edit attributes set name attribute to sink note default value of the process model attribute is sink then close attributes dialog box.

9) Select **create packet stream** from Node edit tool buttons. Connect the src module to the queue module and connect the queue module with the sink module.

10) Select **Node interfaces** from **Interface,** In the **node types**, change the **Supported** value to no, **fixed** to yes. Then close this dialog box.

11) In **Model list** select your mm1 model change it to included then creating the network model. Click and drag your mm1 model from the object palette to the workspace right-click select **set name** and set name to mm1.

12) Defining Statistics to Collect.select Choose Individual DES Statistics from the Object pop-up menu.Expand hierarchy and Select the **queue size (packets)** and **queuing delay (sec)** statistics. Set the simulation **Duration** to **7 hours** and **Seed** to 195. Run and get result.

Part two:

1) Open mm1 queue model.

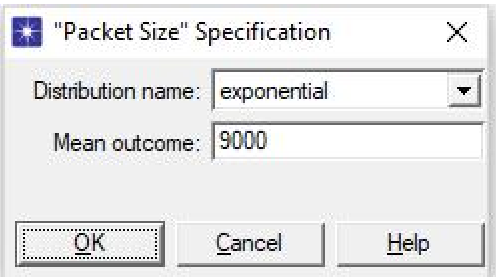
2) Right-click queue and select edit attributes.

3) Change the server rate to 10000, 11250, 15000, 18000, 22500. Then separately run and collect the delay changes under different server rates.

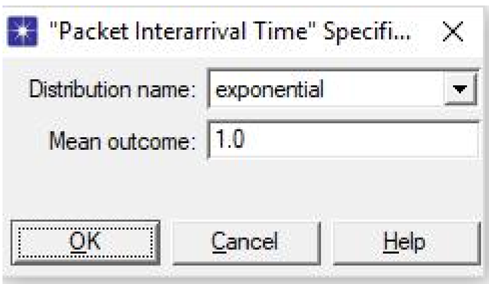
4) Get different delays and plot a graph of latency vs link utilization.

5. Result

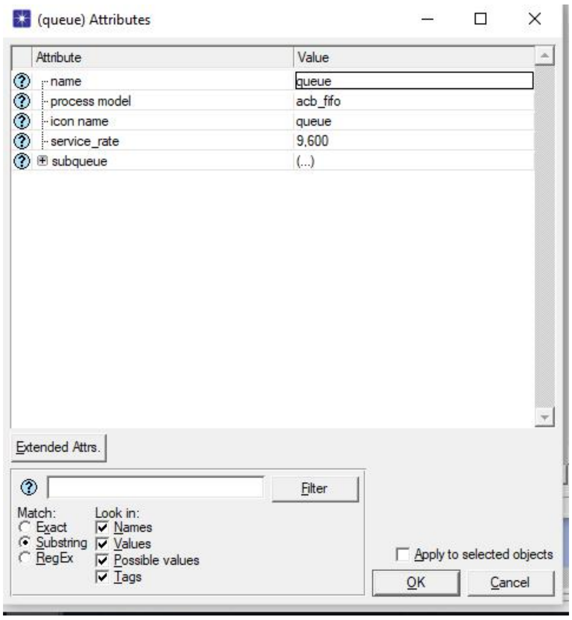
Part one:



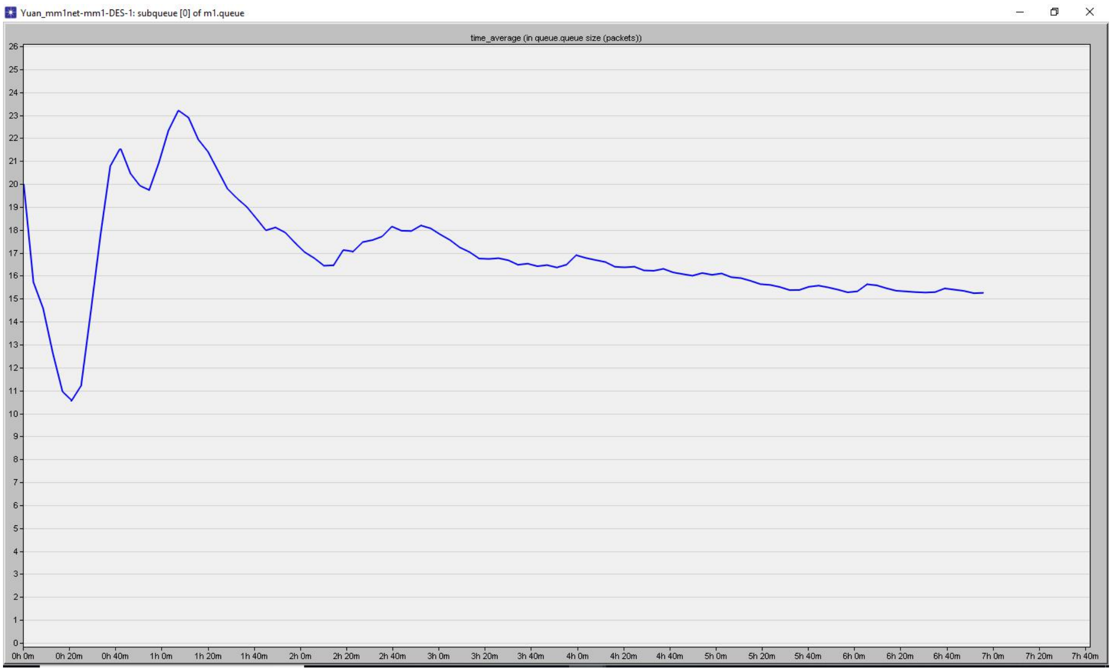
(table1:packet size)



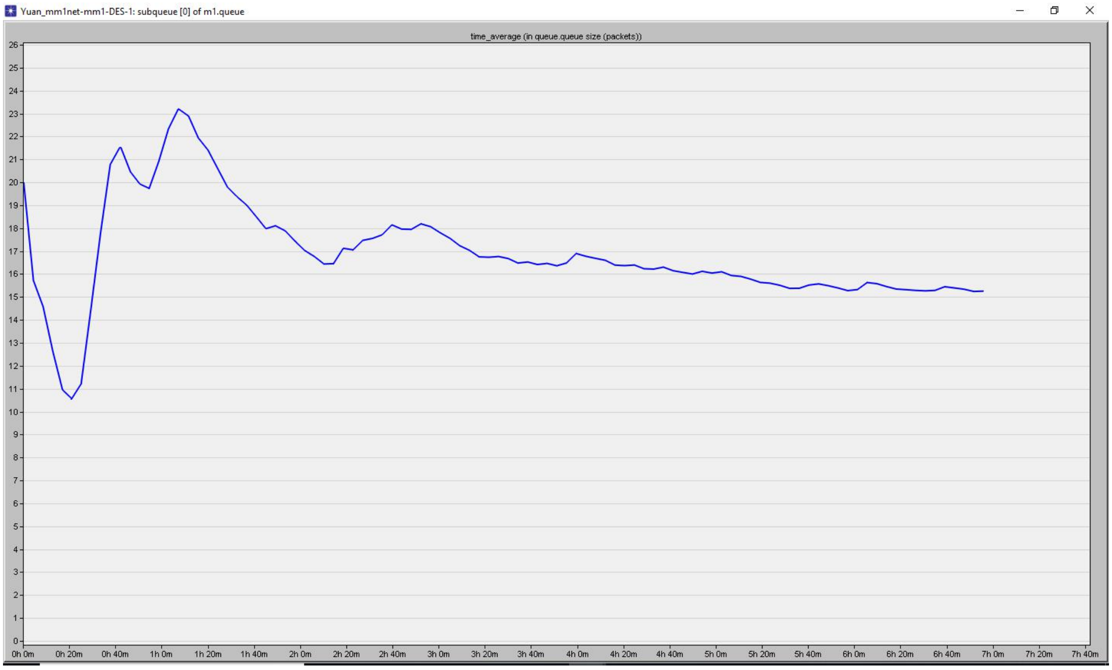
(table2:packet interarrival time)



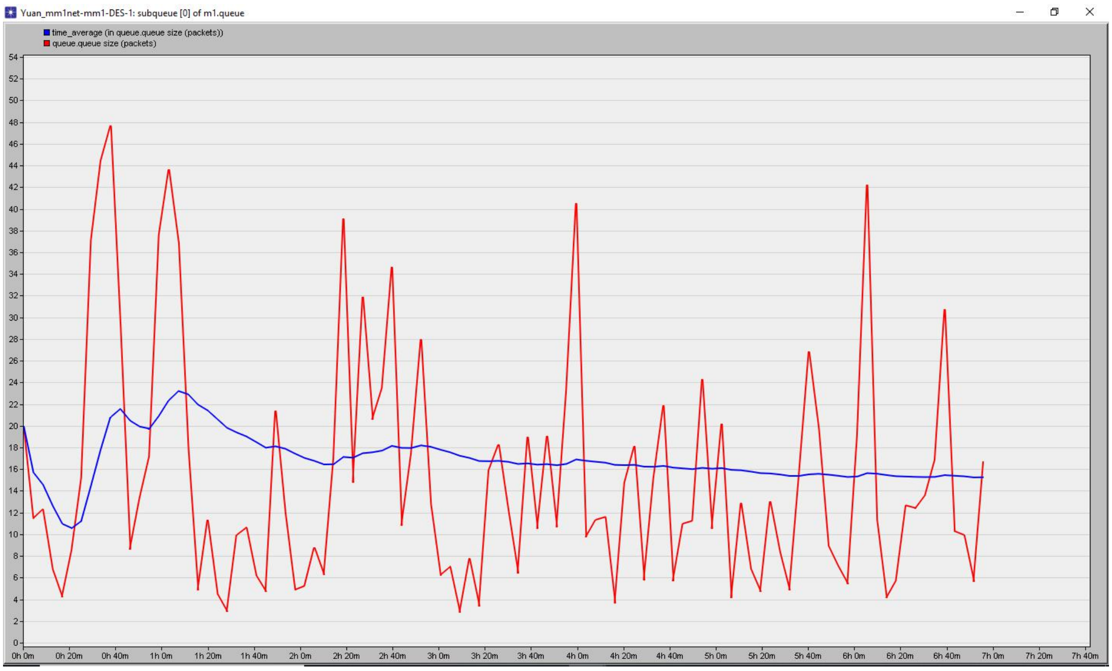
(table3: queue attributes)



(figure1:Time-averaged Queue Size)



(figure2:Average Queuing Delay)



(figure3:Queue Size, Time-averaged and Instantaneous)

Part two:

$\rho(\mathcal{R}ho) = 9000/10000 = 0.90$

$\rho(\mathcal{R}ho) = 9000/12000 = 0.75$

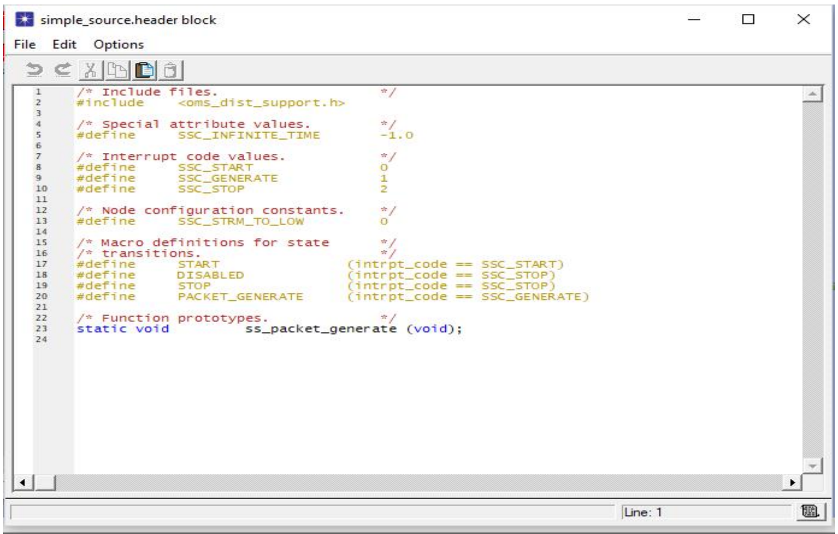
$\rho(\mathcal{R}ho) = 9000/10500 = 0.86$

$\rho(\mathcal{R}ho) = 9000/15000 = 0.65$

$\rho(\mathcal{R}ho) = 9000/11250 = 0.80$

	A	B	C	D	E
	link utilization	delay			
	0.9	9			
	0.86	6			
	0.8	4			
	0.75	3			
	0.65	1.5			

(figure4:Illustration of waiting time vs link utilization)




```
/* Include files.                  */
#include     <oms_dist_support.h>

/* Special attribute values.       */
#define      SSC_INFINITE_TIME     -1.0

/* Interrupt code values.          */
#define      SSC_START             0
#define      SSC_GENERATE          1
#define      SSC_STOP              2

/* Node configuration constants.   */
#define      SSC_STRM_TO_LOW       0

/* Macro definitions for state     */
/* transitions.                    */
#define      START                (intrpt_code == SSC_START)
#define      DISABLED             (intrpt_code == SSC_STOP)
#define      STOP                 (intrpt_code == SSC_STOP)
#define      PACKET_GENERATE      (intrpt_code == SSC_GENERATE)

/* Function prototypes.            */
static void          ss_packet_generate (void);
```

(figure5:Illustration of HB coding)


```
static void
ss_packet_generate (void)
	{
	Packet*				pkptr;
	double				pksize;

	/** This function creates a packet based on the packet generation		**/
	/** specifications of the source model and sends it to the lower layer. **/
	FIN (ss_packet_generate ());

	/* Generate a packet size outcome.					*/
	pksize = (double) ceil (oms_dist_outcome (pksize_dist_ptr));

	/* Create a packet of specified format and size.	*/
	if (generate_unformatted == OPC_TRUE)
		{
		/* We produce unformatted packets. Create one.	*/
		pkptr = op_pk_create (pksize);
		}
	else
		{
		/* Create a packet with the specified format.	*/
		pkptr = op_pk_create_fmt (format_str);
		op_pk_total_size_set (pkptr, pksize);
		}

	/* Update the packet generation statistics.			*/
	op_stat_write (packets_sent_hndl, 1.0);
	op_stat_write (packets_sent_hndl, 0.0);
	op_stat_write (bits_sent_hndl, (double) pksize);
	op_stat_write (bits_sent_hndl, 0.0);
	op_stat_write (packet_size_hndl, (double) pksize);

	/* Send the packet via the stream to the lower layer.	*/
	op_pk_send (pkptr, SSC_STRM_TO_LOW);

	FOUT;
	}
```

(figure6:Illustration of HB coding)

6. Discussion

Part one:

Table 1, 2 & 3 display the parameters configuration about the M/M/1 model.

Table1 is the configure about the process model. It’s defined the packet size be sent and in our example is 9000 bits. Table2 is also the configure about the process model. This sets the mean interarrival time of a packet to 1 second. Table3 is queue model configuration, in this dialog box, the most important attribute is server rate it’s defined the server rate(bits/sec), In our part two of the experiment, it was achieved by changing this value.

Figure1,2&3 are the results about run M/M/1 queue simulation.
Figure1 display the result about time-average queue size. And we can get that in the first two hours, the image fluctuated greatly. After 20 minutes, the queue size dropped from the initial 20 to below 11, and increased to 22 or more in the next 30 minutes.In about an hour and 20 minutes the queue size reached a peak of about 23. In the next running time, it gradually decreases, and finally stabilizes and stabilizes at around 15. Figure 2 is the result of the average queue delay, which basically has the same tendency as Figure 1. From these two figures, we can know that they have a positive correlation.
Figure 3 is a combined comparison chart, in the figure, we can see two curves, one is queue size time-average, the other is queue size as-is. Queue size as-is diagram describes the instantaneous packets at each point in time. Queue size time-average diagram describes the average packets at each point in time. From this graph, we can say that the time-average does not exceed the acceptable limit of 20 seconds, nor is the queue monotonically increasing (it appears to settle down at around 5 hours). Therefore, this is a stable system.

Part two:

Figure 4 display the relationship between link utilization and delay.

$$\rho\ (Rho) = \frac{\lambda}{\mu} \quad \rightarrow \frac{Lambda}{Mu}$$

λ = 9000 bits per second	← Link is set at 1 second Intervals	μ = 9600 bits per second service rate

From this formula we can get that “ λ ” is the packet size, “ μ ” is our server rate. So we keep packet size is unchanged and get different link utilization by change server rate to 10,000, 10,500, 11,250, 12,000 and 15,000. Because we need to keep the link utilization between 0.4 and 0.9. In different server rate we run this simulation and collection different delay, then we use the above formula to calculate the link utilization with different server rate are 0.9, 0.86, 0.8, 0.75 and 0.65 and the corresponding delays are 9, 6, 4, 3 and 1.5. finally, we can get a graph like figure4. From this graph, we can get that when packets keep unchanged, with the server rate increasing the link utilization will decrease and the corresponding delay will be reduced.

Figure 5&6 display the code about how this layer specifically handles the transmitted data. Figure5 describes a function declaration(function prototype) and figure 6 is the specific implementation process of this method. In figure 6 , we can clearly understand how this function work. In this part, we main focusing on two methods:

1. Op-pk-create-fmt (format-str),
2. Op-pk-send (pkptr,ssc-strm-to-low).

Method Op-pk-create-fmt(format-str) use to create a packet with the specified format the parameter “format-str” is the data have been formatted and we need to use a packet type pkptr to receive it.

Method Op-pk-send (pkptr,ssc-strm-to-low) is used to send the data which have been formatted to the next layer. The argument “pkptr” is the one that we have use Op-pk-create-fmt(format-str) method packed. Another parameter “ssc-strm-to-low”, which I think is the method name or flag, is used to call the method to transfer the packet data to the next layer.

<u>Part three:</u>

1. Is it possible to take traces of real traffic from a live network and model it in Opnet. How?

OPNET itself is a network simulation technology package that accurately analyzes the performance and behavior of complex networks. So I think it is achievable. Especially after we have completed so many simulation experiments, I think the simulation is completely feasible. We can complete the simulation through a process called simulation modeling. OPNET's Modeler(En.wikipedia.org, 2018) module has comprehensive simulation capabilities and is the core of OPNET, powerful and user-friendly. And it can perform detailed analysis based on packages and based on mathematical statistics.

2. How are simultaneous events handled in Opnet ?

When there are multiple events in the event list at the same time, the simulation core has two ways of processing:

Natural order. Although the event is at a time, the ordering in the event list is still in order.

Set the priority. There is also a priority in the interface properties that can be set.

In other words handling synchronization events in opnet, there are two modes, the first is to deal with first-come first which is natural order, the second is to set the priority, the event with high priority is executed first.

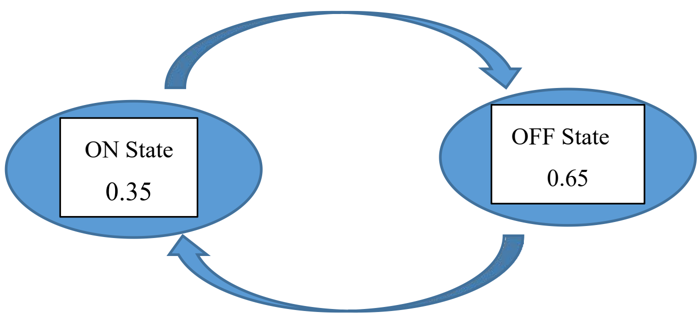
3. **Show and explain an appropriate "process model" for a VoIP traffic source. Using a suitable diagram, detail how such a model would operate.**

Voice communication over the Internet is a very complex system engineering, the most fundamental of which is VoIP (Voice over IP) technology (Federal Communications Commission, 2018).VOIP is a voice call technology, through the Internet Protocol (IP) to achieve voice calls and multimedia conferences, that is, via the Internet to communicate. It digitally processes the voice signal, compresses the code and packets it, transmits it through the network, and then decompresses it, restores the digital signal to sound, and lets the other party hear it.

The basic creation process of VOIP mainly consists of five stages.

1. The first stage is to convert the voice into transmittable analog data.
2. The second stage is to compress and encode the analog data, convert it into IP data that can be transmitted over the IP network, add corresponding headers, time stamps, and other information.
3. The third stage is transmission through the channel.
4. The fourth stage is to convert the transmitted IP packets into data through a decoder.
5. The final stage is to convert the converted data into the corresponding analog voice and play it through the speaker.

The traditional telephone network transmits voice in a circuit-switched manner, and the required transmission bandwidth is 64 kbps.



Based on the generation of a constantly fixed size packet of 384bits at 64kbps, we can get that the result of packets generated is :

64,000/384 = 166.667 $\approx$ 167 per second.

From this process model, “ON State” means the person is talking, “OFF State” means the person is listening not talking. In our process model the average “ON State ” time is 0.35, and the average “OFF State” is 0.65. And we can get that the Results in a Packet Rate is 167 $\times$ 0.35 = 58.45. basically have 58 packets be transmitted per second.

Assume the utilization is 80% then:

From this formula $\rho\ (Rho) = \frac{\lambda}{\mu} \rightarrow \frac{Lambda}{Mu}$ we can get the server rate μ.

$$\rho(\mathcal{R}ho) = \frac{(384 \times 167) \times 0.35}{\mu} = 0.8$$

We can get the server rate is (384 $\times$ 167) $\times$ 0.35 /0.8 = 28,056 bit per sec.

7. Conclusion

From this experiment, we can get that the time-average queue size did not pass the 20 seconds threshold in our simulation M/M/1 node model. And from the result after three hours, it gradually became stable, meaning that the server can afford the network, so created the network is stable which based on this network node model.

From part two we can get that the link utilization and waiting time delay have a positive relationship within the range of link utilization between 0 and 1. With the link utilization increasing the waiting time delay will increase.

Reference:

Web.mst.edu. (2018). [online] Available at: http://web.mst.edu/~gosavia/queuing_formulas.pdf [Accessed 3 Dec. 2018].

Sturgul, John R. (2000). Mine design: examples using simulation. SME. p. vi. ISBN 0-87335-181-9.

Ip.eap.gr. (2018). [online] Available at: http://ip.eap.gr/pdf/Lecture%20Pocta-3_Zilina.pdf [Accessed 3 Dec. 2018].

En.wikipedia.org. (2018). *Finite-state machine*. [online] Available at: https://en.wikipedia.org/wiki/Finite-state_machine [Accessed 4 Dec. 2018].

En.wikipedia.org. (2018). *OPNET*. [online] Available at: https://en.wikipedia.org/wiki/OPNET [Accessed 4 Dec. 2018].

Federal Communications Commission. (2018). *Voice Over Internet Protocol (VoIP)*. [online] Available at: https://www.fcc.gov/general/voice-over-internet-protocol-voip [Accessed 4 Dec. 2018].