



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

**INTERNATIONAL JOURNAL OF
INNOVATIVE COMPUTING**

ISSN 2180-4370

Journal Homepage : <https://ijic.utm.my/>

Queue Scheduling Algorithms: A Comparative Analysis

Essam F. Natsheh

Dep. of Informatics Engineering, College of Engineering
AMA International University
Salmabad, Kingdom of Bahrain
Email: dr_natsheh@hotmail.com

Submitted: 12/02/2020. Revised edition: 28/04/2020. Accepted: 29/04/2020. Published online: 20/05/2020
DOI: <https://doi.org/10.11113/ijic.v10n1.257>

Abstract—Congestion control and reduction is paramount in enhancing the performance and speed of communication networks. Despite its efficiency in improving the speed and performance of communication networks, it is imperative to mention that the technique exhibit several issues, such as scheduling algorithms. This paper presents a comparison of different results associated with different scheduling algorithms. Multisource was used to simulate and test different scheduling algorithms, including First In First Out (FIFO), Random Packet Drop, and Last In First Out (LIFO). The comparison was made by analyzing different results, including the average delay vs arrival rate, average buffer utilization vs arrival rate, and packet loss ratio vs arrival rate. According to the results, it is evident that the average buffer utilization and average packet delays increased as the rate of data transmitted through the network increased. There was no significant change noticed in the average buffer utilization in all the three algorithms. However, the packet loss ratio was high in Random Packet Loss algorithm than in both FIFO and LIFO for slower arrival rates below 1000. Similarly, FIFO exhibited a significantly high average packet delay than both LIFO and Random Packet Drop algorithms.

Keywords—Queue scheduling algorithms, first in first out scheduling, last in first out scheduling, random packet drop scheduling

I. INTRODUCTION

One of the techniques used to enhance the performance and speed of communication networks is via control and reduction of congestion. While this technique proves to be effective in enhancing the performance and speed of communication networks, it exhibits several issues. It is important for new packets to be lined up in a manner that enhances performance while reducing congestion [1]. As such,

scheduling algorithms become of the major issues and challenges in the reduction and control of congestion in communication networks. The general unanimity associated with the scheduling strategy entails the selection of the packet that should leave the queues first.

It is crucial to define scheduling before making a comparison of the existing schedule algorithms. In computer science and technology, scheduling refers to the method applied for regulating or controlling the order of the job or process, which is to be performed by the processor. It can also be defined as a tool or mechanism that is used to regulate the execution of various processes that a computer performs [2]. It is essential to note that CPU is the most significant resource available in a computer system that must be programmed before use. Efficient scheduling of the CPU is critical in attaining Multiprogramming. The primary aim of scheduling is to engage the processor at its maximum capacity, thus to avoid or minimize delays. Scheduling keeps the CPU busy at all time through execution of processes and switching to a different process.

It is important to mention that effective scheduling allows the processor to function at its maximum possible capacity to ensure that there is no process or task that waits for longer periods before processing. The whole technique allows the processor to accomplish the whole task in the shortest time possible [3]. Consequently, the processor employs the use of various scheduling algorithms for congestion control and reduction. In turn, it enhances the performance and speed of the computer and communication networks in equal measure. Multi-programing is attained through effective scheduling.

In most cases, the Multi-Layer Queue (MLQ) scheduling accomplishes its role by partitioning the queue in various separate queues. It then assigns a different scheduling to each

queue. In this manner, the technique classifies each process into groups based in various attributes associated with the process [1]. It then assigns then assigns the process to one queue permanently. It is essential to note that this scheduling technique can relocate the processes in different queues dynamically. Consequently, processes that require significant amount of CPU time are relocated to low priority queues while processes that demand less amount CPU time are moved to high priority queues.

II. QUEUE SCHEDULING ALGORITHMS

Scheduling algorithms play a significant role in the performance and speed of communication networks. They are majorly used in packet scheduling in various mobile networks. Apart from the wireless medium's error-prone characteristic, the mobility of nodes presents several challenges, such as packet losses and frequent route changes [4]. It is essential to mention that these issues not only decrease network throughput, but also increases packet delays. In addition, the performance degradation of a network worsens with the increase in traffic load in the network. It is sad that the proposed solutions in an attempt to solve these issues focus only on media access control and routing protocols [1]. However, an understanding of the queueing dynamics exhibits a significant potential to address these challenges than the existing techniques. It is important to examine the effects of various packet-scheduling algorithms in the queues associated with the nodes [5]. This paper presents an analysis and comparison of various packet scheduling algorithms to determine the best algorithms that are capable of enhancing the performance to control and reduce congestion. In our previous works, we optimized scheduling policy of queuing systems using fuzzy reasoning [6] and [7].

Apart from the multi-ho forwarding of packets, multiple roles of nodes as sources, routers and data sinks, and potential recurrent transmissions of control packets because of mobility, Ad hoc networks exhibit various features that may generate exceptional queueing dynamics. The choice of scheduling algorithm or technique to establish the queued packet that should be processed next has a significant effect on the general performance of the network, especially congestion control and reduction. The following scheduling algorithms will be examined and compared; First In First Out (FIFO), Random Packet Drop, and Last In First Out (LIFO). The comparison was made by analyzing different results, including the average delay vs arrival rate, average buffer utilization vs arrival rate, and packet loss ratio vs arrival rate.

All these scheduling algorithms exhibit different scheduling policies. FIFO scheduling policy requires that processes be completed as they arrived at the CPU. It processes the first processes in the queue followed by the next process in the same queue. As such, the first process to be queue is the first processes to be completed [2]. On the other hand, LIFO scheduling algorithms does the exact opposite of FIFO. The scheduling policy of the LIFO demands that the last process in the queue is given priority over other processes, and processed before the first processes in the queue [4]. On the other hand, the scheduling policy associated with the Random Packet Drop scheduling algorithms suggests a different mode of operation.

The scheduling policy requires that in case a new packet arrives into the system and notices a full queue, then the system should randomly select a packet in the queue to be dropped. It is essential to mention that the mode of operation of each scheduling algorithm presents it to various attributes, pitfalls, and upshots in equal measure.

A. Arrival Function

It is important to understand the arrival functions of various scheduling algorithms before performing a comparative analysis on their performance (Fig. 1 and 2). Generally, the arrival functions of FIFO scheduling policy and LIFO scheduling policy are relatively the same. The packets are sent to either the queue or the server depending on the status of the server. If the server is idle, then the arriving packets are sent to the server, otherwise they are sent to the queue. In the queue section, the algorithm check whether the number of packets are more than or equal to the maximum size. In case number of packets are more than the maximum size, the arriving packets lost, otherwise there are queued according to the respective scheduling policies.

B. Average Delay x Arrival rate

According to Fig. 3, it is evident that FIFO scheduling algorithm exhibits the highest average delay vs arrival rate. As indicated by the curve, the three scheduling algorithms exhibit almost the same duration of average delay when the arrival rate is between 200 and 600. The average delay increases exponentially with an increase in the arrival rate of the packets until 600. After the 600 arrival-rate points has been reached, the three scheduling algorithms exhibit different durations of average delay. It is apparent that FIFO exhibits the longest average delay against the arrival rate [5]. The average delay of packets is shortest in Random Drop scheduling algorithm as compared to the LIFO and FIFO. As opposed to FIFO scheduling algorithm, LIFO and Random Drop scheduling algorithms exhibits a reduction in the average delay past the 600 arrival rate points. The difference in the average delay vs the arrival rate in the three algorithms can be attribute to their mode of operation [3]. FIFO exhibits the highest average delay as the arrival rate increases due to its FIFO scheduling policy. It requires that processes be completed as they arrived at the CPU. It processes the first processes in the queue followed by the next process in the same queue. As such, the first process to be queue is the first processes to be completed. On the other hand, the average delay vs arrival rate is shortest in Random Packer Drop because the system should randomly select a packet in the queue to be dropped in case a new packet arrives into the system and notices a full queue.

C. Packet Loss ratio vs Rate of Arrival

As indicated by Fig. 4, there is no significant difference in the packet loss ratio vs arrival rate. The three algorithms exhibit almost similar correlation between packet loss ratio and arrival rate, at least between arrival rates of 0 to 500. After the 500-arrival rate level, Random Packet drop algorithm exhibits

a higher packet loss ratio compared to the other two algorithms (FIFO and LIFO). The trend continues until the 1000 arrival rate mark. After the 1000 mark, both LIFO and FIFO exhibit

the same loss ratio, which is above the loss ratio associated with Random packet drop algorithm [1].

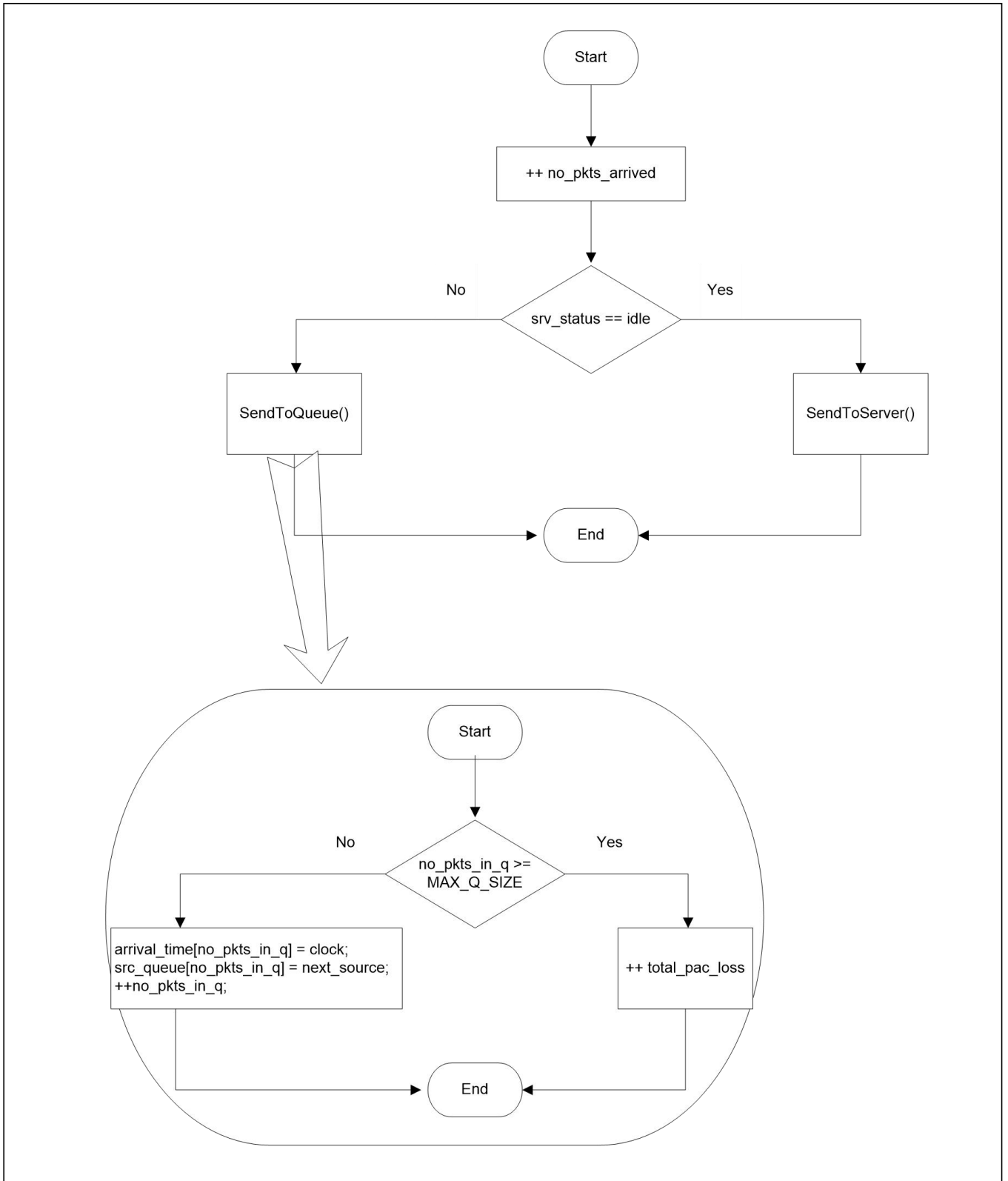


Fig. 1. Arrivals function for LIFO and FIFO

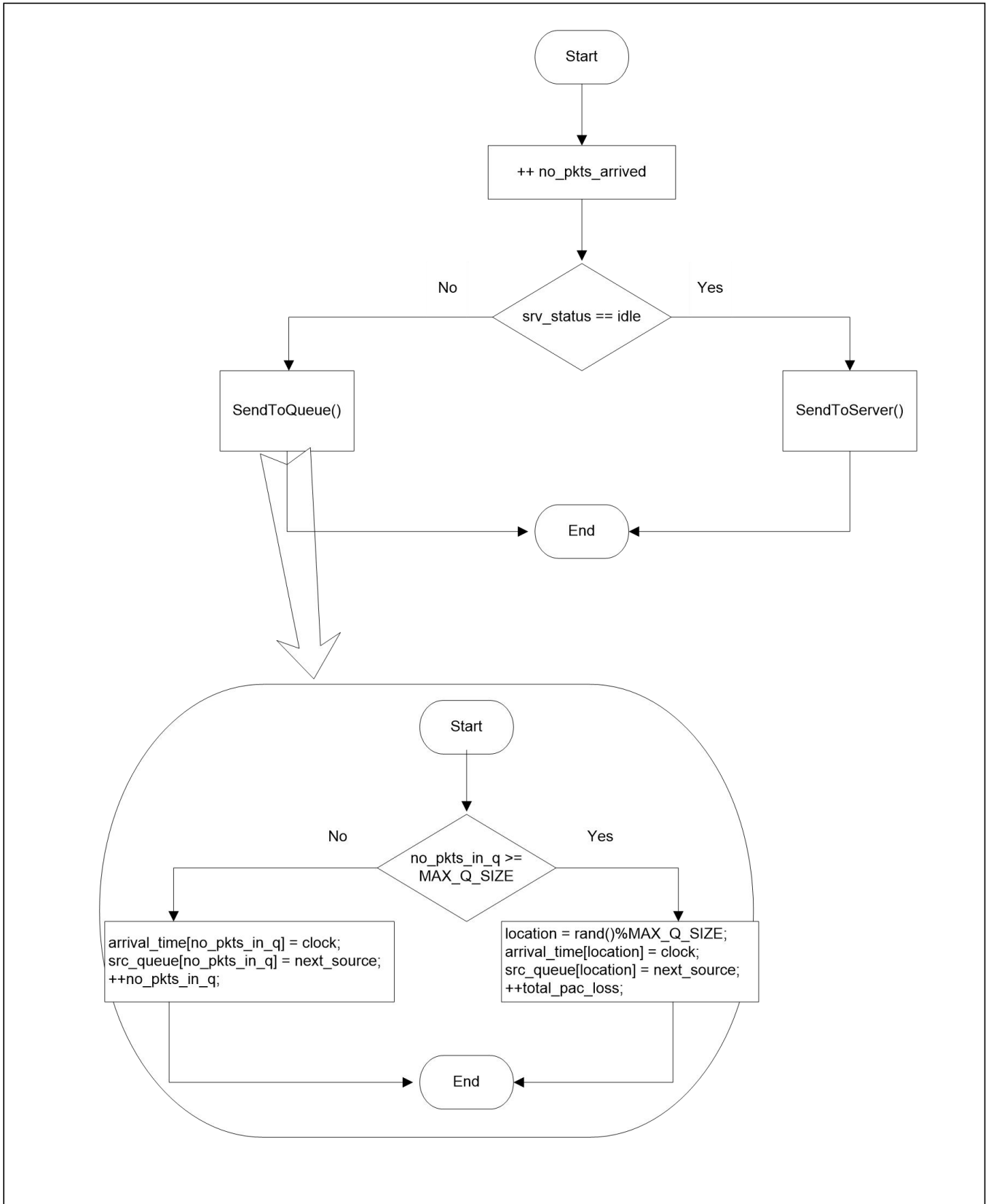


Fig. 2. Random Packet Drop scheduling policy

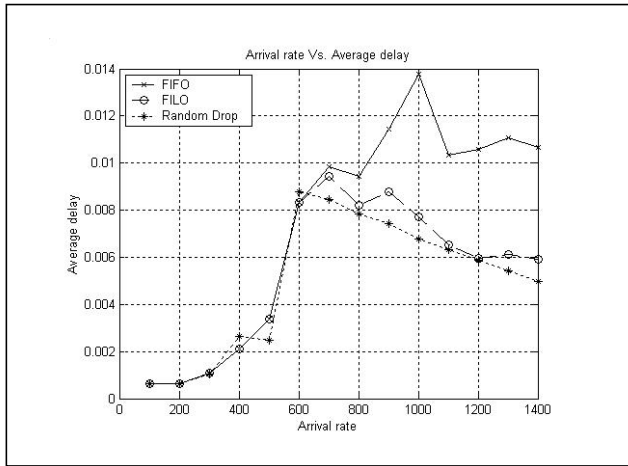


Fig. 3. Arrival rate vs average delay

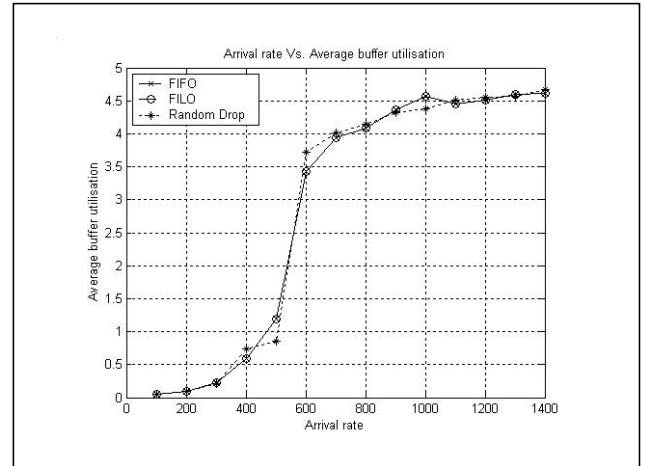


Fig. 5. Arrival rate vs average buffer utilization

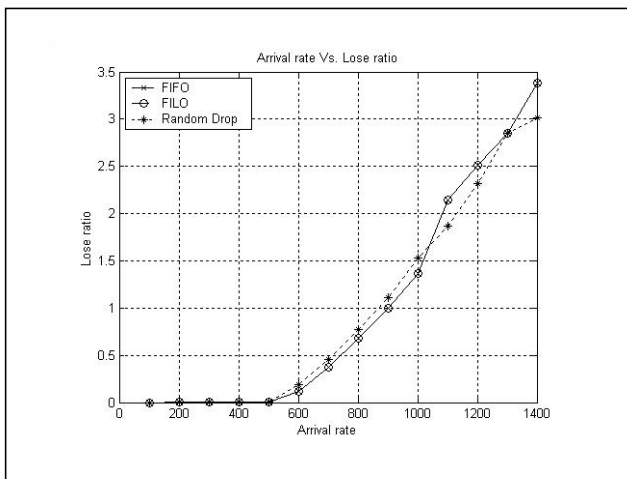


Fig. 4. Arrival rate vs packet loss ratio

D. Average buffer utilization vs arrival rate

According to Fig. 5, it is apparent that there is no significant difference in the correlation between average buffer utilization and arrival rate in all the three algorithms. The average buffer utilization increases linearly as the rate of data transmitted through the network increases [8].

ACKNOWLEDGMENT

The author would like to express his appreciation to College of Engineering at AMA International University Bahrain for supporting this research.

REFERENCES

- [1] C. Annadurai. (2011). Review of Packet Scheduling Algorithms in Mobile Ad Hoc Networks. *International Journal of Computer Applications*, 15(1), 7-10.
- [2] R. Nakkeeran and S. Sulthana. (2017). Performance Evaluation of Downlink Packet Scheduling for Real Time Traffic in LTE System. *International Journal of Information and Communication Technology*, 10(3), 263.
- [3] S. Almakdi, M. Aleisa and M. Alshehri. (2015). Simulation and Performance Evaluation of CPU Scheduling Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 1-6.
- [4] S. Radhakrishnan, S. Neduncheliyan and K. Thyagarajan. (2016). A Review of Downlink Packet Scheduling Algorithms for Real Time Traffic in LTE-Advanced Networks. *Indian Journal of Science and Technology*, 9(4).
- [5] S. Rohil. (2018). A Review on CPU Scheduling Algorithms in Cloud Environment. *International Journal for Research in Applied Science and Engineering Technology*, 6, 792-798.
- [6] E. Natsheh, A. B. Jantan, S. Khatun and S. Subramaniam. (2007). Intelligent Reasoning Approach for Active Queue Management in Wireless Ad-hoc Networks. *International Journal of Business Data Communications and Networking (IJBDCN)*, 3(1), 16-33.
- [7] E. Natsheh and K. Buragga. (2010). Optimizing Scheduling Policy of Queuing Systems in Heterogeneous Environments using Fuzzy Reasoning. *International Journal of Computer Science and Network Security (IJCSNS)*, 10(4), 206-213.
- [8] S. Suranauwarat. (2007). A CPU Scheduling Algorithm Simulator. *WI 37th ASEE/IEEE Frontiers in Education Conference*. October 10-13, 2007, Milwaukee, USA.

