Journal of Optimization of Soft Computing (JOSC)
Vol. 3, Issue 1, pp: (41-52), Summer-2025
Journal homepage: https://sanad.iau.ir/journal/josc

Paper Type: (Research paper)

Implementation and comparison of active queue management algorithms in traditional and SDN networks

Khoshnam Salimi Beni[1], Mohammadreza Soltanaghaei [1*] and Rasool Sadeghi[2]

[1]*Institute of Artificial Intelligence and Social and Advanced Technologies, Isf.C., Islamic Azad University, Isfahan, Iran.*

[2]*Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran*

Article Info

Article History:
Received: 2025/07/28
Revised: 2025/08/30
Accepted: 2025/09/13

DOI: josc.2025.202507281213297

Keywords:
Network, SDN, resource allocation, queuing mechanisms

**Corresponding Author's Email Address: soltan@iau.ac.ir*

Abstract

In the past decade, networks have experienced significant improvements in scale and data transfer rates, and network traffic rates will soon increase dramatically. Network management and traffic control play key roles in real-time data transmission (such as video conferencing, high-bandwidth streams, video calls, etc.) and data transmission in the Internet of Things (IoT). Although technologies such as SSD storage and virtualization are very effective in meeting network traffic needs. Future networks will require centralized management, easy upgradeability, application optimization, efficient resource allocation, and dynamic routing. To meet these requirements, the benefits of software-defined networking (SDN) must be used. By separating the control part from the data part, SDN will lead to scalability, flexibility and centralized management of the network. With excessive demands on limited network resources, it is inevitable to create long queues of information packets in intermediate routers, and the use of active queue management (AQM) algorithms of TCP/IP network in order to make more use of available bandwidth and reduce Transmission delay is necessary. In this article, we examine some of the most important active queue management algorithms including PFIFO_fast, ARED, CoDel, FQ-CoDel and PIE in traditional and SDN networks. The results of the simulation show that the use of AQM algorithms in the SDN network reduces the average delay and packet loss rate and increases the network efficiency.

1. Introduction

Configuring and implementing different scenarios in traditional networks not only has problems in the field of managing these networks and wasting time, but also has its own errors and problems in the field of extensibility. Therefore, it is necessary to change the network architecture and use other structures such as software-defined networks (SDN)[1]. SDN is one of the new network architectures that separates the data part from the control part to improve the use of network resources, reduce operational costs and provide network innovation and evolution. However, the main challenge in SDN is to provide high quality services and resource allocation in these programmable networks. Proper allocation of resources improves network performance and reduces overall network costs. In this regard, various techniques are used to allocate resources in SDN in order to increase network efficiency, one of which is the optimal allocation of resources to each task in the network[2]. Various parameters are defined in these techniques for resource allocation in SDN. Overview the controller in SDN can easily collect data from available network resources and basically allocate resources to different services through the OpenFlow protocol[3]. Various techniques and methods have been used to improve network resource allocation in SDN. SDN can be used in various technologies such as virtual networks (VN), data centers (DC), cloud environment, 5G and wireless networks, and can also be used in combination to improve network performance[4]. In SDN networks, two important resources, the bandwidth capacity of the switch to the controller and the capacity of the flow table, which mutually influence each other, must be carefully analyzed[5]. For example, the acceleration of the incoming flow to a switch can greatly intensify the message exchange between the switch and the controller, which causes more bandwidth consumption[6]. In order to optimally transmit data in the network, some challenges such as congestion, delay and packet loss must be considered. Some papers focus on the channel congestion problem and suggest the use of queue management algorithms. These algorithms are categorized into Active Queue Management (AQM) or Passive Queue Management (PQM) such as Drop-Tail[7] depending on the congestion control mechanism. Some of the famous AQM algorithms are RED[8], ARED[8], PIE[9], CoDel[10], FQ-CoDel[11] and PFIFO_fast[12], which are used in various papers to address these challenges. The queue management system controls the size of the communication channel queue by enabling or disabling queue management. One passive queue management algorithm is Drop-Tail, which drops packets when the queue is full, but in active systems, such as Random Early Detection (RED), network packets are dropped before the queue becomes saturated. In addition, other queue management algorithms such as CoDel have proposed Adaptive Random Early Detection (ARED), Packet-First-In-First-Out (PFIFO), to solve the congestion problem.

SDN is one of the new network architectures in which the information control part is separated from the data part. In traditional networks, routers and network switches, data transfer and information control operations are performed together. In SDN architecture, the control part is separated from the switch and router hardware and is performed by software at a higher layer. Therefore, the speed, flexibility, scalability, availability and reliability of the network are improved. Researchers can centralize and integrate network management by creating programming interfaces. Another advantage of using SDN is network and hardware reconfiguration without the need for the involvement of hardware manufacturers. In traditional networks, they must use the technology and architecture provided by hardware manufacturers, and network development is not possible, but in SDN networks, according to the needs, the network can be localized[13].

OpenFlow is a key protocol in SDN that enables centralized control and programmability of network devices. In an SDN architecture, OpenFlow decouples the control plane from the data plane, allowing a centralized controller to manage and direct network traffic dynamically. This separation facilitates the implementation of network policies and configurations through a logically centralized controller, leading to more efficient network management and flexibility. OpenFlow operates by defining a set of communication messages between the SDN controller and network devices, such as switches and routers, allowing the controller to instruct these devices on how to forward, modify, or block packets based on the network's current state and requirements. The protocol enhances network agility, scalability, and programmability, making it a fundamental component in the evolution of modern networking architectures[14].

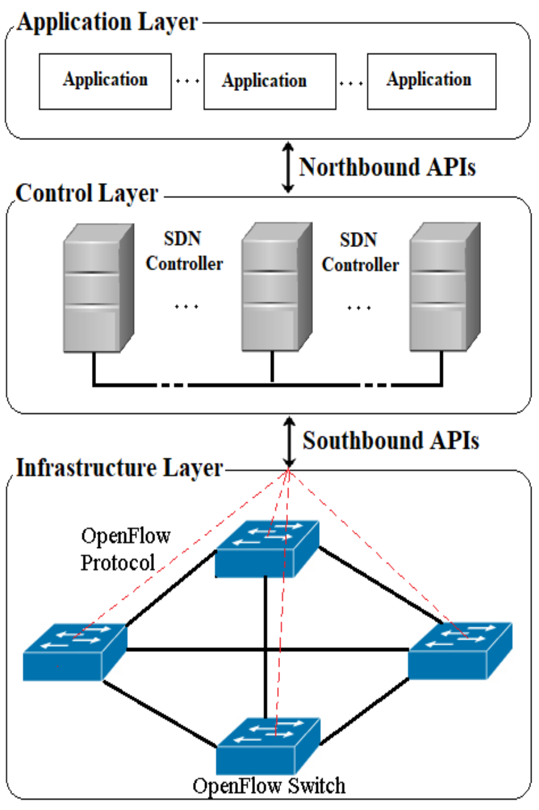


Figure (1). SDN Architecture

In this article, we examine the use of queue management algorithms in congestion control. For this purpose, the most important queue management algorithms including PFIFO_fast, ARED, CoDel, FQ-CoDel and PIE in traditional networks and Software-Defined Networks have been investigated separately. One of the most important factors affecting network efficiency is the average delay and packet loss rate that have been evaluated. In the second part, software-defined networks and basic concepts will be examined, and the studies and works done on queue management algorithms will be examined. In the third part, some of the most important queue management algorithms have been implemented in the SDN network using the NS3 simulator. In the fourth part, the results of the implementation of queue management algorithms in traditional and SDN networks will be analyzed and evaluated, and finally, the fifth part will include the conclusion.

2. Related works

In this section, we examine the research that has been done on queue management algorithms. Optimal allocation of resources in SDN plays a key role in improving performance and is very challenging. To achieve this goal, many solutions have been proposed in the existing research. Some resources, such as the capacity of the communication link between the switch and the controller, the rate of messages sent and received from the switch to the controller, and vice versa, have limitations. Therefore, bandwidth management is very important for resource sharing in SDN [15]. Various techniques and methods have been proposed for optimal bandwidth allocation in SDN[16], which we have discussed in this section.

Some research investigates resource consumption in SDN controller and switches using queuing mechanisms. Packet loss may occur simultaneously while transmitting traffic with the same queue priority. J. Hao and his colleagues in [17] presented a flow-level bandwidth provisioning algorithm (FBP) to deal with the switch scheduling problem using a fair queuing algorithm. This algorithm schedules multiple fair queues on OpenFlow switches to separate flows and allocate bandwidth between flows on a shared link. However, this paper has disadvantages such as time complexity, large hardware, unreliability of service guarantee, and inefficient processing of variable-length packets. Some of these problems were solved by traffic classification and queue prioritization, which was studied by H. Cui and his colleagues in the article [18]. In this article, the queuing mechanism is used to classify traffic, collect information about the network status, and determine the optimal route for allocating network resources to different services. The purpose of the proposed plan is to guarantee QoS for different services and balance the load on the communication link and prevent time wastage and congestion in the network. This mechanism divides the core network's proposed queue into smaller virtual subnets and allocates resources to each service. Multiple queues with different priorities can be configured on a switch port. However, when a new flow arrives on a port, other lower-priority flows in the queues may experience delays and jitter. However, dynamic queue mapping that can improve resource allocation in the network is not studied in this paper.

Connections that have a common destination in the network share their communication links to use these links. However, without providing a solution to protect and isolate services, connections with high data transfer rates will send more traffic to the core network than others. Therefore, packet loss occurs more often in low-rate connections. To address this problem J. Guo. and colleagues in [19] proposed an application-layer fair bandwidth allocation (FBA) protocol called Falloc to distribute network resources at the virtual machine (VM) level in IaaS data centers. Falloc assigns a base bandwidth and a weight to each virtual machine. Virtual machines with less required bandwidth than the original bandwidth share the remaining bandwidth among all machines in proportion to the weight. Therefore, fairness can be ensured by balancing the bandwidth allocated to VMs and the shared bandwidth between virtual machines. However, due to competition between other VMs, Falloc cannot guarantee bandwidth for all VMs.

Another fair bandwidth allocation scheme in the application layer named UFalloc is proposed in the article [20] to achieve fairness in bandwidth allocation between virtual machines using the max-min algorithm [21]. UFalloc not only limits the bandwidth of each flow on OpenFlow switches to guarantee performance, but also shares bandwidth resources across dense switches and links. UFalloc uses a factor called relaxation-fairness to maintain a certain degree of fairness for bandwidth allocation. As shown in the results of this paper, UFalloc can reduce the application utilization switching degree by 5.9% to 10.9% compared to the traditional TCP rate control mechanism and max-min fair allocation algorithm. However, the computational overhead is one of the drawbacks of this method.

Heuristic methods can be used to solve problems that are not guaranteed to be optimal, logical, or complete, but can be sufficient to arrive at an approximate solution. In cases where it is impractical or impossible to identify an optimal solution, heuristic strategies can be used to speed up the process of finding a satisfactory solution. Using heuristic algorithms, KT. Bagci and his colleagues in [22] proposed a heuristic model based on a packet-based shortest path (GCSP) for fair allocation of resources among a group of requests with the same service level. This method is close to the optimal solution and uses a divide-and-conquer strategy in networks that are divided into smaller subnets. So many service requests can be handled by processing groups of service requests in minimum time. Another algorithm based on near-optimal heuristic methods was presented by W. Aljoby and his colleagues in the paper [23] to share bandwidth between several active applications in SDN. This algorithm is obtained from stream processing in programs and formulation of bandwidth allocation between streams belonging to these streams. Although this method can be used in a variety of platforms, including parallel and pipelined network flows, however, balancing bandwidth between multiple broadcast applications with different performance and bandwidth-optimizing communication overheads is a challenge. This article is important.

A. Marin and colleagues in [24] analyzed AQM techniques for bandwidth sharing in TCP and UDP traffic and analyzed the performance of these techniques in different scenarios using mean field methods. Because TCP and UDP streams exist on the same channel, and because TCP's congestion control mechanism uses more resources, packets for UDP-based applications must be queued in buffers. Therefore, UDP-based applications will experience unfavorable latency. To deal with this challenge, AQM mechanisms are used with the help of congestion control mechanism to avoid congestion in bottleneck links and make optimal use of available bandwidth. In this study, their proposed method is compared with RED. Another optimal model of Internet congestion control using AQM is proposed by C. Han and his colleagues in [25]. In this paper, a state monitoring system is used to collect system state information. This observer can estimate the window size in the real network and estimate the queue length obtained by measuring the output of the system. Another efficient AQM algorithm is proposed by L. Chrost and A. J. T. S. Chydzinski in the paper [26]. This algorithm keeps the queue size short and stable to reduce packet loss and high throughput. The proposed algorithm reduces the energy consumption in routers by reducing the complexity of calculations. Some AQM schemes such as RED calculate the probability of dropping packets using the average queue size. In heavy traffic, increasing the frequency of crossing the maximum threshold value will lead to frequent dropping of packets. To address this problem, S. Patel and S. J. T. S. Bhatnagar in [27] proposed an adaptive queue management mechanism using information obtained from the average queue size and the rate of change of the queue size. Therefore, using the rate of change in queue size as an additional parameter leads to an increase in system efficiency in terms of average queue size, throughput and queue delay compared to other AQM algorithms such as RED. The problem of bandwidth consumption in wireless networks of TCP streams is investigated by K. O. Okokpujie and his colleagues in the article [28]. Using AQM algorithms in this paper, two adaptive TCP strategies are proposed for queue management by implementing feedback control techniques in AQM. The comparison result states that these models have better performance than PI and RED proportional integral controllers.

In large networks, simultaneous communication between network equipment leads to increased channel congestion. One of the important challenges in these networks is to maintain fairness between TCP-based packets that are sent on a congested channel. Some papers suggested using Drop-Tail for fair channel allocation to each stream. But dropping the packet in the Drop-Tail method reduces the flow throughput. To overcome this challenge, M. M. Hamdi and his colleagues in [29] proposed the use of AQM to solve the problem of packet loss and delay. In this article, Drop-Tail is compared with different active queue management methods such as PFIFO, RED, ARED and CoDel. The comparison results show that CoDel and RED have higher throughput with minimum delay. Also, ARED performs better than RED. In addition, the use of RED algorithms reduces the congestion problem, but it does not guarantee network efficiency and QoS due to the predefined and fixed parameters in RED algorithms. To address this problem, A. F. AL-Allaf and A. A Jabbar in [30] proposed an improved adaptive RED algorithm using reconfigurable policy for multimedia traffic. In this approach, the maximum drop probability (maxp) in RED is replaced by another parameter obtained from the network traffic load. According to this policy, the average queue size and queue delay time are reduced without increasing the packet drop rate or reducing the link utilization.

In order to control congestion and improve network performance, D. Kumhar et al in [31] proposed an AQM method based on the RED algorithm named QRED random early detection (QRED). Compared with RED, the simulation results showed that QRED performs better in terms of end-to-end delay, packet loss, packet delivery conditions, and jitter. Although AQM is considered as a solution for congestion control, the selection of accurate parameters for AQM methods is an important problem in inter-domain structures due to the dynamics of IP networks. To address this challenge, C. A. Gomez and his colleagues in [32] proposed an architecture called (FIAQM) to adjust AQM parameters dynamically in a multi-domain network. In this method, artificial neural network is used to check and predict congestion. Therefore, the performance of inter-domain communication is improved by reducing link congestion.

3. Appling of AQM algorithms in SDN architecture

In this section, we apply some of the most important queue management algorithms including PFIFO_fast, ARED, CoDel, FQ-CoDel and PIE first in the traditional network and then in the SDN network. In the traditional network, according to “figure 2”, node1 is connected to node2 through a switch, and also in “figure 3”, node1 is connected to node2 through a switch equipped with SDN technology. The required parameters in the simulation are determined according to “table 1”.

Table 1. Parameters required in the simulation

Parameters	Value
Delay	0.1 ms
Datarate	20-100 mb/s
Queue capacity	1000 packet
Similation time	60 s
Packet size	1024 bit

Different metrics exist in the network to assess and monitor the performance and reliability of computer networks. These metrics provide a significant understanding of various dimensions of network performance, encompassing elements such as data transfer speed, bandwidth utilization, latency, jitter, packet loss, and other fundamental indicators of network performance. Through meticulous monitoring and analysis of these network metrics, network administrators can identify existing constraints, diagnose issues, and improve network configurations to enhance network performance, reduce downtime, and ensure a better end-user experience. In this article, three metrics, namely average delay, average jitter, and packet loss ratio, have been used for network evaluation[33].

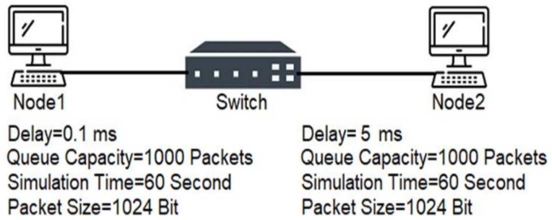


Figure 2. Scenario in Traditional network

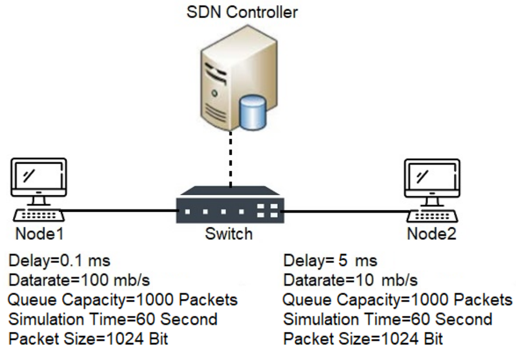


Figure 3. Scenario in SDN Architecture

The structured methodology for evaluating and comparing Active Queue Management (AQM) algorithms in both traditional and Software-Defined Networking (SDN) architectures is shown in Figure 4. The process begins by defining key simulation parameters such as delay, data rate, queue capacity, and packet size, which establishes a consistent baseline for testing. The critical branching point involves selecting either a traditional network or an SDN-based environment, allowing for a direct comparison of how the centralized control and programmability of SDN impact network performance. Each network type then undergoes systematic testing with five prominent AQM algorithms (PFIFO_fast, ARED, CoDel, FQ-CoDel, and PIE) during a 60-second simulation, after which three crucial performance metrics—average delay, average jitter, and packet loss ratio—are collected and analyzed. This comprehensive approach enables researchers to quantitatively assess the effectiveness of each algorithm in both conventional and modern SDN environments, ultimately providing insights into how SDN's centralized control can optimize traffic management and improve overall network quality of service.

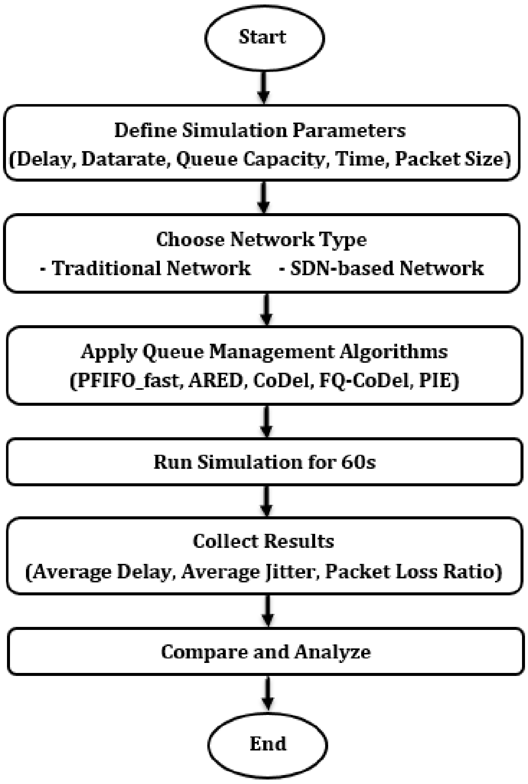


Figure 4. The structured methodology of the proposed algorithm

4. Evaluation and comparison of results

In this section, we implement some of the most important queue management algorithms including PFIFO_fast, ARED, CoDel, FQ-CoDel and PIE first in the traditional network and then in the SDN network. For simulation, the NS3 simulator[34], which is one of the most powerful network simulators, has been used.

A. Evaluation Based on Mean Delay

The metric known as mean delay, or commonly referred to as "average delay," serves as a vital indicator of network performance. It measures the average time it takes for data packets or information to traverse from a source to a destination within the network, typically expressed in milliseconds. Formula (2) is employed to calculate the mean delay, providing a quantitative insight into the efficiency and responsiveness of the network's data transmission.

$$\overline{delay} = \frac{delaySum}{rxPackets} \qquad (2)$$

In this formula, "delaySum" represents the cumulative sum of all end-to-end delays incurred by every packet received within a specific flow. Meanwhile, "rxPackets" denotes the total count of packets received for that particular flow. This information is crucial for assessing the overall performance and efficiency of the network, offering insights into the latency experienced by the transmitted data and the volume of packets successfully received[35].

In Figure (4), the diagram depicts the average delay across varying data rates within the range of 20 to 100 Mbps in a traditional network. The visualization provides a comparative analysis of how different AQM algorithms, including PFIFO_fast, ARED, CoDel, FQ-CoDel, and PIE, influence the average delay for data transmissions. This representation is instrumental in understanding the performance of these AQM algorithms across a spectrum of data rates in a traditional network environment. This visualization provides a comparative analysis of the average delay experienced by data packets as they traverse the network under different AQM strategies. The graph serves as a valuable tool for understanding how these AQM algorithms impact the latency of data transmission in a traditional network setting.

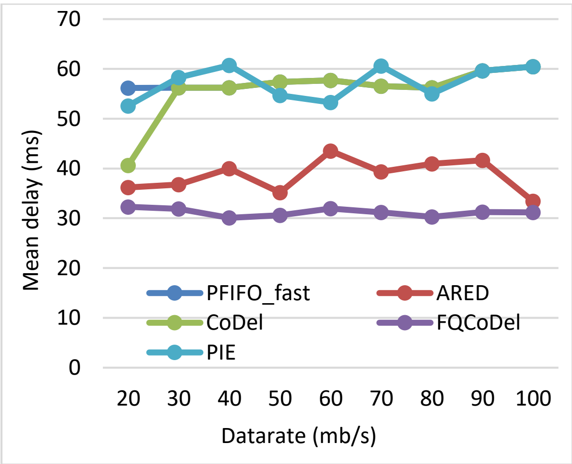


Figure 5. Average delay diagram in traditional network

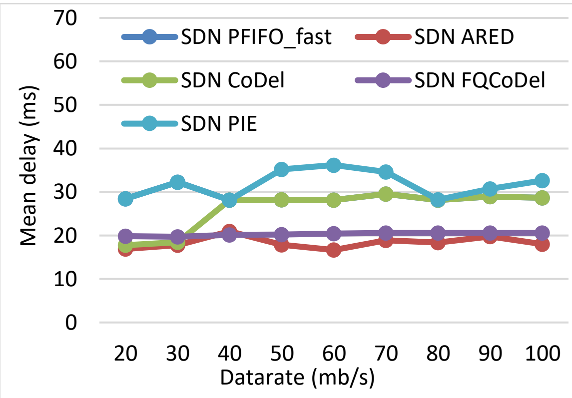
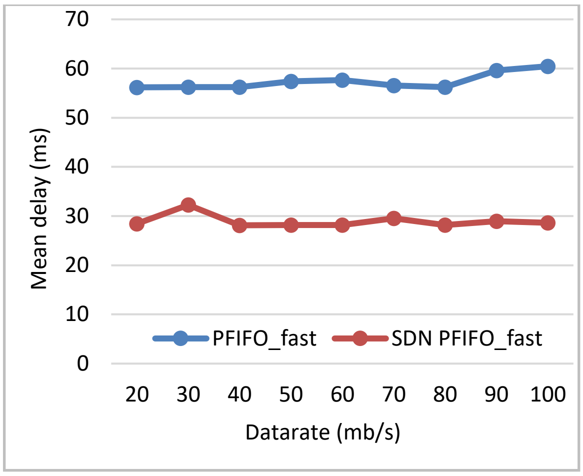


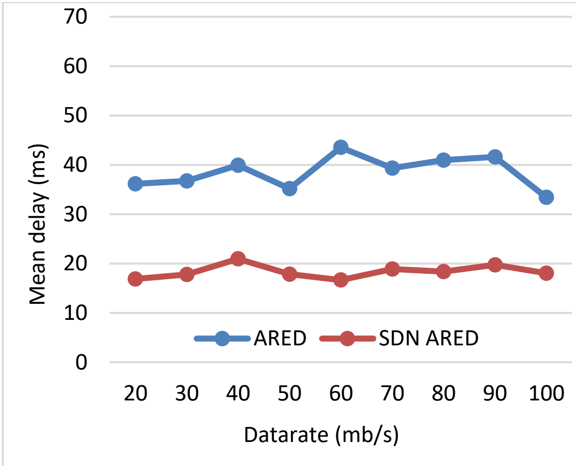
Figure 6. Average delay diagram in SDN network

In Figure (5), the diagram illustrates the average delay across a range of data rates from 20 to 100 Mbps within an SDN architecture. This visualization offers a comparative analysis of the impact of various AQM algorithms, including PFIFO_fast, ARED, CoDel, FQ-CoDel, and PIE, on the average delay experienced by data packets during their traversal in the network. Providing insights into how these AQM algorithms influence latency, this graph serves as a valuable tool for understanding the performance of data transmissions within the context of an SDN environment.

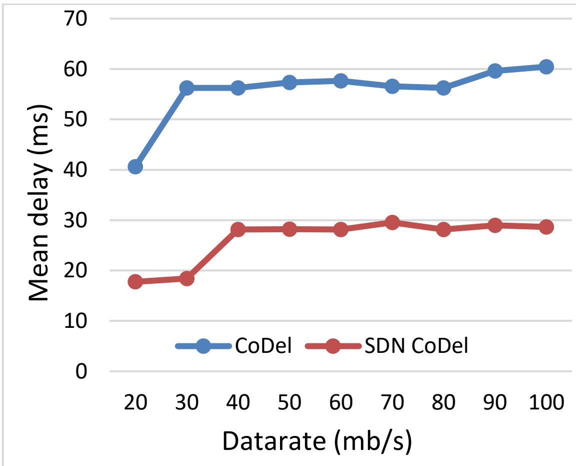
In SDN architecture, implemented AQM algorithms excel at early congestion detection through the monitoring of queue lengths or patterns of packet drops. Upon congestion detection, proactive measures are taken, selectively dropping or marking packets before the queue reaches excessive congestion levels. This preemptive action prevents the network from reaching a critical state, effectively reducing the mean delay for packets. Furthermore, these algorithms support traffic prioritization, offering preferential treatment to specific traffic types (e.g., voice-over IP) or ensuring superior service for time-sensitive applications. This prioritization significantly contributes to reducing the mean delay for critical data, enhancing overall network responsiveness. SDN-based AQM algorithms prove to be a versatile solution, extending beyond mere maintenance of low and consistent delay to actively minimizing overall delay. This dual focus on delay management positions SDN-based AQM algorithms as valuable tools for optimizing network performance. The selection of an appropriate AQM policy within the SDN framework, tailored to the unique network requirements, holds the potential for significant improvements in mean delay. The adaptability to network specifics, coupled with the overarching goal of delay reduction, underscores the potential of SDN-based AQM algorithms to enhance the efficiency of network traffic management. The dynamic adjustment of parameters based on network conditions, facilitated by their implementation in an SDN environment and the centralized aggregation of required information in the SDN controller, allows SDN-based AQM algorithms to fine-tune their settings in response to changing network conditions. The comparative results indicate a significant reduction in mean delay for each of these algorithms. Figure (6) illustrates the comparison chart of each of these algorithms in both traditional and SDN networks.



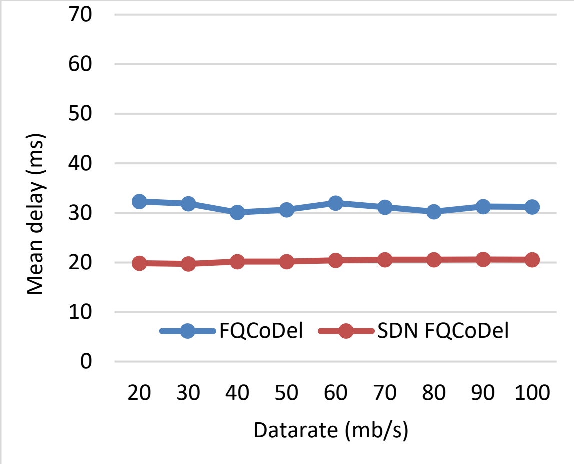
a) Average Mean delay in PFIFO-fast and SDN PFIFO-fast



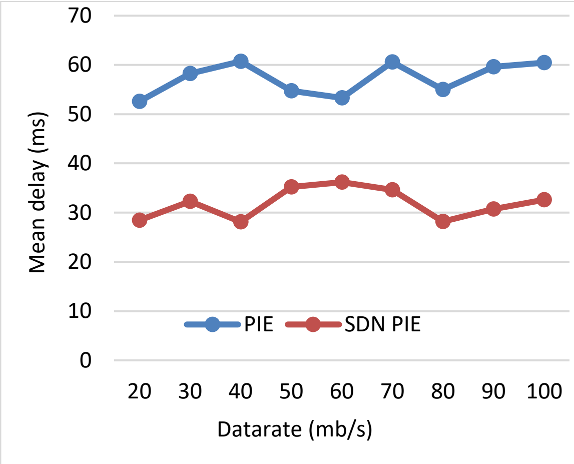
b) Average Mean delay in ARED and SDN ARED



c) Average Mean delay in CoDel and SDN CoDel



d) Average Mean delay in FQCoDel and SDN FQCoDel



e) Average Mean delay in PIE and SDN PIE

Figure 7. Average delay diagram in traditional network and SDN architecture for a) PFIFO-fast b) ARED c) CoDel d) FQ-CoDel e) PIE

B. Evaluation Based on Packet loss Ratio

Packet loss ratio, commonly known as "packet loss," serves as a networking metric gauging the proportion of data packets that do not successfully reach their intended destination or are discarded during transmission across a network. There are various factors contributing to packet loss, such as network congestion, hardware malfunctions, software glitches, or the deliberate discarding of packets by network devices. Mitigating packet loss stands as a primary objective in the realm of network management and optimization, necessitating actions like the enhancement of network infrastructure, implementation of Quality of Service (QoS) mechanisms, or the deployment of error correction techniques to bolster the dependability of data transmission. Formula (4) is provided to articulate the calculation of the Packet loss ratio.

$$q = \frac{lostPackets}{rxPackets + lostPackets} \quad (4)$$

Where the lostPackets variable represents the total count of packets assumed to be lost, meaning those that were transmitted but have not been reported as received or forwarded within an extended timeframe. By default, packets not acknowledged within a duration exceeding 10 seconds are considered lost, though this threshold is adjustable during runtime. On the other hand, the rxPackets parameter denotes the overall number of received packets for the specific flow[35].

In Figure (7), the graphic illustrates the packet loss ratio across diverse data rates ranging from 20 to 100 Mbps within a traditional network. The diagram offers a comparative examination of the impact of various Active Queue Management (AQM) algorithms, namely PFIFO_fast, ARED, CoDel, FQ-CoDel, and PIE, on the Packet Loss Ratio during data transmissions. This visualization proves crucial in comprehending the efficacy of these AQM algorithms across a spectrum of data rates in a conventional network setting. It facilitates a side-by-side comparison of the Packet Loss Ratio encountered by data packets as they navigate the network under distinct AQM strategies. The graph acts as a valuable instrument for gaining insights into how these AQM algorithms influence the latency of data transmission within a traditional network environment.

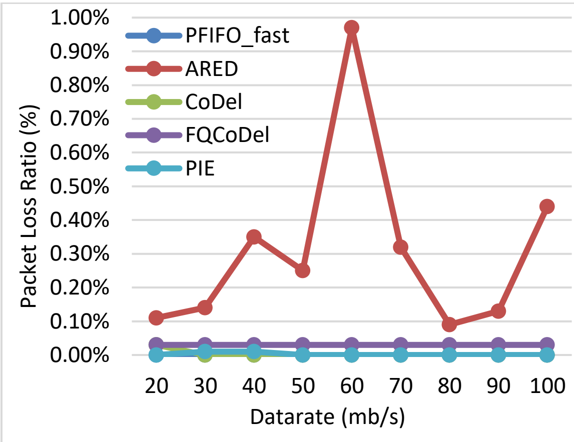


Figure 8. Packet loss ratio diagram in traditional network

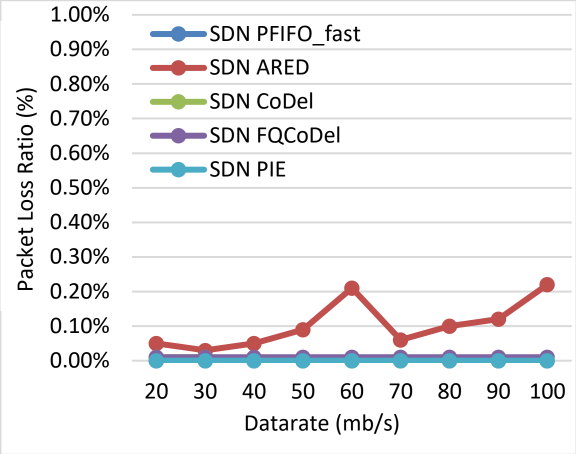
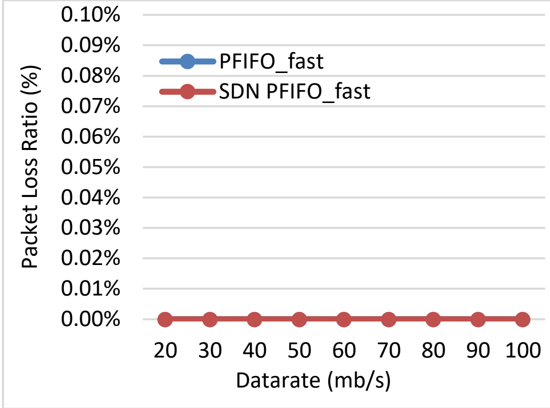


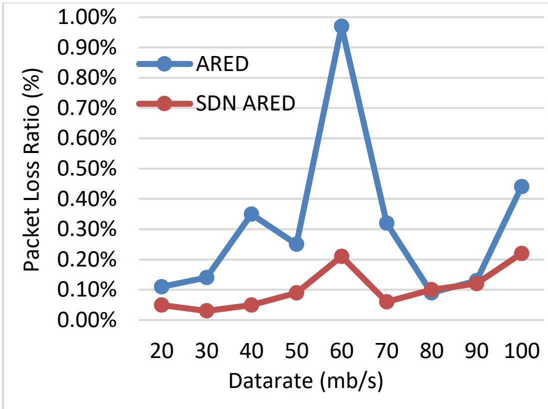
Figure 9. Packet loss ratio diagram in SDN network

In Figure (8), the diagram showcases the Packet Loss Ratio across a spectrum of data rates, spanning from 20 to 100 Mbps within an SDN architecture. This visual representation facilitates a comparative assessment of the influence of different Active Queue Management (AQM) algorithms, such as PFIFO_fast, ARED, CoDel, FQ-CoDel, and PIE, on the Packet Loss Ratio encountered by data packets as they traverse the network. Offering insights into the impact of these AQM algorithms on latency, this graph acts as a valuable resource for comprehending the performance of data transmissions within the framework of an SDN environment.

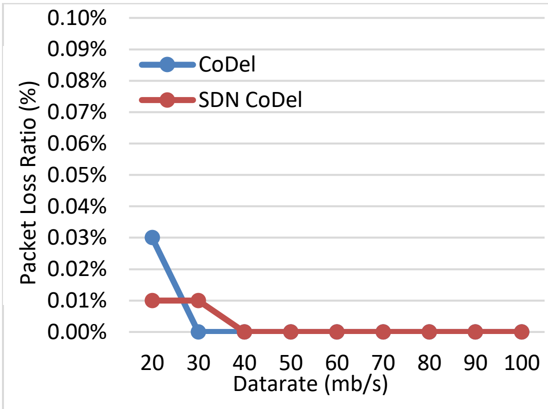
As outlined in Section A, SDN-based algorithms enable traffic differentiation and prioritization, ensuring that critical or time-sensitive data, such as voice or video packets, receives preferential treatment. This prioritization minimizes the risk of losing crucial packets, leading to an overall reduction in packet loss. These algorithms adeptly manage the network queue, exerting a significant influence on mitigating packet loss. By achieving low and consistent queue delays while proactively avoiding congestion-related losses, this algorithm emerges as a pivotal factor in the reduction of packet loss. The comparative findings reveal a notable decrease in the packet loss ratio across all of these algorithms. Figure (9) visually represents the comparative analysis of each algorithm in both traditional and SDN networks.



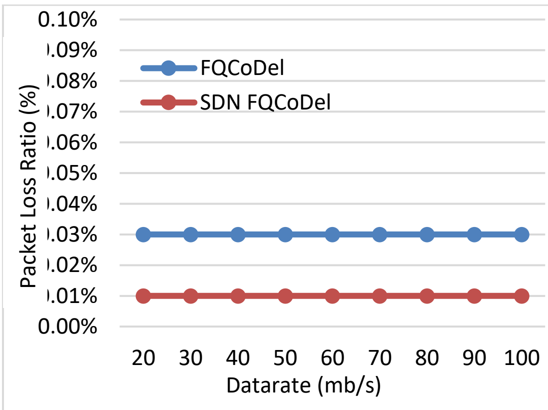
a) Packet loss ratio in PFIFO-fast and SDN PFIFO-fast



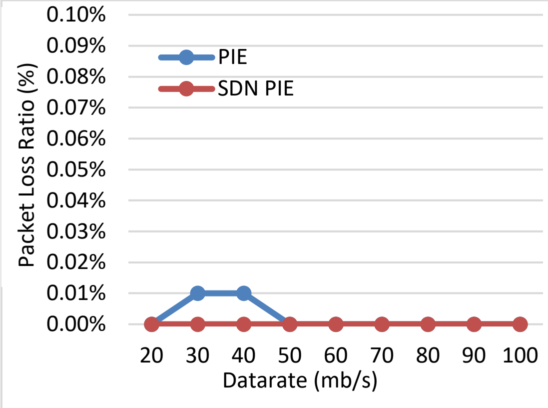
b) Packet loss ratio in ARED and SDN ARED



c) Packet loss ratio in CoDel and SDN CoDel



d) Packet loss ratio in FQ- CoDel and SDN FQ- CoDel



e) Packet loss ratio in PIE and SDN PIE

Figure 10. Packet loss ratio diagram in traditional network and SDN architecture for a) PFIFO-fast b) ARED c) CoDel d) FQ-CoDel e) PIE

According to the scenario in “figure 2” and “figure 3”, the bandwidth and transmission delay between Node1 and the switch are considered to be 100 mb/s and 0.1 ms. Also, the bandwidth and transmission delay between Node2 and the switch are 10 mb/s and 5 ms for traditional and SDN networks. To generate congestion and get better simulation results, the bandwidth between Node2 and the switch varies within the range of 20 to 100 Mbps.The simulation time is 60 seconds and the size of each packet is 1024 bits. Based on the simulation results in the graphs of “figure 4” to “figure 9”, in traditional networks, the FQ-CoDel algorithm has a lower average delay than other algorithms, and in SDN networks, the ARED algorithm has a lower average delay than other algorithms. Also, in traditional networks, the packet loss rate in PIE, CoDel and PFIFO_fast algorithms has the lowest value compared to other algorithms, and for SDN networks, the packet loss rate in PIE and PFIFO_fast algorithms is zero and it has the lowest value compared to other algorithms. The comparison results of each of the aforementioned algorithms in traditional and SDN networks indicate that the average delay and packet loss rate in SDN networks have been significantly reduced.

5. Conclusion

In this article, using SDN technology, we implemented active queue management (AQM) algorithms of TCP/IP network in SDN environment. This caused more utilization of the available bandwidth and reduced transmission delay. In this article, we examined some of the most important active queue management algorithms including PFIFO_fast, ARED, CoDel, FQ-CoDel and PIE in traditional and SDN networks. The results of the simulation show that the use of AQM algorithms in the SDN network reduces the average delay and packet loss ratio and increases the network efficiency.

References

[1] Al-Darrab, A., I. Al-Darrab, and A. Rushdi, Software-Defined Networking load distribution technique for an internet service provider. Journal of Network and Computer Applications, 2020. 155: p. 102547, https://doi.org/10.1016/j.jnca.2020.102547.

[2] Al-Tarawneh, L. and O.A. Saraereh, An optimal method for resource allocation in SDN optical networks. Optical Fiber Technology, 2022. 74: p. 103120, https://doi.org/10.1016/j.yofte.2022.103120.

[3] Prabha, C., A. Goel, and J. Singh. A survey on sdn controller evolution: A brief review. in 2022 7th International Conference on Communication and Electronics Systems (ICCES). 2022. IEEE, https://doi.org/10.1109/ICCES54183.2022.9835810.

[4] Salami, Y., et al., A Novel Approach for Intrusion Detection System in IoT Using Correlation-Based Hybrid Feature Selection and Harris Hawk Optimization Algorithm. Journal of Optimization in Soft Computing, 2024: p. 54 - 63, https://doi.org/10.82553/josc.2024.140308141189548.

[5] Banitalebi Dehkordi , A., An optimal approach to detect anomalies in intrusion detection systems. Journal of Optimization in Soft Computing, 2024: p. 54 - 63, https://doi.org/10.82553/josc.2024.140212131104279.

[6] Aldabbas, H., Efficient bandwidth allocation in SDN-based peer-to-peer data streaming using machine learning algorithm. The Journal of Supercomputing, 2023. 79(6): p. 6802-6824, http://dx.doi.org/ 10.1007/s11227-022-04929-y.

[7] Hamdi, M.M., et al. A review on queue management algorithms in large networks. in IOP Conference Series: Materials Science and Engineering. 2021. IOP Publishing, DOI 10.1088/1757-899X/ 1076/1/012034.

[8] Hassan, S., AD-RED: A new variant of random early detection AQM algorithm. Journal of High Speed Networks, 2023(Preprint): p. 1-15, https://doi.org/ 10.3233/JHS-222.

[9] Menacer, O., A. Messai, and L. Kassa-Baghdouche, Improved variable structure Proportional–Integral Controller for TCP/AQM network systems. Journal of Electrical Engineering & Technology, 2021. 16: p. 2235-2243, doi:10.1007/s42835-021-00737-1.

[10] Kundel, R., et al. P4-codel: Experiences on programmable data plane hardware. in ICC 2021-IEEE International Conference on Communications. 2021. IEEE, https://doi.org/10.1109/ICC42927.2021.9500943

[11] Kua, J., P. Branch, and G. Armitage. Detecting bottleneck use of pie or fq-codel active queue management during dash-like content streaming. in 2020 IEEE 45th conference on local computer networks (LCN). 2020. IEEE, http://dx.doi.org /10.1109/LCN48667.2020.9314804.
[12] Park, J., Application Study of FQ-CoDel Algorithm based on QoS-guaranteed Class in Tactical Network. The Journal of The Institute of Internet, Broadcasting and Communication, 2019. 19(3): p. 53-58, http://dx.doi.org/10.7236/JIIBC.2019.19.3.53.
[13] Bhuiyan, Z.A., et al., On the (in) Security of the Control Plane of SDN Architecture: A Survey. IEEE Access, 2023, https://doi.org/10.1109/ACCESS. 2023.3307467.
[14] Alsaeedi, M., M.M. Mohamad, and A.A. Al-Roubaiey, Toward adaptive and scalable OpenFlow-SDN flow control: A survey. IEEE Access, 2019. 7: p. 107346-107379, https://doi.org/10.1109/ACCESS.2019.2932422.
[15] Cao, J., et al., Towards tenant demand-aware bandwidth allocation strategy in cloud datacenter. Future Generation Computer Systems, 2020. 105: p. 904-915, https://doi.org/10.1016/j.future.2017.06.005.
[16] Mosa Al-Mha, A. and P. Khosravian Dehkordi, Using Machine Learning to Discover Traffic Patterns in Software Defined Networks. Journal of Optimization in Soft Computing, 2025: p. 24 - 29, https://doi.org/10.82553/josc.2025.140305271129 363.
[17] Jin, H., et al., OpenFlow-based flow-level bandwidth provisioning for CICQ switches. IEEE Transactions on Computers, 2012. 62(9): p. 1799-1812, https://doi.org/10.1109/TC.2012.167.
[18] Cui, H., et al. Accurate network resource allocation in SDN according to traffic demand. in 2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering. 2015. Atlantis Press, https://doi.org/10.2991/ icmmcce-15.2015.227.
[19] Guo, J., et al. Falloc: Fair network bandwidth allocation in iaas datacenters via a bargaining game approach. in 2013 21st IEEE international conference on network protocols (ICNP). 2013. IEEE, https://doi.org/10.1109/ICNP.2013.6733583.
[20] Xu, F., et al., Ufalloc: Towards utility max-min fairness of bandwidth allocation for applications in datacenter networks. Mobile Networks and Applications, 2017. 22(2): p. 161-173, https://doi.org/10.1007/s11036-016-0739-z.
[21] Obaid, H.S.B. and T.B. Trafalis, An approximation to max min fairness in multi commodity networks. Computational Management Science, 2020. 17(1): p. 65-77, https://doi.org/10.1007/s10287-018-0336-7.
[22] Bagci, K.T. and A.M. Tekalp, Dynamic resource allocation by batch optimization for value-added video services over sdn. IEEE Transactions on Multimedia, 2018. 20(11): p. 3084-3096, https://doi.org/10.1109/ TMM.2018.2834699.
[23] Aljoby, W., et al., On SDN-enabled online and dynamic bandwidth allocation for stream analytics. IEEE Journal on Selected Areas in Communications, 2019. 37(8): p. 1688-1702, https://doi.org/ 10.1109/JSAC.2019.2917541.
[24] Marin, A., et al. Performance evaluation of AQM techniques with heterogeneous traffic. in 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). 2016. IEEE, https://doi.org/10.1109/CCNC.2016.7444755.
[25] Han, C., et al. Optimal active queue management for internet with reduced-order state-observer. in 2016 Chinese Control and Decision Conference (CCDC). 2016. IEEE, https://doi.org/10.1109/CCDC.2016. 7492857.
[26] Chrost, L. and A.J.T.S. Chydzinski, On the deterministic approach to active queue management. 2016. 63(1): p. 27-44, https://doi.org/10.1007/s11235-015-9969-9.
[27] Patel, S. and S.J.T.S. Bhatnagar, Adaptive mean queue size and its rate of change: queue management with random dropping. 2017. 65(2): p. 281-295, https://doi.org/10.1007/s11235-016-0229-4.
[28] Okokpujie, K.O., et al., Novel active queue management scheme for routers in wireless networks. International Journal on Communications Antenna and Propagation (I. Re. CAP), 2018. 8(1): p. 53-61, https://doi.org/10.15866/irecap.v8i1.14589.
[29] Hamdi, M.M., et al. Performance evaluation of active queue management algorithms in large network. in 2018 IEEE 4th International Symposium on Telecommunication Technologies (ISTT). 2018. IEEE, https://doi.org/10.1109/ISTT.2018.8638590.
[30] AL-Allaf, A.F. and A. A Jabbar, RED with Reconfigurable Maximum Dropping Probability. International Journal of Computing and Digital Systems, 2019. 8(01): p. 61-72, https://doi.org/10.12785/ijcds/080107.
[31] Kumhar, D. and A.J.I.J.o.I.T. Kewat, QRED: an enhancement approach for congestion control in network communications. 2021. 13(1): p. 221-227, https://doi.org/10.1007/s41870-020-00538-1.
[32] Gomez, C.A., X. Wang, and A.J.I.A. Shami, Federated Intelligence for Active Queue Management in Inter-Domain Congestion. 2021. 9: p. 10674-10685, https://doi.org/10.1109/ACCESS.2021.3050174.
[33] Chen, Q., et al., Research on assessment method of network quality performance for a private data network. Microsystem Technologies, 2021. 27: p. 1475-1482, https://doi.org/10.1007/s00542-020-05669-7.
[34] Campanile, L., et al., Computer network simulation with ns-3: A systematic literature review. Electronics, 2020. 9(2): p. 272, https://doi.org/10.3390/electronics9020272.

[35] Eyinagho, M.O., Minimum and maximum packets' delays determination for communication flows' delays jitters computation. Australian Journal of Electrical and Electronics Engineering, 2023. 20(1): p. 27-34, https://www.tandfonline.com/doi/full/10.1080/1448837X.2023.2000000.