Performance Analysis of Weighted Fair Queuing (WFQ) Scheduler Algorithm through Efficient Resource Allocation in Network Traffic Modeling

Mays A. Mawlood, and Dhari Ali Mahmood

***Abstract*—Bursty traffic patterns require precise classification, modeling, and comprehension to ensure adequate resource allocation, improved network security, and Quality of Service (QoS) assurance. This study introduces a methodology integrating three critical correlation metrics with scheduler algorithms, demonstrating adaptability and improved network performance. Our approach highlights the handling of irregular patterns, contributing to the development of systems that can quickly adapt to changes, significantly enhancing network performance in the context of scheduler algorithms, resource allocation, correlation metrics, and bursty traffic.**

***Index Terms*—Quality of Service (QoS), scheduler algorithms, resource allocation, correlation metrics, bursty traffic.**

I. Introduction

THE aggregate traffic in computer and telecommunications networks comprises various packet flows with different requirements for quality where some packets must be sent immediately, and others can wait longer. Most current communication protocols include a specific field within the packet header, like the class of service (CoS) field in the Ethernet frame header or the DiffServ code point (DSCP) in the IP header; DSCP is used to classify and prioritize network traffic for QoS in a 6-bit field in IP networks, which signifies the packet's class membership. To guarantee the necessary service level, packet schedulers in networking equipment, including switches and routers, are required to consider the data in these fields. Approaching a complicated mixture of network and application traffic from many perspectives is necessary to achieve optimal results, even if the traffic uses the same network path or session [1]. Concerning network functioning, for instance, a mathematical formula or framework for overall network performance, which includes all relevant network attributes, is developed [2].

The paper discusses the need for precise classification and modeling of network traffic to optimize resource allocation. It highlights scheduling algorithm Weighted Fair Queuing (WFQ), focusing on their impact on QoS aspects such as jitter, latency, and packet loss probability. The WFQ service is a widely used multi-class scheduling discipline. Before broadcasting, packets from various traffic classes in these systems are kept in separate queues.

The WFQ schedulers preserve work among the active classes, they consistently divide the entire service capacity. This allocation system allows for a flexible representation of the weights of different traffic classes [3]. WFQ was chosen over other scheduler algorithms due to its unique ability to balance, fairness, flexibility, and effective use of network resources. The primary purpose of WFQ is to improve bandwidth availability for several applications; this makes it ideal for settings with diverse traffic and QoS requirements [4].

In the next paragraphs, we will state some essential theoretical concepts:

A. Traffic Model Definition

Network traffic includes transferring several kinds of data between nodes, impacting overall performance, efficiency, and reliability. Different types of networks, such as Metropolitan, wide-area, and local area networks, require routers to implement queue scheduling algorithms to handle congestion and ensure fair resource allocation [5]. Routers forward traffic to external networks established on destination IP addresses [6]. Figure 1 shows the router's queuing scheme. Conceptually, multiple resources send packets to routers so they are placed in the queue depending on the policy of the network device, packets scheduling in the network device based on queuing algorithm, and then departure from queue [7].

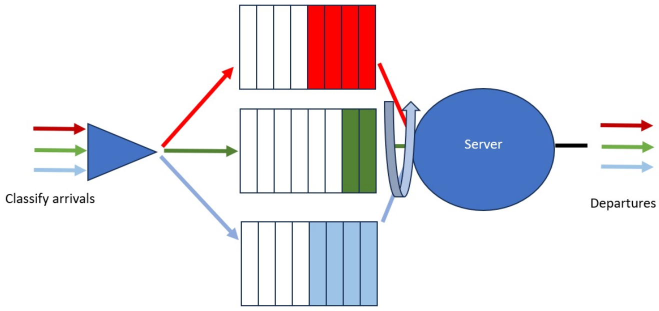

Fig. 1. Router queue scheduling.

Manuscript received May 15, 2024; revised June 8, 2024. Date of publication October 1, 2024. Date of current version October 1, 2024. The associate editor prof. Pascal Lorenz has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Department of Computer Engineering, University of Technology, Iraq, (e-mails: ce.22.02@grad.uotechnology.edu.iq, dhari.a.mahmood@uotechnology.edu.iq).

Digital Object Identifier (DOI): 10.24138/jcomss-2024-0009

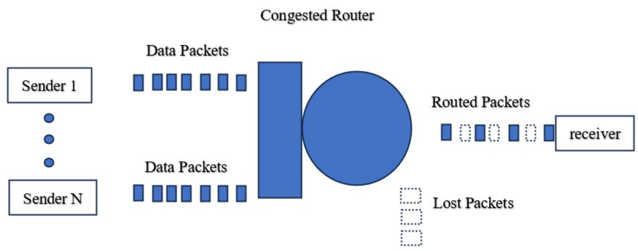

Fig. 2. Router congestion results in lost packets.

As shown in Figure 2, Congestion results in lost packets, congestion occurs when a router cannot handle the volume of network traffic when too many data packets compete for the limited space (router capacity) [8]. In allocating resources, routers must implement queue scheduling algorithms, which control the most efficient method for transmitting packets stored in the buffer. Many studies utilize scheduling algorithms to achieve appropriate QoS and guarantee fair resource allocation in network performance [9].

B. Network Traffic Characteristics

Quality of Service (QoS) is an approach by which networks offer different classes of service for various types of traffic. Providing parameters such as throughput, latency, jitter, and packet loss is essential. Affecting the efficiency of priority systems as (WFQ) [10]. The research can use metrics like throughput, average queue length, average waiting time, and utilization to compare the system's performance [11]. Mean bandwidth allocation was accomplished by looking at the WFQ scheduler and developing a reduplicated mathematical model. These metrics may be monitored throughout this process: the ratio of packet arrival, average throughput, average delay, and packet loss ratio [12].

C. Queueing System and Scheduler algorithms

A queuing system models the process of elements queuing for processing or service. Scheduling algorithms like WFQ are essential for fair and effective distribution of network resources. The following are terms and components of a queueing system: Arrival and service procedures, queuing systems arranging incoming jobs, and scheduler algorithms intelligently controlling their processing. Figure 3 represents a basic queue model containing arrivals packets and queues where waiting for position and service after service completes the packet departure [13].

The classical queuing models that are most widely recognized include the M/M/1 and M/M/c models [14]. Packet scheduling algorithms handle bandwidth allocation per flow to guarantee QoS and effective congestion-fair resource allocation mitigation across different traffic classes [15].

D. The Role of WFQ in Network Traffic Modeling

Zhang, Demers, Keshav, and Schenke presented this algorithm in 1989 [16]. This algorithm offers fair output bandwidth allocation regarding the assigned weights. It is different from

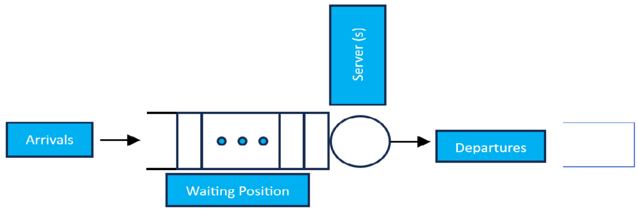

Fig. 3. A basic queue model.

fair queueing prepared with a weighted bandwidth allocation. Figure 4 represents a WFQ scheduling, which guarantees fairness in bandwidth allocation among known flows. Bandwidths allocated to each flows based on its weights [15].

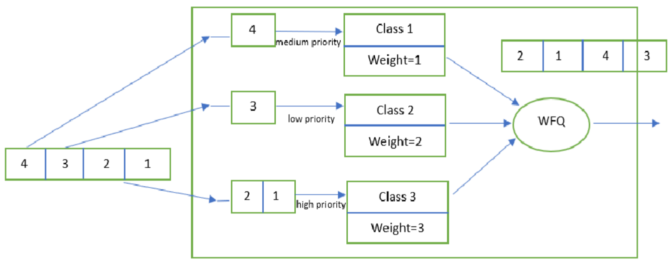

Fig. 4. WFQ scheduling.

In Cisco routers, the WFQ prioritizes traffic according to the specified weights. Every flow in WFQ is given a weight, and the scheduler allots bandwidth to these weights. The most considerable bandwidth is allocated to flows with greater weights to ensure that various traffic streams are treated equally. The essential purpose of WFQ is to increase bandwidth availability for several applications [17]. Figure 5 represents the WFQ algorithm of Cisco routers where every queue has a specific weight for allocated bandwidth fairly [18].

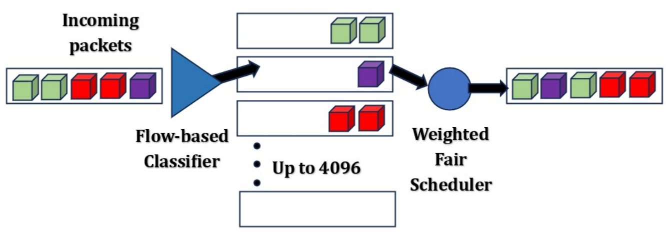

Fig. 5. The WFQ algorithm of Cisco routers.

Technical Details:

- This algorithm can characterize the statistical features of bursts, such as duration and volume, by analyzing the technical elements necessary to explain how burst patterns are exhibited in network traffic.
- It can indicate the flaws of traditional scheduling algorithms in confronting highly irregular and random traffic. Additionally, it emphasizes the possibility of failure of

these methods in allocating optimal resources and defending Quality of Service (QoS).
- It extends the technical need to develop scheduling algorithms that can dynamically adapt to the varying nature of network traffic. It also inspects the likelihood of these methods to improve bandwidth utilization and network throughput.
- It is essential to effectively integrate scheduling algorithms with correlation metrics and investigate the technical situations. Furthermore, it observed the possible computational and experimental challenges to overwhelm them. It also emphasizes the importance of this integration in reaching more knowledgeable scheduling conclusions that lead to optimizing network performance.
- It is worth noting that this algorithm can improve network performance metrics like maximizing throughput, minimizing latency, and improving reliability by applying the integrated scheduling solution.

The main contributions of this study are listed below:

- Integration of scheduling algorithms with correlation metrics: This is a new strategy for examining network traffic by integrating sophisticated scheduling algorithms with correlation measures. It was improved to resolve a gap in existing research approaches: patterns, especially bursty traffic, and their ambiguous effects on bandwidth use and network performance.
- Through analysis of bursty traffic patterns: Analyzing erratic internet traffic patterns comprehensively, especially bursty traffic, bursty traffic on network traffic is characterized by sudden, unpredictable surges in data transmission, often followed by periods of low activity. Network performance is affected by This variability, which can significantly lead to congestion, increased latency, packet loss, and decreased overall throughput. Network administrators can only expect these bursts by analyzing traffic and understanding its patterns, resulting in inefficient resource allocation (bandwidth) and performance of the network. Failure to manage burst traffic effectively can degrade the quality of service (QoS), affecting real-time applications like video streaming. Analyzing traffic patterns helps in proactive management, ensuring a highly efficient network process.
- Introduction of a coordinated scheduling system: Participating in presenting a more practical method for controlling network traffic. A novel integrated scheduler solution that uses both scheduling algorithms and experimental data analysis was suggested. Such an approach enhanced QoS. By reviewing previous studies, It becomes necessary to analyze real data. Understanding network traffic can help manage congestion more efficiently and ensure fair resource allocation. In our approach, we analyze simulated data using the same pattern of tracing data of Ethernet packets.
- Discoveries into network traffic dynamics: Suggesting new viewpoints on irregular traffic's fundamental patterns and dynamics and improving comprehension of maximizing network efficiency in different traffic situations.
- Improvement of network management strategies: The findings extend to more flexible and effective network management techniques, which may promote bandwidth usage and overall network performance.
- Empirical validation of the theoretical models: Empirical analysis indicated the advantage of this integrated approach for scheduling. Furthermore, one of the most significant aspects of this study is ending the gap between theoretical models and real network traffic behavior.

This study is arranged as follows: Section II discusses related work. Section III presents the problem statement. Section IV involves bursty traffic measures based on correlation metrics. Section V includes results and analysis. Section VI indicates discussion. Section VII exhibits the conclusion.

II. Related Work

In the work introduced by [19], the burst method has been utilized as a key factor for assessing certain scheduling algorithms and the characteristics of internet traffic. These scheduling algorithms, aimed primarily at fair service distribution, take into account only the guaranteed service rate in their scheduling decisions. Work by [20] aims to offer comprehensive insights into enhancing QoS, utilizing networks, and calculating the effects of packet scheduling in relation to traffic intensities (TI). For this purpose, the Traffic Intensity-based Packet Scheduling (TIPS) algorithm was employed for packet scheduling in a simulated network environment. The research measured factors such as throughput, end-to-end latency, and jitter. The results indicated that TIPS achieved superior QoS performance in terms of network utilization. As per [21], a prioritized Traffic Intensity-based Media Access Control (PTI MAC) protocol has been effective in ensuring the on-time delivery of high-priority packets while also boosting throughput. In addition, the study looked at the link between traffic intensity and network load and divided the priority threshold based on the different degrees of traffic intensity. The results showed that the proposed protocol met the capacity and end-to-end total latency requirements. Wrok in [22] proposed comparing the most popular burstiness measures with various traffic kinds generated by a variety of traffic models in order to investigate their use. The obtained results indicate that first-order metrics like the squared coefficient of variations and the ratio of peak-to-mean cannot effectively describe the prominent bursty feature under several scenarios. The indices of dispersion derived from the second-order measurements were discovered to be highly practical and valuable. The research in [13] focused on modeling and evaluating various services by investigating traffic shaping mechanisms within the WFQ framework, implementing QoS in packet networks, assessing the performance of these modeled systems, and validating the traffic shaping approaches. The findings highlighted the impact of the number of high-priority flows on average waiting times and queue length, which is crucial for calculating delay, jitter, and packet loss. Possibility with QoS metrics for FQ, CQ, PQ, WFQ, and FIFO algorithms. Work in [17] conducted a comprehensive analysis of traffic scheduling in IPv4 and IPv6 networks, particularly for multimedia applications.

This study explored the performance of different scheduling algorithms across various traffic types, including FTP and HTTP. It was observed that among the queuing techniques (WFQ, FIFO, PQ), WFQ emerged as the superior choice, offering enhanced performance in comparison to the other algorithms. While FIFO did not result in as much end-to-end jitter, lost packets, or latency for audio and video conferencing applications, it did produce better performance than PQ for HTTP applications. In order to prevent network congestion and enhance performance, [23] developed a packet-scheduling method that blends PQ with WFQ. As opposed to situations in which packet scheduling was not used, the method effectively reduced latency, greatly enhancing network performance. [12] intended to improve a convenient simulation algorithm model to identify queue formation and loading control by investigating many types of flows. The findings illustrated that the PQ algorithm is designed primarily for high-priority flows. In contrast, the FIFO algorithm serves more as a benchmark for comparison than for actual service data. The WFQ algorithm demonstrated the most consistent optimal modeling results, as all its factors stayed included in the established limitation states across all examined kinds of data streams. Work in [24] looked into queuing systems with a range of arrival rates to consider periodic patterns and long-term correlations. A practical model that allows for the separate analysis of the effects of periodic and stochastic components was used in the study. Furthermore, the authors proposed an approximate analytical method to do this. The proposed model took into account the additional waiting times brought on by arrival rate fluctuations, which reduced the estimation error. Approaching a complicated mixture of network and application traffic from many perspectives is necessary to achieve optimally, even if the traffic uses the same network path or session [1]. The challenges must be addressed by redefining the performance of the network. Concerning network functioning, for instance, it developed a mathematical formula or framework for overall network performance [2], which includes all relevant network attributes.

III. Problem Statement

This study aims to bridge the knowledge gaps in the literature regarding the advantages of integrating scheduling algorithms with correlation metrics. It addresses the challenge posed by bursty traffic, characterized by irregular bursts of data transmission, which traditional scheduling techniques need help managing.

Irregular internet traffic is one of the most critical factors that is overlooked. Bursty traffic from Mobile users generates variability in latency; this irregular nature of many forms of Internet communication might have different effects on network performance and bandwidth use. As user needs change and modern networks become more complicated, new approaches to network management are needed. Bursty traffic, characterized by irregular bursts of rapid data transmission, offers a remarkable challenge to traditional scheduling techniques. To address this challenge, we provide an integrated scheduler solution incorporating advanced scheduling methods and correlation measurements. By building on and extending previous research, this work can offer a comprehensive analysis of the relationship between the scheduling method and empirical measurement data. By employing this inclusive approach, we hope to obtain insight into the underlying patterns of bursty traffic; comprehending traffic behavior patterns assists in optimizing network performance, effectively managing resources, maintaining high QoS, avoiding congestion, and ensuring a robust network environment. which will offer more effective and adaptable network management strategies.

The mathematical model in this paper includes the following:

1) Correlation Metrics:
   - Squared Coefficient of Variation (CV^2) is a measure of burstiness about a Poisson process. It is defined using a mathematical formula. The Squared Coefficient of Variation is computed as indicated in formula equation (7) illustrated in section IV.
   - Inter-Departure Interval (IDI) is a mathematical modeling of IDI that may have an important role in offering awareness about the temporal features of burst traffic. It is utilized to quantify the time intervals between packet arrivals. This index is estimated by equation (8) indicated in section IV.
   - Inter-Departure Correlation (IDC) is known for its ability to capture the correlation between packet inter-arrival times over different delays via autocorrelation functions. equation (10) can be applied as shown in Section IV.

   In summary, correlation refers to the connection between packet arrival times in network inter arrivals. It indicates how the arrival time of one packet is related to the arrival times of subsequent packets, which can affect congestion and network performance. Correlation metrics provide practical insights into different aspects of traffic patterns, allowing network operators to predict, manage, and optimize network performance effectively. Understanding these metrics helps address congestion, optimize resource allocation, and ensure a stable and efficient network operation.
2) Scheduling Algorithm and Network Performance:
   - According to this algorithm, mathematical equations will be produced through network performance measurements like latency and packet loss throughput to clarify the interaction between scheduling algorithms and them.
   - It is also significant to identify the performance quality of several scheduling strategies across multiple traffic scenarios. It may involve creating utility functions.

IV. Bursty Traffic Measures Based on Correlation Metrics

The study introduces several metrics to analyze bursty traffic:

- Squared Coefficient of Variation (CV^2): Measures variation in inter-arrival times between packets.

- Inter-Departure Interval (IDI): Assesses the regularity of traffic bursts.
- Inter-Departure Correlation (IDC): Evaluates the correlation between inter-departure times of consecutive packets.
- Intensity: Measures the rate of data packet transmission.
- Stationary: Indicates the consistency of statistical properties over time.
- Correlation coefficient: Quantifies the linear relationship between two variables.

This study used the tracing data of Ethernet packet arrivals; the Ethernet traffic between Bellcore corporate add-in laboratory hosts and all hosts outside Bellcore on the Internet represented mainly as external traffic was collected. The number of arrivals was a record one million. They collected the data by observing the Ethernet, the key supplier of this separate router. When managing these data, the entry between Bellcore hosts and the external world was unlimited [25].

The simulated datasets were generated from the OMNeT++ simulation environment. In our simulation, the WFQ algorithm was used in a network configuration with two sources, one queue and one sink. In order to ensure equitable resource allocation, they are then scheduled in the queue using the WFQ algorithm based on their weight assignments. The number of arrivals was generated by simulation, and it was one million. To collect the data, they were gathered by observing the Ethernet. According to [26], OMNeT++ incorporates C++ using message forwarding and the descriptive language NED (Network Description Language) to define dynamic behavior. OMNeT++ (Objective Modular Network Testbed in C++) has several benefits, such as visualization tools for examining simulation results, modularity, scalability, and extension. The WFQ algorithm and network performance under many circumstances may be well understood using this simulation.

A. *Performance Metrics*

This model is important for representing the unpredictable nature and variability of traffic bursts. It also supplies mathematical descriptions of bursty traffic using probability distributions. Models may use high distributions to depict packet sizes and inter-arrival periods.

An arrival process is defined by its sequence of inter-arrival time random variables $A_1, Az, ...$, which can be used to describe a basic queueing system. A frequent assumption is that the inter-arrival time sequence is independent and uniformly distributed, resulting in an-arrival process. $E[A] = T_A$ is the average inter arrival time, while the average arrival rate λ is its opposite [27].

$$\lambda = \frac{1}{\bar{T_A}} \quad (1)$$

For the exponential interarrival time distribution, which is commonly observed, the process of arrival corresponds to the Poisson model. Additionally, it is crucial to define the service times, $B_1, B_2, ...$, for the following tasks in sequence. This sequence is typically considered as a set of independent random variables that possess a shared distribution function. T_B denotes the average service time, symbolized as $E[B]$, and μ represents the inverse of T_B [27].

$$\mu = \frac{1}{\bar{T_B}} \quad (2)$$

- Intensity: Traffic intensity refers to the count of active sources at a given moment, specifically when the instantaneous traffic intensity falls within a specified range of sources. This range is associated with various lines, servers, trunks, circuits, channels, and computers. The traffic intensity's statistical instants (variance, mean value) can be calculated over a specific T as a time. Regarding the mean traffic intensity, we obtain [28]

$$Y(T) = 1/T. \int_0^T n(t)\, dt \quad (3)$$

- Stationary: This property can be identified as the possibility distributions illustrating the point method, which are independent of the instant of the period. Traffic intensity can also be defined as a random $t_2 > 0$ and all of k values ≥ 0, the likelihood that there are k arrivals in $t_1, t_1 + t$ is independent of t1, i.e. that there are k arrivals. For all values of t, k, there is [28]:

$$p(N_{t_1+t_2} - N_{t_1} = k) = p(N_{t_1+t_2+t} - N_{t_1+t} = k) \quad (4)$$

- Squared Coefficient of Variation: To determine random factors, the following parameters related to the initial two moments are employed; the expected value, or mean value, is E{T} in the first moment:

$$m_i = E\{T\} \quad (5)$$

Variance = standard deviation: 2nd central moment

$$\sigma^2 = E\{(T - m_1)^2\} \quad (6)$$

Standard deviation: It is recognized as the square root of the variance, equivalent to σ. The coefficient of variation is a normalized measure for assessing distribution distortion. The ratio of the standard deviation to the mean value is used to calculate it:

$$CV : standarddeviation = \sigma / m_1 \quad (7)$$

This quantity is also used to describe discrete distribution.

- Index of dispersion for intervals: For the sake of defining second-order features for the interval illustration, (IDI) is the index of dispersion that can be used for intervals, and it is determined as:

$$IDI = Var\{X_i\} / E\{X_i\}^2 \quad (8)$$

where Var represents the variance, which equals to:

$$Var = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (9)$$

Xi is identified as the inter-arrival time, and $\bar{x}$ is characterized as the central evaluator of the unknown population means value. The IDI is equivalent to one under the Poisson approach, represented by exponentially distributed service times. Typically, IDI is more complected to assess than IDC, especially in measuring accuracy and facilitating traffic processes. Digital technology is better suited for monitoring IDC, whereas accurately observing IDI presents more challenges [28].

- Inter-Departure of correlation: To define the number illustration's second-order features, the dispersion index can be used for counting the (IDC).This index characterizes the differences in the arrival procedure throughout a time interval t and is identified as [28]:

$$IDC = Var\{N_i\}/E\{N_i\} \tag{10}$$

where Var refers to Variance, and $E\{N_i\}$ refers to expected mean. To calculate An IDC (t), divide the time interval$t$ into x breaks of period t/x and note the numeral of incidents through these periods. According to the Poisson method, IDC gets comparable to one [28].

- Correlation: A statistical metric known as the correlation coefficient is used to determine the degree and direction of the linear relationship between two variables, which is defined by [29]

$$\gamma = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^2}} \tag{11}$$

B. Scheduler Algorithms

All the output bandwidth will be distributed equitably in the queue based on weight, according to the WFQ algorithm. The available bandwidth is distributed across the service classes based on their associated weights and wait times [30]. To guarantee equitable and effective distribution of network resources between two flows or classes of traffic, our algorithm (WFQ) leveraged networking. Each class has its arrival rate (λ) by a traffic source in a given class per unit of time, the average number of packets produced [31]. In the first case, two separate arrival rates correspond to a different class and an equal bandwidth distribution. We compared the analysis to tracing data of Ethernet packet arrival. In the second case, we compared the analysis of two datasets. The first dataset is generated from two classes, each with the same weight. The second dataset is generated from two classes with priority distribution. The weight of the high-priority class is 0.9, and for the low-priority class, it is 0.1. The rationale for researching the case with equal weight allocation has been clarified to underscore its crucial role in providing a baseline for comparison with other weighting schemes. This comparison is instrumental in demonstrating the proposed scheduling methodology's relative performance and benefits under different conditions.

C. Resource Allocation

In the network simulation we conducted, the total capacity for data transfer was capped at 10,000,000 bits per second (bps). In this network simulation, the data transfer rate peaks at 10 million bits per second, representing the network's total capacity. The study introduces two distinct traffic classes; each one emits packets to mimic various network traffic patterns for different services. An exponential distribution shapes their lengths. The WFQ algorithm manages bandwidth distribution between these classes. The key to WFQ's efficiency is the allocation of weights to each traffic class, which impacts network access priority. In the first case, both classes received equal weights of one, ensuring a balanced bandwidth division with Tracing data of Ethernet packet arrival. Both of them had the same arrival rate. This equal weighting is crucial, as it guarantees each class an equal share of the total capacity. The WFQ algorithm is vital in maintaining resource fairness and promoting equal distribution of network resources across traffic classes. Considering the second case, the first dataset generated from two classes ensures each class has equal weights and that bandwidth is distributed evenly between the two classes. The bandwidth distribution reflects these weight ratios, as in the second dataset generated, 0.9 for class 1 and 0.1 for class 2.

D. Traffic Model

A powerful and feature-rich simulation tool, OMNeT++ can simulate many networks. As a powerful simulation tool for network modeling, OMNeT++ stands out due to its modular and flexible architecture, which allows for the comfortable customization and extension of simulation components. It features a user-friendly graphic interface that encourages the design and visualization of network models in real-time, improving efficiency and effectiveness in simulation tasks. Its implementation and scalability drive it to be capable of running large-scale simulations efficiently, including similar and distributed simulations. Network behaviors and protocols' capability to provide detailed and accurate modeling makes it ideal for academic research and educational purposes, enabling a deep understanding. Figure 6 illustrates a diagram of the WFQ scheduling simulation model. The traffic sources produce packets for their appropriate classes based on predetermined parameters (such as packet sizes and inter-arrival periods). After that, a shared queue module uses a scheduling algorithm to enqueue these packets (WFQ). As packets are dequeued from the queue, the sink module acts as the endpoint, receiving and processing them. The simulation simulates the behavior of a network with distinct traffic classes by specifying the traffic sources, queue, sink, and the appropriate scheduling algorithm. That enables the examination of performance metrics. This paper presents the scenarios of the weighted fair queue as a queuing system.

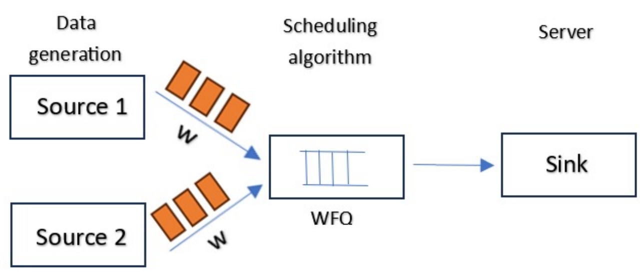

Fig. 6. Illustration diagram of WFQ scheduling simulation model.

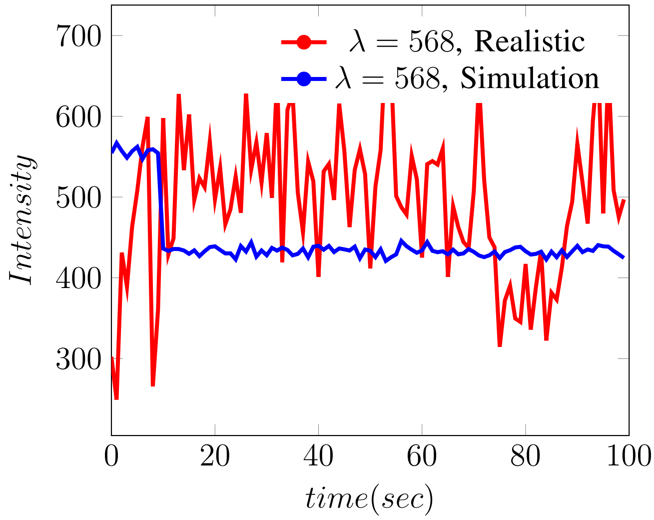

Fig. 7. Traffic Intensity for simulated and real-world data.

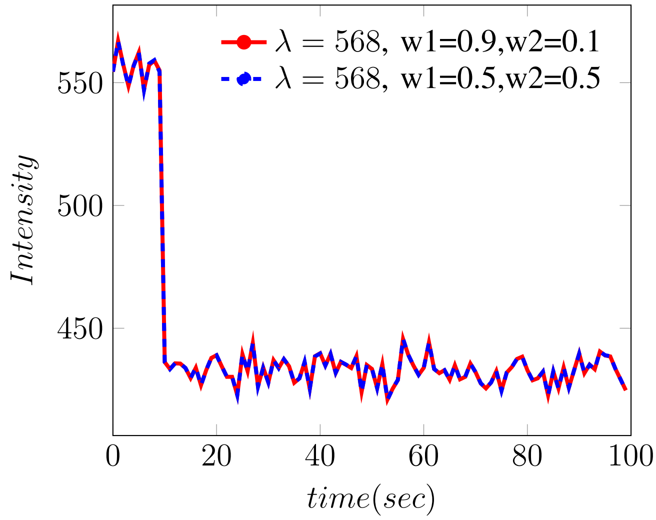

Fig. 8. Traffic Intensity for two simulated data.

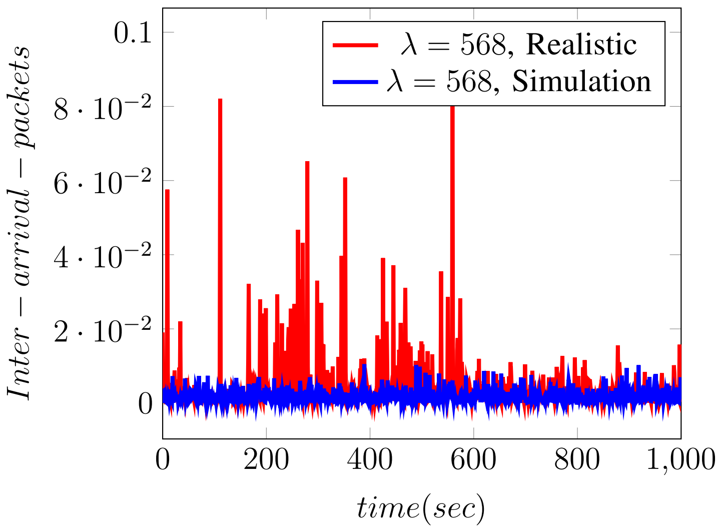

Fig. 9. Stationary of inter-arrival times for simulated and real-world data.

V. Results and Analysis

In this section, we proceed with the statistical analysis of datasets. We used equation (1) to evaluate the arrival rate. The results showed that there was practically no difference between the arrival rate (λ) of the data generated and the tracing data of Ethernet packet arrival to reflect and address bursty traffic for real-world network management indicated bursty traffic. This equation allowed us to accurately compare the datasets and ensure the validity of our findings.

To calculate the traffic intensity of data arrivals for datasets, it should be noted that a 10-second length of the period was employed in the traffic intensity calculation as stated in the equation (3), which is expressed via the ratio of the number of arrival to the length of period. In the first case, Tracing data of Ethernet packet arrival with a simulated dataset has equal weights with two classes, both of them having the same arrival rate. The results, as seen in Figure 7, indicated that the tracing data of Ethernet packet arrival irrigates more than the simulated data, but overall, the traffic intensity of the simulated data increases with time. The tracing data of Ethernet packet arrival had clear peaks and valleys, and there were several times when the traffic intensity levels were higher than the simulated data. The pattern in the simulated data was smoother, more constant, and had fewer sharp fluctuations in traffic intensity. When comparing the simulated and tracing data of Ethernet packet arrival, the tracing data of Ethernet packet arrival shows a more noticeable bursty traffic pattern and a substantially higher traffic intensity. In the simulated data, the traffic flow is more uniform. It lacks burst-like features, which contrast with the fluctuations in the tracing data of Ethernet packet arrival and suggest a potential bursty element.

To compute traffic intensity for the second case illustrated in Figure 8, it is worth noting that the traffic intensity curves for the two datasets are almost identical. The traffic intensity values of the datasets generated from two classes with unequal weights are marginally higher than those for the dataset generated from two classes with equal weights. According to Stationary, it converts the arrival time of packets to inter-arrival time for datasets as in equation (4). Figure 9 shows the results of computing the stationary for the first case. Over time, each curve is visually stationary. This indicates that their statistical characteristics, such as mean, variance, and auto correlation, stay primarily unchanged during the observed time frame. Tracing data of Ethernet packet arrival oscillates at a particular level, indicating that its statistical characteristics vary slightly over time. Concerning the simulated data, it has equal weight for two classes that show a slightly flat trend, which confirms stationary. According to both datasets, fluctuation has been reduced, and system behavior is more stable and constant. As illustrated in Figure 10, the difference in the stationary curves between datasets generated from two classes with equal weights and unequal weights highlights the impact of weight distribution. The stationary value of datasets created from classes with equal weights tends to be barely higher and slightly more variable than that of datasets with

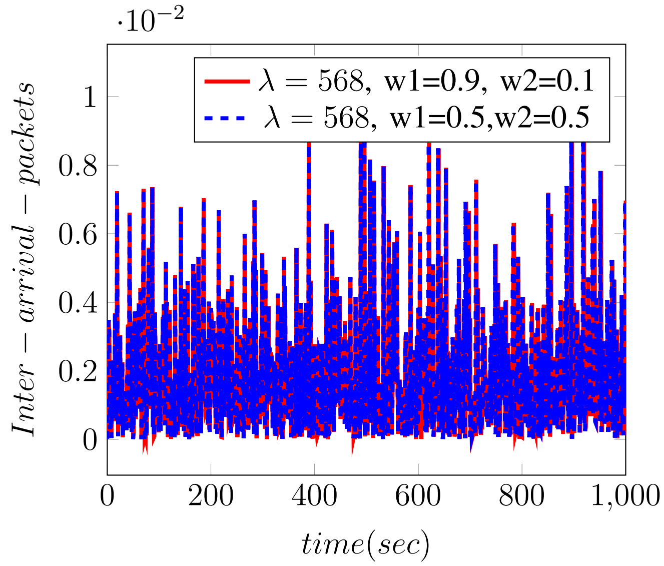

Fig. 10. Stationary of inter-arrival times for two simulated data.

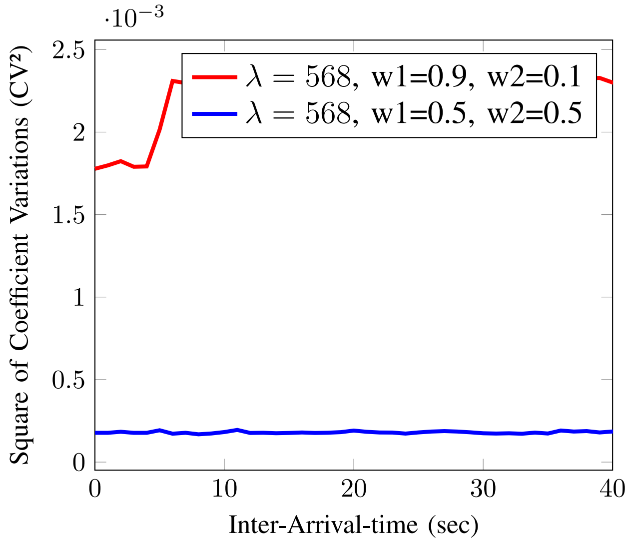

Fig. 12. Square coefficient of variations (CV²) for two simulated data.

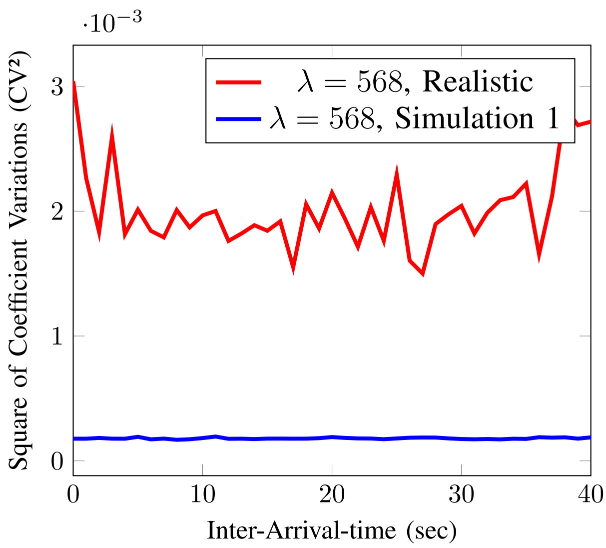

Fig. 11. Square coefficient of variations (CV²) for simulated and real-world data.

unequal weights.

Regarding CV^2, datasets were computed in equation (7). The variance of inter-arrivals was computed in equation (9) and mean for inter-arrivals. After computing CV^2 for the first case, as illustrated in Figure 11, A higher CV^2 indicated that the tracing data of Ethernet packet arrival traffic arrival patterns were more unpredictable and random. On the other hand, the simulated data's lower CV^2 indicates more regular, less chaotic bursts with stable inter-arrival periods. The higher CV^2 notably demonstrates more dispersion throughout the distribution in the tracing data of Ethernet packet arrival or more variability in bursts.

According to computing CV^2 for the second case with two datasets, as shown in Figure 12. A higher CV^2 shows that the dataset generated from classes of unequal weights had more varied and unpredictable behavior. In contrast, the other dataset appears to be more regular. The higher CV^2 notably exhibits more dispersion throughout the distribution in the dataset with unequal weights and more variability in bursts.

For the calculation of IDI, this study dealt with stationary as inter-arrivals with 20 lag, expressed via the variance of inter-arrivals as in equation (9) to the squared mean of inter-arrivals. After performing the equation (8), the results are shown in Figure 13. The IDI curves highlight the contrast between the datasets having equal weights and the tracing data of Ethernet packet arrivals. The simulated data curve shows a pattern of increasing consistency, characterized by IDI values and a flatter trajectory, indicating a smoother flow of arrival times with less fluctuation. The minor variations in IDI values imply a stable and predictable pattern. In contrast, the tracing data of Ethernet packet arrival's IDI curve exhibits more dynamic and bursty tendencies. The presence of irregular, high IDI peaks, quick changes, and overall higher IDI values point to a traffic pattern with sporadic and unpredictable increases in arrival times. The wider dispersion of arrival times is evident from a much larger variance than the squared mean, which further emphasizes the bursty nature of the simulated data. In conclusion, the simulated data exhibits a more constant and consistent flow. In contrast, the features of the IDI curves suggest that the tracing data of Ethernet packet arrival has a bursty traffic pattern.

The IDI curves, as shown in Figure 14, emphasize the difference between datasets generated from classes with unequal weights and datasets having equal weights. About the first dataset, the IDI curve displays more dynamic and bursty trends. It appears irregular, with slightly high IDI peaks and overall higher variability. IDI values point to a traffic pattern with sporadic and unpredictable growths in arrival times. The second dataset with equal weights for each class shows a pattern of raising consistency, characterized by IDI values and more balance, showing a smoother flow of arrival times with less fluctuation. Here, the features of the higher IDI curves indicate a bursty traffic pattern.

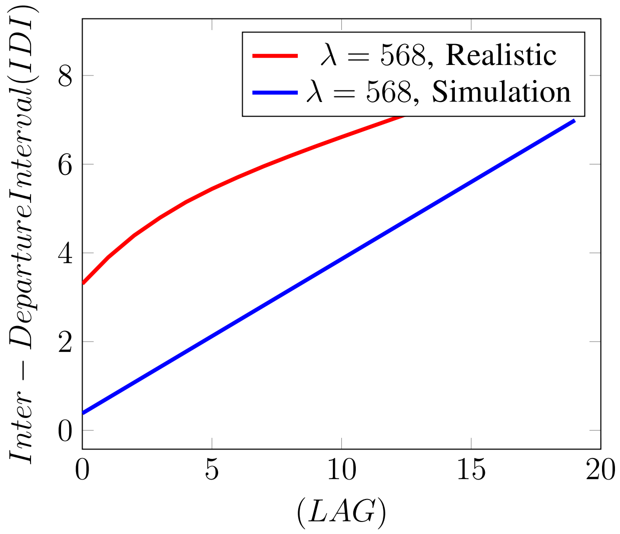

Fig. 13. Inter-Departure Interval (IDI) for simulated and real-world data.

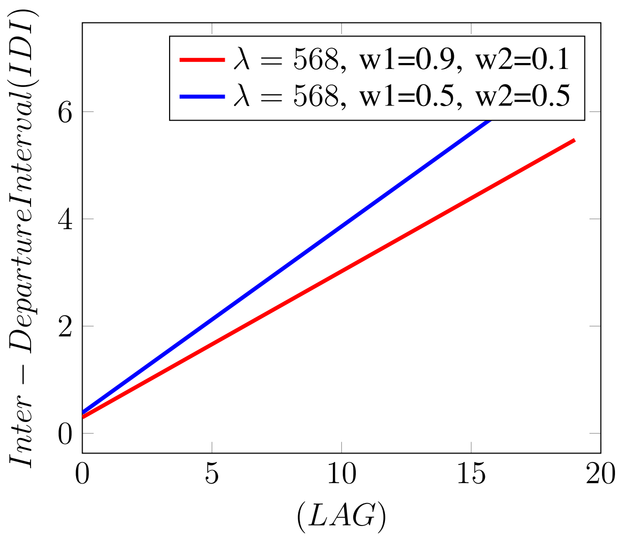

Fig. 14. Inter-Departure Interval (IDI) for two simulated data.

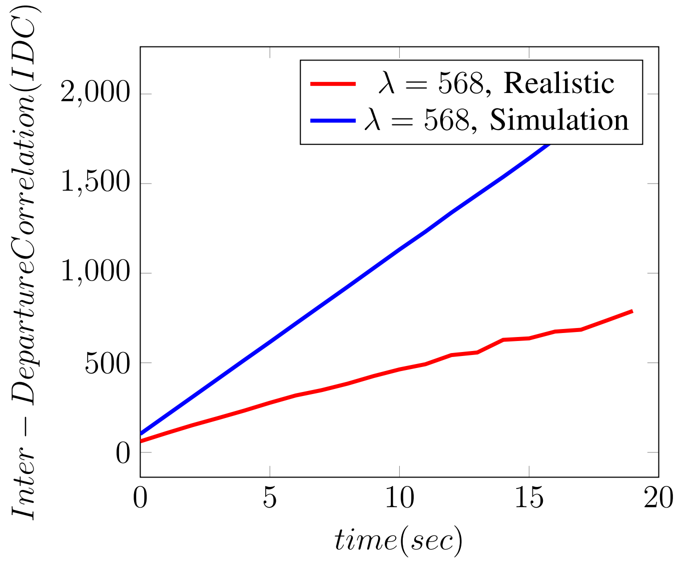

Fig. 15. Inter-Departure Correlation (IDC) for simulated and real-world data.

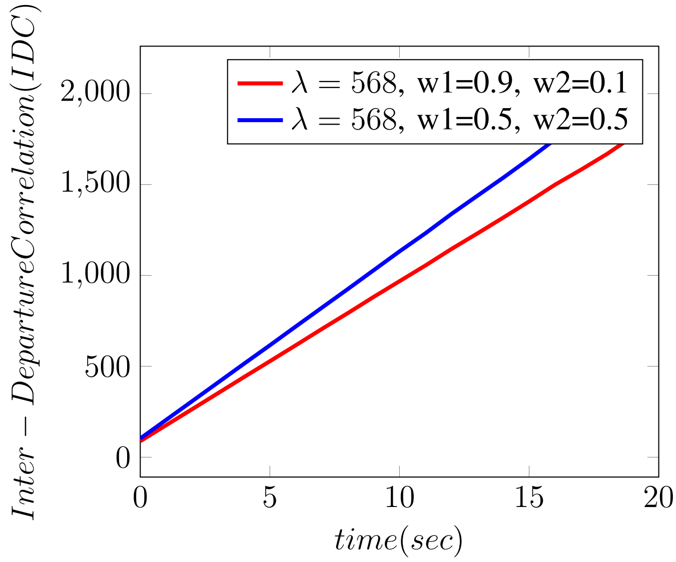

Fig. 16. Inter-Departure Correlation (IDC) for two simulated data

IDC is known as a count of the arrival time of packets. To assess the degree of dispersion or variability for count data, the following equation (10) can be applied, which takes twenty seconds for datasets when the lag is equal to 20. A 20-unit lag indicates that a 20 (sec) time shift is used when comparing the inter-departure periods in the IDC study. This enables analysts to evaluate how well inter-departure timings at one-time points correlate with those 20 sec later. This parameter aids in comprehending patterns and temporal relationships in network traffic behavior. As shown in Figure 15, the results of the tracing data of Ethernet packet arrival indicate a small value of correlation association within the inter-departure, and it shows more fluctuations, randomly or irregularly inter-arrival departure, which in turn displays septal cluttering of events. The sudden increase in IDC may refer to bursts in arrivals interspersed by depressive intervals. The inter-departure interval association was evident in the simulated data generated from two classes with equal weights due to a slight rise in IDC, and arrivals are less grouped than in Ethernet packet arrival tracing data. The gradual and continuous increase in IDC exhibits a less unpredictable traffic flow than Ethernet packet arrival tracing data. As shown in Figure 16, the results of the IDC of the simulated dataset generated from two unequal classes show a slight correlation association within the inter-departure, which indicates more instabilities and irregular inter-arrival departure. Potential bursty traffic patterns are consistent with the growing trend toward bursty behavior. The IDC association was apparent in the simulated data generated by two classes with equal weights due to a slight increase in IDC, and arrivals are less grouped than the irregular IDC curve of the dataset generated from the unequal weight of the two classes.

To compute correlation, arrival time is translated to inter-

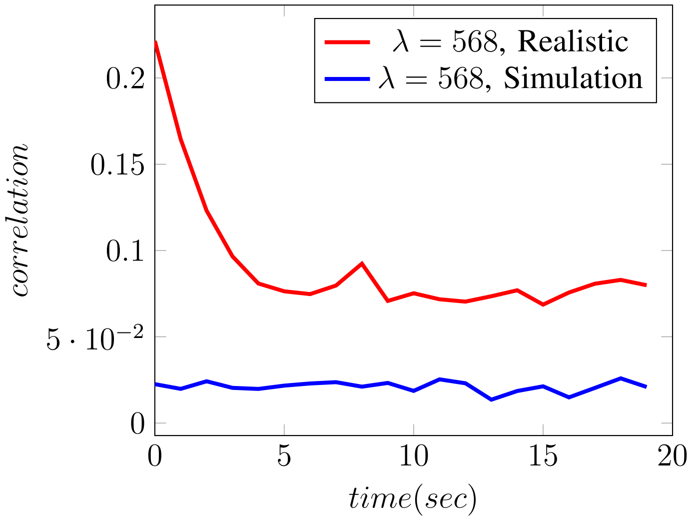

Fig. 17. correlation for simulated and real-world data.

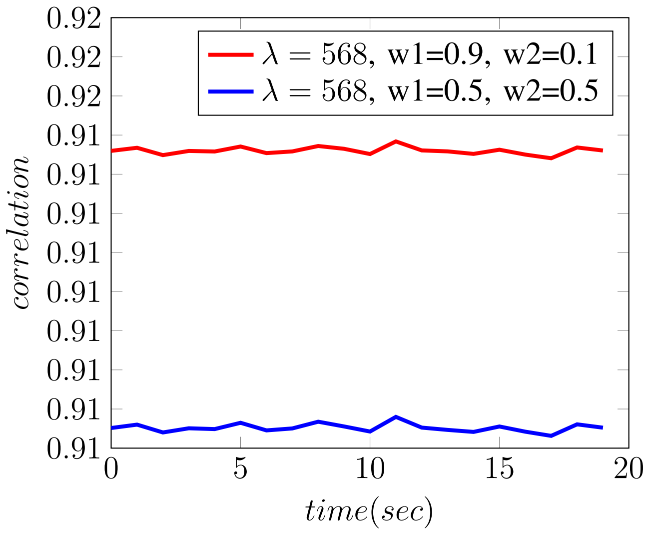

Fig. 18. correlation for two simulated data.

arrivals, then the copy of data will be shifted to one position, and the correlation coefficient for datasets will be computed as in equation (11). The results illustrated in Figure 17 showed that the simulated data generated from equal weights had a more robust overall correlation, characterized by its less abrupt fluctuations and smoother trend, which indicated a more stable relationship between successive datasets. Compared to the simulated dataset, the Ethernet packet arrival tracing data exhibits a more dynamic pattern in its correlation. The genuine statistics clearly showed patterns of bursty traffic. Even if the simulated data showed a smoother trend, some slight variations might indicate a minor bursty component but less than the tracing data of Ethernet packet arrival.

The correlation results illustrated in Figure 18 showed that the simulated data generated from equal weights of two classes had a more robust overall correlation, representing slight abrupt fluctuations and smoother tendency, which showed a more regular relationship between successive datasets. On the other hand, the correlation of simulated data generated from unequal weights of two classes shows a more dynamic pattern in its correlation. The genuine statistics clearly showed patterns of bursty traffic.

According to [32] and [33], their algorithm can be more valuable in certain conditions. It insufficiently handles the problems caused by bursty traffic, making it less effective and less suitable for practical applications. Robust performance and scalability in dynamic network conditions may be achieved by including bursty traffic management considerations in algorithm design and optimization. The study of [34] included the traffic intensity distributions of traffic on Internet networks by analyzing the traffic intensity over various periods using statistical techniques. The findings demonstrate the statistical significance of long-term patterns in traffic intensity changes. This research must construct computer network simulation models and propose ways to manage irregular and bursty real-world traffic by reflecting the behavior of real-world traffic in the scheduler algorithm and handling it in dynamic network environments to provide robust performance and scalability. Our study proposes a new methodology that combines the crucial correlation metrics, including Squared Coefficient of Variations (CV²), Inter-Delay Interval (IDI), Inter-Delay Correlation (IDC), stationary, traffic intensity, and correlation with scheduler algorithms, which were not presented in the previous studies.

VI. Discussion

This study used the tracing data of Ethernet packet arrivals, the Ethernet traffic between Bellcore corporate add-in laboratory hosts, and all hosts outside Bellcore on the Internet, representing mainly the collected external traffic. The number of arrivals was a record of one million. The data were collected by observing the Ethernet, which is the key supplier of this separate router. When managing these data, the entry between Bellcore hosts and the external world was unlimited [25]. According to the first case, the statistical analysis included both the tracing data of Ethernet packet arrivals and simulated data arrivals generated from two sources, two queues (WFQ) scheduling algorithms attain equitable bandwidth distribution across various flows or queues. The number of arrivals was one million. The total capacity for data transfer was capped at 10,000,000 bits per second (bps) distributed equally to sources. Each class in the scenario has its arrival rate (λ), and both sources have equivalent weights. According to the second case, the statistical analysis contains two simulated datasets; first simulated data arrivals were generated from two sources, two queues (WFQ) scheduling algorithms bandwidth distribution; the first class assigns a higher weight of 0.9, and the second class assigns a weight of 0.1, with lower priority. Referring to the second dataset generated from two sources, two queues (WFQ) scheduling algorithms have two equal weights. For both datasets, the number of arrivals was one million. The total capacity for data transfer was capped at 10,000,000 bits per second (bps) distributed based on their weights. However, all datasets have the same arrival rate (λ), which was 568 (packets/sec), indicating bursty traffic, and the number of arrivals is one million. The results revealed that the correlation measurements of the Ethernet packet arrival tracing data had a bursty pattern. In the first case, the traffic pattern of the simulated dataset exhibited less abrupt fluctuations in traffic intensity over time and was smoother and more constant. However, Ethernet packet arrival tracing data displayed a more chaotic and dynamic traffic pattern with clear peaks and valleys in traffic intensity. Stationary and correlation for tracing data of Ethernet packet arrival indicated more fluctuations in contrast to the simulated data. According to the IDI of Ethernet packet arrival tracing data, a high IDI value indicates more differences between inter-arrival times and more stability in the simulated data. Referring to counting the arrival of packets, there is more unpredictability for Ethernet packet arrival tracing data. For the second case, the simulated data generated from two classes with unequal weights has a slightly high traffic intensity, barely high stationary variability, and a more dynamic pattern in its correlation. However, the second dataset is a smooth pattern. This is attributed to the combination of simple network traffic modeling and bursty traffic patterns, which leads to the following:

1) The scheduler algorithm is adaptive to make traffic patterns more stable with no bandwidth wastage.
2) In the networks, the fairness introduced by WFQ made scenarios of simulated data less irregular and more predictable about the arrival time of packets.
3) Regarding the statistical analysis of the first case, we use the tracing data of Ethernet packet arrivals and datasets generated by the WFQ algorithm, which was standard and had equal weights with the same arrival rate. It enhances and effectively manages bandwidth allocation.
4) According to the second case, the first data set was generated from the WFQ algorithm with unequal weights with priority, and the second dataset was generated from the same algorithm with equal weights for each class. The appropriately distributed (λ) significantly enhances the performance of the queues by providing balanced load distribution corresponding to the configured weights, decreasing queue variance, improving stability, reducing delay, and optimizing throughput.
5) Referring to queue management, each source never starves of service and ensures that the behavior of one source does not affect the other.

In summary, the simulated datasets were represented as abstract data from tracing data on Ethernet packet arrival. Tracing data of Ethernet packet arrival was handled in queues router without exploiting total bandwidth. On the other hand, in the same case of the first analysis, equal weights ensure fairness across all classes for balanced traffic handling. In the second case, the study of datasets with equal weights shows slightly more stability, descending variance, and better balance in handling traffic compared to the analysis datasets with unequal weights. The equal allocation of weights ensures that both queues share the bandwidth evenly, leading to more predictable and steady performance. Therefore, appropriately distributing the weights and arrival rates is essential for achieving optimal performance and stability in the WFQ algorithm. In scenarios where specific traffic needs prioritization, unequal weights are effectively managed and allocated bandwidth to accomplish the needs of higher-priority traffic, thus optimizing overall network performance. Additionally, the traffic of classes allocates the whole bandwidth proportionate to their weights. The WFQ algorithm ensures that each class is served according to its assigned weight, holding fairness.

VII. Conclusions

A thorough statistical analysis was conducted to comprehend the different traffic patterns and traffic intensity characteristics of the datasets used in this study—simulated and tracing data of Ethernet packet arrival. The created data's arrival rate closely matched the tracing data of Ethernet packet arrival, suggesting that real-world conditions were successfully emulated. However, examining the datasets' intensity trends and bursty traffic patterns revealed clear distinctions.

A comparison analysis utilizing a 10-second window size for traffic intensity calculation, as shown in Figure 7, significantly highlighted higher overall average traffic intensity in the tracing data of Ethernet packet arrival. In contrast to the more consistent traffic flow shown in the simulated data, the tracing data of Ethernet packet arrival's bursty traffic pattern—characterized by erratic bursts of fast data transmission—was more prominent. Figure 8 shows almost identical traffic intensity curves for the datasets. The traffic intensity values of the datasets generated from two classes with unequal weights are slightly higher than those for the datasets generated from two classes with equal weights.

Our suggested analysis approach of data traffic generated from two classes of the WFQ algorithm successfully overcame real-world data problems represented by these data's inherent complexity and unpredictability. Consequently, our approach is capable of enhancing network performance.

The results indicated that although some parts of real-world situations were successfully recreated in the simulated data, burst-like traits are absent from the simulated data. For a more accurate depiction of network dynamics, capturing irregular bursts of activity is crucial, as demonstrated by the observed bursty element in the tracing data of Ethernet packet arrival. Subsequent investigations could focus on improving simulation models to include bursty components, resulting in more thorough observations.

References

[1] A. M. Elnaka, Q. H. Mahmoud, and X. Li, "Simulation based comparative performance analysis of qos traffic scheduling using fair and delay adaptive scheduler (fdas) versus wfq and edf," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 916–923.

[2] O. J. Kigodi, K. Michael, and D. Machuve, "Review on network performance: Meaning, quantification and measurement," 2013.

[3] J. F. Shortle and M. J. Fischer, "Approximation for a two-class weighted fair queueing discipline," *Performance Evaluation*, vol. 67, no. 10, pp. 946–958, 2010.

[4] S. Nandhini, "Low latency weighted fair queuing for real time flows with differential packet dropping," *Indian Journal of Science and Technology*, vol. 8, no. 22, p. 1, 2015.

[5] J. Zhang, J. Tang, X. Zhang, W. Ouyang, and D. Wang, "A survey of network traffic generation," 2015.

[6] M. S. Mazhar, "Comparative study of wan services and technologies in enterprise business networks."

[7] A. Shahrabi, "Honours project (ccis) final report."

[8] C. Hollot and Y. Chait, "Nonlinear stability analysis for a class of tcp/aqm networks," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, vol. 3. IEEE, 2001, pp. 2309–2314.

[9] D. Efrosinin, V. Vishnevsky, and N. Stepanova, "Optimal scheduling in general multi-queue system by combining simulation and neural network techniques," *Sensors*, vol. 23, no. 12, p. 5479, 2023.

[10] H. H. Elkarash, N. M. Elshennawy, and E. A. Saliam, "Evaluating qos using scheduling algorithms in mpls/vpn/wimax networks," in *2017 13th International Computer Engineering Conference (ICENCO)*. IEEE, 2017, pp. 14–19.

[11] Z. Afzal, P. A. Shah, and K. M. Awan, "Optimum bandwidth allocation in wireless networks using differential evolution," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1401–1412, 2019.

[12] H. Attar, M. R. Khosravi, S. S. Igorovich, K. N. Georgievan, and M. Alhihi, "Review and performance evaluation of fifo, pq, cq, fq, and wfq algorithms in multimedia wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 6, p. 1550147720913233, 2020.

[13] D. Strzkeciwilk, R. Nafkha, and R. Zawi'slak, "Performance analysis of a qos system with wfq queuing using temporal petri nets," in *Computer Information Systems and Industrial Management: 20th International Conference, CISIM 2021, Elk, Poland, September 24–26, 2021, Proceedings 20*. Springer, 2021, pp. 462–476.

[14] S. Ghimire, G. B. Thapa, R. Ghinire, and S. Silvestrov, "A survey on queueing systems with mathematical models and applications," *American Journal of Operation Research*, vol. 7, no. 1, pp. 1–14, 2017.

[15] I. Zakariyya, M. N. A. Rahman, and M. N. Ismail, "Modelling the performance of class-based weighted fair queue using opnet," *IJNCAA*, p. 53, 2015.

[16] M. E. Mustafa and S. A. Talab, "The effect of queuing mechanisms first in first out (fifo), priority queuing (pq) and weighted fair queuing (wfq) on network's routers and applications," 2016.

[17] M. Koyuncu and A.-F. Hayder, "Comparison of scheduling algorithms for multimedia applications in ipv4 and ipv6," in *2015 9th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2015, pp. 418–422.

[18] S. Szilágyi and B. Almási, "A review of congestion management algorithms on cisco routers," *Journal of Computer Science and Control Systems*, vol. 5, no. 1, p. 103, 2012.

[19] M. D. Farzanegan, H. Saidi, and M. Mahdavi, "An approach to scheduling bursty traffic," *ETRI Journal*, vol. 36, no. 1, pp. 69–79, 2014.

[20] A. H. Rashid and S. S. Muhammad, "Traffic intensity based efficient packet schedualing," in *2019 International Conference on Communication Technologies (ComTech)*. IEEE, 2019, pp. 88–101.

[21] X. Jian, N. Jianguo, and H. Jinke, "An traffic intensity based media access control protocol with priorities," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 177–181.

[22] S. Molnár and G. Miklós, "On burst and correlation structure of teletraffic models," in *5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, 1997, pp. 21–23.

[23] T. A. Assegie and H. D. Bizuneh, "Improving network performance with an integrated priority queue and weighted fair queue scheduling," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 241–247, 2020.

[24] M. I. Bogachev, A. V. Kuzmenko, O. A. Markelov, N. S. Pyko, and S. A. Pyko, "Approximate waiting times for queuing systems with variable long-term correlated arrival rates," *Physica A: Statistical Mechanics and its Applications*, vol. 614, p. 128513, 2023.

[25] W. E. Leland and D. V. Wilson, "High time-resolution measurement and analysis of lan traffic: Implications for lan interconnection," in *IEEE INFCOM'91. The conference on Computer Communications. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings*. IEEE, 1991, pp. 1360–1366.

[26] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, 2010.

[27] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006.

[28] V. B. Iversen, "Teletraffic engineering and network planning," *Technical University of Denmark*, p. 270, 2010.

[29] P. Kumar, "Application of correlation-regression method to correlate the statistical relationship between the parameters of some indexed journals," *Transportation*, vol. 2, no. 1, p. 9.

[30] S. Khanam, I. Ahmedy, and M. Y. I. Idris, "Performance evaluation of weighted fair queuing model for bandwidth allocation," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 1*. Springer, 2021, pp. 175–183.

[31] D. A. Mahmood and G. Horváth, "A simple approximation for the response times in the two-class weighted fair queueing system," in *Analytical and Stochastic Modelling Techniques and Applications: 24th International Conference, ASMTA 2017, Newcastle-upon-Tyne, UK, July 10-11, 2017, Proceedings 24*. Springer, 2017, pp. 125–137.

[32] C. You, Y. Zhao, G. Feng, T. Q. Quek, and L. Li, "Hierarchical multiresource fair queueing for packet processing," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 726–740, 2022.

[33] E. F. Alnatsheh, "Queue scheduling algorithms: A comparative analysis," *International Journal of Innovative Computing*, vol. 10, no. 1, 2020.

[34] V. Kosenko, "Statistical analysis of data on the traffic intensity of internet networks for the different periods of time."

Mays Ali Mawlood received a B.Sc. Degree in Computer Engineering from the University of Technology, Iraq, in 2009. She is currently a M.Sc. student at the University of Technology, Department of Computer Engineering, Iraq. Her research interest is Computer Networks.

Dhari Ali Mahmood is currently a Lecturer at the Department of Computer Engineering, University of Technology, Iraq. Received a B.Sc. degree in computer engineering from University of Technology, Baghdad, Iraq, in 2005 and M.Sc. degree in computer engineering from GGSIP University, India in 2014. He obtained his Ph.D. in Computer Engineering from Budapest University of Technology and Economics, Hungary in 2021. Research interests mainly focus on Mobile Ad-hoc Network (MANET), Vehicular Ad-hoc Network (VANET), Queueing performance and Computer Networks.