**Name: Jeeva Krishnasamy**

**Pitt ID: 4689595**

**TELCOM 2310 Fall 2023**

**Project 1**

**Introduction**

Network devices, like routers and switches, employ queues, also referred to as buffers, to temporarily store incoming packets when they cannot be immediately transmitted on an outgoing link. These queues have a finite capacity and play a pivotal role in preventing network congestion and data loss. One of the primary concerns in network queuing is the phenomenon of packet loss. When a queue reaches its maximum capacity, any new packets arriving at the queue are dropped and lost. This can lead to data loss, network congestion, and degradation of network performance.

**Discrete event simulator**

The packet to/from the router is always a discrete event whenever it happens the queue is altered according to the arrival or departure of the packet. In this project, two discrete event simulators have been done for two experiments one for the constant input rate and the other for variable input rates with the events. The constant input rate simulator uses input parameters like arrival rate (λ), departure rate (μ) and the buffer size (n) for the number of events and the variable input rate simulator uses input parameters like arrival rate (λ), departure rate (μ), buffer size (n), number of events and the event rate for each arrival rate.

The simulation program was developed using Python, utilizing libraries such as random for generating random numbers and matplotlib for data visualization.

**Implementation: Constant Input Rate Simulator**

The program's purpose is to simulate the behaviour of packet queues in network devices, specifically routers and switches. It examines how varying combinations of arrival rates (λ), departure rates (μ), and queue sizes (n) for the number of events (1000000) the performance and behaviour of these devices.

|  |  |  |  |
| --- | --- | --- | --- |
| λ (pkt/sec) | 25 | 75 | 125 |
| μ (pkt/sec) | 60 | 100 | 125 |
| n (packets) | 60 | 100 | 150 |

**Program Development**

**PacketQueueSimulator Class**: This class represents the simulation model. It initializes with parameters like arrival rate, departure rate, buffer size, and the number of events to simulate. It simulates packet arrivals and departures over time and collects data on queue sizes and dropped packets.

* arrival\_rates: A list containing different arrival rates for various scenarios.
* departure\_rate: The fixed departure rate (μ) from the queue.
* buffer\_size: The maximum size of the packet queue (n).
* num\_events: The number of events through the program run.
* reset: The method to prepare the simulator for a new instance.
* packets\_in\_queue: A variable to track the number of packets in the queue.
* packets\_dropped: A variable to record the number of packets dropped from the queue.
* Lists time\_steps, queue\_sizes, and dropped\_events are used to store simulation data over time.

**Simulate\_event** function to simulate the packet queue. The step-by-step explanation of the code:

* For Loop: The method iterates over a range of time steps, specified by self.num\_events. Each time step represents an event in the simulation.
* **Probability Calculation:** probability\_arrival: The probability of a packet arriving at the queue during the current time step. It's calculated as the ratio of the arrival rate to the total rate (departure rate + arrival rate). probability\_departure: The complementary probability that a packet departs from the queue during the current time step.
* **Random Number Generation:** random\_number: A random number between 0 and 1 is generated using the random.random() function. This number will be used to determine whether a packet arrives or departs based on the probabilities calculated.
* **Arrival Event:** If random\_number is less than or equal to probability\_arrival, it signifies that a packet arrives. There's an additional check to ensure that the queue is not already full (self.packets\_in\_queue < self.buffer\_size). If the queue has available space, a packet is added to the queue (self.packets\_in\_queue += 1). If the queue is already full, a packet is considered "dropped" (self.packets\_dropped += 1), indicating that the packet could not be accommodated in the queue.
* **Departure Event:** If random\_number is greater than probability\_arrival, it signifies that a packet departs from the queue. There's an additional check to ensure that the queue is not empty (self.packets\_in\_queue > 0). If there are packets in the queue, one packet is removed (self.packets\_in\_queue -= 1), indicating successful departure.
* **Data Collection:** For each time step, the method records the following data:
  + time\_steps: The current time step/event number.
  + queue\_sizes: The number of packets in the queue at that moment.
  + dropped\_events: The cumulative number of dropped packets up to that point.

**Plotting Function:** The plot\_simulation\_results function generates line plots to visualize the number of packets in the queue and the packets dropped over time. These plots help analyze the behaviour of the packet queue.

**Main Function:** Parses command-line arguments, including arrival rates, departure rates, buffer sizes, and the number of events for simulation. Check if multiple scenarios (combinations of parameters) have been provided. If yes, it creates and runs simulations for each combination. It initializes a PacketQueueSimulator with specific parameters for each scenario and executes the simulation. Generates unique plot titles for each scenario and plots the simulation results. If only a single set of parameters is provided, it runs a single simulation and generates a plot for the results.

**Program Execution**

Command-Line Interface (CLI): The program is equipped with a command-line interface (CLI) that accepts user input for various parameters. The program can execute both single combinations and multiple combinations Users can specify arrival rates, departure rates, buffer sizes, and the number of events to simulate.

**Simulating Single Combinations:** This command initiates the simulation for a specific combination of arrival rate (25), departure rate (60), buffer size (60), and a specified number of events (1,000,000).

**Bash - cmd**

python packet\_queue\_simulator.py --arrival\_rates 25 --departure\_rates 60 --buffer\_sizes 60 --num\_events 1000000

**Simulating Multiple Combinations**: For simulating multiple combinations, the program accepts lists of arrival rates, departure rates, and buffer sizes. The program runs simulations for multiple combinations of parameters and generates plots and results for each combination.

**Bash – cmd**

python packet\_queue\_simulator.py --arrival\_rates 25,75,125 --departure\_rates 60,100,125 --buffer\_sizes 60,100,125 --num\_events 1000000

**Implementation: Variable Input Rate Simulator**

The program has been enhanced to simulate the behaviors of packet queues in network devices, particularly routers and switches, under varying event rates. It investigates how arrival rates (λ ), change with the change in event rate for the fixed departure rates ( μ), and queue sizes (n) for the number of events (1,000,000).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Events (%) | 0 - 10 | 10 - 70 | 70 - 80 | 80 - 90 | 90 - 100 |
| (pkt/sec) | 70 | 200 | 130 | 120 | 70 |

**Program Development**

**PacketQueueSimulator Class:** This class represents the simulation model. It initializes with lists of arrival rates, departure rates, buffer size, and the number of events to simulate. The primary modification is the introduction of dynamic arrival rate adjustment based on the event percentage.

* arrival\_rates: A list containing different arrival rates for various event percentages.
* departure\_rate: The departure rate (Œº) from the queue.
* buffer\_size: The maximum size of the packet queue (n).
* num\_events: The number of events to simulate.
* packets\_in\_queue: Tracks the number of packets in the queue.
* packets\_dropped: Records the number of packets dropped from the queue.
* time\_steps, queue\_sizes, and dropped\_events lists are used to store simulation data over time.

The simulate\_event method is the core of the simulator and controls the simulation process.

* It takes the event\_rate\_range as input, which represents the maximum event percentage for the corresponding arrival rate. The method iterates over a range of time steps, simulating events for each step.
* For each time step: It calculates the current event percentage based on the current time step. Compares the current event percentage with the specified event\_rate\_range to determine if the event falls within this range.
* Calculates the arrival rate based on the current event percentage using the get\_arrival\_rate method. Computes probabilities for packet arrivals and departures.
* Simulates packet arrivals and departures based on these probabilities. Updates the number of packets in the queue and the number of packets dropped.
* Records the simulation data in time\_steps, queue\_sizes, and dropped\_events lists.

**Arrival Rate Calculation Method (get\_arrival\_rate):**

* The get\_arrival\_rate method calculates the arrival rate based on the current event percentage. It uses a series of conditional checks to determine which arrival rate from the arrival\_rates list to use. This allows the simulator to adapt to changing event percentages during the simulation.

**Plotting Function (plot\_simulation\_results):**

The plot\_simulation\_results function generates graphical representations of the simulation results. It takes a simulator instance and a title as input. Using the matplotlib library, it creates a figure and plots the simulation data. The x-axis represents simulated events, while the y-axis displays the number of packets in the queue and packets dropped. The plot includes labels, titles, legends, and grid lines for clarity and is displayed to the user. It is saved as an image named "Variable\_input\_event.png."

**Main Program (if \_\_name\_\_ == "\_\_main\_\_"):**

The main program block allows the script to run as a standalone program. It uses the argparse library to parse command-line arguments. Input parameters, such as arrival rates, departure rate, buffer size, the number of events, and event rate ranges, are extracted from the command line.

A loop iterates over different event rate ranges provided in the command line. For each event rate range, a new PacketQueueSimulator instance is created, and the simulation is executed. Simulation results are plotted and saved as images for analysis.

**Program execution**

**Command-Line Interface (CLI):** The program accepts command-line arguments for specifying arrival rates, departure rates, buffer sizes, the number of events, and event rate ranges. The program will dynamically adjust the arrival rate based on the specified event rate ranges and generate corresponding results.

**Bash-cmd**

python variable\_event\_simulator.py --arrival\_rates 70 200 130 120 70 --departure\_rate 125 --buffer\_size 100 --num\_events 1000 --event\_rates "10-70" "70-80" "80-90" "90-100"

**Experiment 1: Top of FormAnalysis of the Simulation Result for Constant Input Rate Simulator.**

The given simulation results below from (1 – 9) show that when the rate at which packets arrive is less than the rate at which they depart, the system operates efficiently with minimal or zero packet drops. Due to the following reasons.

1. **Arrival Rate Less Than Departure Rate:** When the arrival rate is lower than the departure rate, it means that the system can process incoming packets faster than they arrive.
2. **Efficiency is high:** In such underloaded systems, packets do not queue up in large numbers, and there are typically no or very few packet drops. The system can process packets without significant delays, and resources are efficiently utilized.
3. **Queuing Delay is very low:** Even though packets are efficiently processed, there may still be some queuing delay, particularly when packets arrive sporadically or unevenly. However, the fluctuation in the queue is minimal due to the departure rate being higher than the arrival rate.
4. **Packet Drops are low:** Packet drops are more unlikely to occur in these systems, where the arrival rate is less than the departure rate.
5. **Fluctuation in a queue is very low:** Due to the arrival rate being less than the queue, the queue is used efficiently to avoid congestion and the fluctuation is very minimal compared to the overloaded systems.

Top of Form

Bottom of Form

|  |  |
| --- | --- |
| A graph with green lines  Description automatically generated | Simulation 1:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 60pkts. |

Bottom of FormIn this case, the arrival rate is less than the departure and the Buffer size of the packets getting dropped in this scenario is unlikely zero. The maximum number of packets in the queue is 16.

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Simulation 2:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 100 pkts. |

In this case the arrival rate is less than departure and the buffer size is larger than departure rate, so no packets get dropped from queue and the maximum packet in the queue is 14.

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Simulation 3:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 150 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is larger than the departure rate, so no packets get dropped from the queue and the maximum packet in the queue is 15. The queue is almost empty at most times due to the larger departure rate and the buffer size.

|  |  |
| --- | --- |
| A graph with green lines  Description automatically generated | Simulation 4:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 60 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is smaller than the departure rate but larger than the arrival rate, so no packets get dropped from the queue and the maximum packet in the queue is 11.

|  |  |
| --- | --- |
| A graph of a graph showing a green line  Description automatically generated with medium confidence | Simulation 5:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 100 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is equal to the departure rate. Still, no packets get dropped from the queue and the maximum packet in the queue is 9.

|  |  |
| --- | --- |
| A graph of a graph showing a green line  Description automatically generated with medium confidence | Simulation 6:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 150 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is larger than the departure rate. Still, no packets get dropped from the queue and the maximum packet in the queue is 10.

|  |  |
| --- | --- |
| A graph with green lines  Description automatically generated | Simulation 7:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 60 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is smaller than the departure rate. Still, no packets get dropped from the queue and the maximum packet in the queue is 9.

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Simulation 8:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 100 pkts. |

In this case, the arrival rate is less than the departure and the buffer size is smaller than the departure rate, so no packets get dropped from the queue and the maximum packet in the queue is 8.

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Simulation 9:  Arrival Rate (λ) = 25 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 150 pkts. |

In this case, the arrival rate is less than departure and the buffer size is larger than the departure rate, so no packets get dropped from queue and the maximum packet in queue is 8.

In the simulation results below from (10 -12):

* **Arrival Rate higher than Departure Rate:** When the arrival rate is higher than the departure rate, it means that the system cannot process incoming packets faster.
* **Efficiency is low:** In such underloaded systems, packets get queue up in large numbers, and there are typically very high packet drops. The system can process packets with significant delays, and resources are not utilized efficiently utilized.
* **Queuing Delay is very high:** Even though packets are not efficiently processed, and more queuing delay, particularly when packets arrive sporadically or unevenly. However, the fluctuation in the queue is high due to the departure rate is lower than arrival rate.
* **Packet Drops is high:** Packet drops are more likely to occur in overloaded systems, where the arrival rate exceeds the departure rate. In such cases, packets may need to be dropped or delayed due to resource constraints.
* **Fluctuation in queue is very high:** Due to the arrival rate is more than the departure rate, so the queue is not used efficiently to avoid congestion and the fluctuation is very high.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 10:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 60 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated with medium confidence | Close view in the event where packet start to get dropped from queue |

In the above simulation the arrival rate is greater than the departure and buffer the packets get dropped from the event of 500 and the point where the queue is full. The system is overloaded most of the time.

|  |  |
| --- | --- |
| A graph with a line drawn on it  Description automatically generated | Simulation 11:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 100 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated | Close view in the event where the packet get start drops |

In the above simulation the arrival rate is larger than the departure rate and but smaller than buffer size, so the congestion of the packets happened only after the event of the 650 and the packet getting dropped. The queue is full all the time and the arrival rate are larger due to which the packets drop eventually.

|  |  |
| --- | --- |
| A graph with a line drawn on it  Description automatically generated | Simulation 12:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 125 pkts. |

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Close view in the event where the packet get start drops |

Since the arrival rate is larger than the departure but the queue is larger than the arrival rate, Still the packets get dropped. But the packet drops happened only after the event of 1600 this is due to the buffer which is twice as the arrival rate.

In the simulation (13 -18) result shows that the

|  |  |
| --- | --- |
| A graph showing a green line  Description automatically generated | Simulation 13:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 60 pkts. |

The system faces congestion challenges due to the high incoming traffic rate and limited buffer size. The fluctuation of the queue in this scenario is high when compared to the simulation (1 -9). The maximum queue in this scenario is half the size of the buffer size.

|  |  |
| --- | --- |
| A graph with green lines  Description automatically generated | Simulation 14:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 100 pkts. |

Since the arrival rate is smaller than the departure and the buffer the queue is efficiently managed and the fluctuation in the queue is minimal when compared to the before simulation 13.

|  |  |
| --- | --- |
| A graph of green and blue lines  Description automatically generated | Simulation 15:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 150 pkts. |

Since the arrival rate is smaller than the departure and the buffer the queue is efficiently managed and the fluctuation in the queue is minimal.

|  |  |
| --- | --- |
| A graph of green lines  Description automatically generated | Simulation 16:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 60 pkts. |

Since the arrival rate is smaller than the departure and the buffer size is smaller than the arrival rate the queue is efficiently managed and the fluctuation in the queue is minimal.

|  |  |
| --- | --- |
| A graph of green and blue lines  Description automatically generated | Simulation 17:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 100 pkts. |

Since the arrival rate is smaller than the departure and the buffer is smaller than departure rate the queue is efficiently managed and the fluctuation in the queue is not quite very high, but the fluctuation is happening with the queue.

|  |  |
| --- | --- |
| A graph of green lines  Description automatically generated | Simulation 18:  Arrival Rate (λ) = 75 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 150 pkts. |

Since the arrival rate is smaller than the departure and the buffer the queue is efficiently managed and the fluctuation in the queue is minimal.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 19:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 60 pkts. |

|  |  |
| --- | --- |
| A graph with a line  Description automatically generated | Close view in the event where the packet get start drops |

The arrival rate is twice as the departure rate and the buffer size, so the packets get dropped at the event of 125 because the queue got full. The packet gets dropped when the arrival size equal to sum of the departure rate and the buffer size when both departure rate and buffer size is less than the arrival rate.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 20:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 100 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated | Close view where the packet starts to get dropped from the queue. |

The arrival rate is twice as the departure rate and the buffer size, so the packets get dropped at the event of 320 because the queue got full. The packet gets dropped when the arrival size equal to twice the sum of the departure rate and the buffer size.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 21:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 60 pkts/sec Buffer Size (n) = 150 pkts. |

|  |  |
| --- | --- |
| A graph with a line  Description automatically generated | Close view where the packet starts to get dropped from the queue. |

The arrival rate is twice as the departure rate and the buffer size, so the packets get dropped at the event of 480 because the queue got full. This is due to the buffer is still highest than the arrival, but the departure rate is very less than the arrival rate.

|  |  |
| --- | --- |
| A graph with a line drawn on it  Description automatically generated | Simulation 22:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 60 pkts. |

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Close view in the event where the packet get start drops |

In this scenario, the arrival rate is higher than the departure rate, which can lead to congestion in the packet queue. However, the limited buffer size of 60 packets means there is not much space to temporarily store incoming packets at the event of 600 the packets started to get dropped. This situation may result in frequent packet loss when the buffer becomes full, leading to dropped packets.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 23:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 100 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated with medium confidence | Close view in the event where the packet get start drops |

In this scenario, the arrival rate is higher than the departure rate, which can lead to congestion in the packet queue. However, the limited buffer size of 60 packets means there is not much space to temporarily store incoming packets at the event of 675 the packets started get dropped. This situation may result in frequent packet loss when the buffer becomes full, leading to dropped packets.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 24:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 100 pkts/sec Buffer Size (n) = 150 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated | Close view in the event where the packet get start drops |

In this scenario, the arrival rate exceeds the departure rate, which may lead to periods of congestion. However, the buffer size of 150 packets provides additional storage capacity to accommodate temporary traffic spikes and at the event of 950 the packets started get dropped from the queue. This results in better handling of incoming packets, minimizing the likelihood of packet loss.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 25:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 60 pkts. |

|  |  |
| --- | --- |
| A graph showing a graph of a graph  Description automatically generated with medium confidence | Close view in the event where the packet get start drops |

In this scenario, the arrival rate matches the departure rate, which means that the network is in a state of equilibrium. With the buffer size of 60 packets, it may be challenging to handle periods of increased traffic because the buffer can become quickly saturated, leading to packet loss. The packets started getting dropped from the queue at the event of 1900.

|  |  |
| --- | --- |
| A graph of a graph  Description automatically generated | Simulation 26:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 100 pkts. |

|  |  |
| --- | --- |
| A graph showing a line graph  Description automatically generated with medium confidence | Close view in the event where the packet get start drops |

In this scenario, the arrival and departure rates are perfectly balanced, meaning that the network is operating at its maximum capacity without any packet accumulation or loss. The buffer size of 100 packets is insufficient to handle the incoming and outgoing traffic, so some packets are dropped eventually. The packets start to get dropped from the queue at the event of 4500.

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated | Simulation 27:  Arrival Rate (λ) = 125 pkts/sec Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 150 pkts. |

|  |  |
| --- | --- |
| A graph with green and blue lines  Description automatically generated | Close view in the event where the packet get start drops |

In this scenario, the arrival and departure rates are perfectly balanced, indicating that the network operates at full capacity with some packet loss which is less when compared to other scenarios. With a buffer size of 150 packets, the network can effectively manage the incoming and outgoing traffic, but still some packets are dropped eventually at the event of 29500.

**Experiment 2: Analysis of the Simulation Result for Variable Input Rate Simulator**

|  |  |
| --- | --- |
| A graph with a line  Description automatically generated | Simulation:  Arrival Rate (λ) =  70, 200, 130, 120, 70 pkts/sec  Event (%) =  0-10, 10-70, 70-80, 80-90, 90-100  Departure Rate (μ) = 125 pkts/sec Buffer Size (n) = 100 pkts. |

|  |  |
| --- | --- |
| A graph with a line graph  Description automatically generated with medium confidence | A graph with a line  Description automatically generated |
| Plot shows the (10 -70 %) event where  packets started getting dropped due to arrival rate > departure rate & buffer size. | Plot shows the 80% – 100% event where packet drop gets less when compared to (10- 70%) events. |

* 0 - 10% Event Range: Arrival Rate (λ) is less than the Departure Rate (μ) and Buffer Size (n). No packets are dropped. The queue remains relatively empty, and all incoming packets can be accommodated and transmitted without losses.
* 10 - 70% Event Range: Arrival Rate (λ) is greater than the Departure Rate (μ) and Buffer Size (n). Packets start getting dropped. The queue becomes congested due to the higher arrival rate, and some packets are discarded to prevent buffer overflow and packets gets dropped at the event of 100200.
* 70 - 80% Event Range: Arrival Rate (λ) is still greater than the Departure Rate (μ) and Buffer Size (n). Packets are dropped, but not as frequently as in the 10 - 70% event range. The queue is still congested, but the dropping rate is somewhat reduced at the event of 700000.
* 80 - 90% Event Range: Arrival Rate (λ) is less than the Departure Rate (μ) and Buffer Size (n). The arrival rate is less than the departure rate, so packets are not dropped. However, the queue starts to fill up with packets and starts to drop at the event of 860000.
* 90 - 100% Event Range: Arrival Rate (λ) is like 0 - 10% event range, less than the Departure Rate (μ) and Buffer Size (n). No packets are dropped. Like the 0 - 10% event range, the queue remains relatively empty, and all incoming packets are transmitted without losses.
* This observation showcases the dynamic nature of the packet queue's behaviour under varying event rates. The queue adapts to fluctuations in traffic, attempting to accommodate incoming packets while minimizing packet loss.