

Research Article

Modeling of Wireless Traffic Load in Next Generation Wireless Networks

Sohaib Manzoor ¹, **Khalid Bashir Bajwa**,² **Muhammad Sajid**,¹ **Hira Manzoor** ³,
Mahak Manzoor,⁴ **Nouman Ali** ⁵, and **Muhammad Ilyas Menhas**¹

¹Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, AJK, Pakistan

²Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan

³School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

⁴Department of CS&IT, Mirpur University of Science and Technology (MUST), Mirpur 10250, AJK, Pakistan

⁵Department of Software Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, AJK, Pakistan

Correspondence should be addressed to Sohaib Manzoor; sohaib.ee@must.edu.pk

Received 8 June 2021; Accepted 17 August 2021; Published 31 August 2021

Academic Editor: Jugoslav Jokovic

Copyright © 2021 Sohaib Manzoor et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software defined WiFi network (SD-WiFi) is a new paradigm that addresses issues such as mobility management, load management, route policies, link discovery, and access selection in traditional WiFi networks. Due to the rapid growth of wireless devices, uneven load distribution among the network resources still remains a challenging issue in SD-WiFi. In this paper, we design a novel four-tier software defined WiFi edge architecture (FT-SDWE) to manage load imbalance through an improved handover mechanism, enhanced authentication technique, and upgraded migration approach. In the first tier, the handover mechanism is improved by using a simple AND operator and by shifting the association control to WAPs. Unauthorized user load is mitigated in the second tier, with the help of base stations (BSs) which act as edge nodes (ENs), using elliptic ElGamal digital signature algorithm (EEDSA). In the third tier, the load is balanced in the data plane among the OpenFlow enabled switches by using the whale optimization algorithm (WOA). Moreover, the load in the fourth tier is balanced among the multiple controllers. The global controller (GC) predicts the load states of local controllers (LCs) from the Markov chain model (MCM) and allocates packets to LCs for processing through a binary search tree (BST). The performance evaluation of FT-SDWE is demonstrated using extensive OMNeT++ simulations. The proposed framework shows effectiveness in terms of bandwidth, jitter, response time, throughput, and migration time in comparison to SD-WiFi, EASM, GAME-SM, and load information strategy schemes.

1. Introduction

WiFi networks have increased rapidly in the past decade due to their simple technical implementation and low costs [1, 2]. WiFi networks are optimized by formulating an explicit expression for the request-to-send (RTS) threshold using all key system parameters such as the number of nodes, window size, cutoff phase, and transmission rates [3]. Software defined network (SDN) is introduced to eliminate the traditional network problems which do not meet all the user requirements, such as routing policies, load balancing, and scalability [4]. Flexible and adaptive functionalities of SDN

support various developments in the past. In SDN the limitations of a single controller architecture are overcome by a multiple SDN controllers design [5, 6]. Potentials of these multiple SDN controllers include management efficiency, scalability, minimized latency, and robustness. Load balancing in multiple SDN controllers achieves scalability, delay reduction, and robustness.

The recent focus of research for the computer network community is on cloud computing, Internet of Things (IoT), vehicular ad hoc network (VoAH), wireless body area networks (WBANs), and software defined networks [7–10]. Intelligent software defined WiFi solution (ISD-WiFi) provides end-to-

end control and mobility management. The devices are linked via WAP, which manages load balancing and mobility [11]. The traffic pattern is predicted at WAP and decisions are made according to the movement of traffic. Switch migration in SDN is a promising approach to address the issue of load imbalance among the network resources. Increased traffic in the data plane is tolerated by switch migration [12]. Many decision-making algorithms are studied for switch migration. SD-WiFi monitors the traffic to maintain a certain threshold of load at WAP that is presented as RFlow+ [13]. Network monitoring is required on WAP, switches, and controllers to manage the incoming traffic. Workload at controllers is managed by incorporating a lightweight flow stealer approach that proceeds with an immediate action over the traffic [14]. Queues of data packets are maintained at each individual controller to detect load imbalance among the controllers. The load is balanced by shifting the data packets from the overloaded controller to the underloaded controller.

SDN is integrated with emerging technologies such as fog computing and Internet of things (IoT) to support a wide range of applications [15]. This integration enables connectivity with the edge devices supporting 3G, 4G, and WiFi. SDN based edge computing presents an adaptive transmission architecture which categorizes the data streams into normal and emergent classes for packet processing [16]. Optimal routing of packets enhances the network resource utilization. Signal-to-noise ratio (SNR) is used to identify the optimal path for routing [17]. Different streams of data flows are applicable in the SDN based edge computing architecture. SDN dependent solutions are developed in recent years enhancing the intelligence growth of decision-making criteria in wireless networks. The provisioning and adaptability of SDN introduce several advancements for load balancing.

Scalability in the control plane is achieved by performing load balancing based on the idea of hybrid switching [18]. It makes use of both SDN switches and traditional network switches. Each switch is enabled to measure the arrival of traffic loads. A controller is responsible for rerouting the traffic if any switch gets overloaded. Load balancing, that is, flow control of traffic arrival tends to maximize throughput and QoS [19]. The arrived packets are forwarded in accordance with their priority. The priority depends on the type of packets. Edge computing also supports the design of the load balancer [20]. The load balancer is executed on data plane which operates in a two-level structure.

However, the previous research works for load balancing have been discussed on core SDN, SDN with edge computing, and SDN with WiFi (SD-WiFi). Factors such as security, traffic priorities, load balancing in data and control plane, and optimal route policies are not studied in a hybrid software defined WiFi edge architecture. To tackle the aforementioned shortcomings, we develop a novel load balancing architecture that integrates WiFi, SDN, and edge computing to manage and balance load at each tier. The proposed FT-SDWE has the following contributions:

- (i) The handover mechanism is improved in the first tier by using a simple AND operator. Handover decisions are made by WAPs themselves.

- (ii) Unauthorized devices that enter into the network and contribute towards unnecessary load are mitigated in the second tier. Edge nodes are responsible for authenticating devices using the elliptic ElGamal digital signature algorithm.
- (iii) Load at data plane is balanced among the OpenFlow enabled switches using a whale optimization algorithm, which migrates packets from overloaded switch to an optimal switch. Switches are equipped to route real and non-real-time traffic patterns according to their port numbers.
- (iv) Markov chain model is applied to estimate load information of local controller. The fourth tier maintains a binary search tree in the global controller to allocate packets to a least loaded local controller. The global controller prioritizes packets according to packet types.

The rest of the paper is organized as follows: Section 2 describes the recent research works on SDN and SD-WiFi for load balancing. The section also illustrates the issues and limitations of previous research works. Problem definition is dealt with in Section 3. Section 4 explains the proposed system and the solution derived for the current issue. Section 5 discusses the simulation setup, FT-SDWE evaluation, and comparative results to report the performance evaluation of the proposed scheme. Finally, we conclude this paper and present the future research directions in Section 6. Table 1 shows the list of abbreviations used in this study.

2. Related Work

2.1. Load Balancing in SDN. A load balancing scheme based on server response time (LBBSRT) is presented in order to achieve higher scalability [21]. A single server is selected based on the minimum response time. The Packet_In messages from the users reach the controllers via switches. The user requests are then forwarded to servers from controllers via switches. Response time is estimated with respect to the difference between arrival time to that of the transmitted time of the data packets. Switches receive the packets from controllers and users at the same time and hence it increases the load dynamically.

The load information strategy method is used to balance the load among the multiple controllers in SDN [22]. Load at controllers is determined from the average message arrival rate. The switches are sorted in descending order with respect to the message arrival rate for migration. The overloaded controllers trigger the request to perform switch migration. However, the migration procedure is lengthier, leading to delayed decisions in switch migration. Switch migration is performed to mitigate unnecessary resource utilization. GAME-switch migration is presented on the idea of a game model [23]. A switch for migration is chosen in accordance with the utilization ratio of switches. The game-theoretic approach gives the best decision for migration, whereas it involves multiple complex mathematical computations and allows the arrival of numerous malicious requests into the network.

TABLE 1: List of abbreviations.

Abbreviation	Description
ACH	Adaptive connection handoff
AP	Access point
BS	Base station
BST	Binary search tree
EASM	Efficiency aware switch migration
EEDSA	Elliptic ElGamal digital signature algorithm
EN	Edge node
FTP	File transfer protocol
FT-SDWE	Four-tier software defined WiFi edge architecture
GC	Global controller
LBBSRT	Load balancing based on server response time
LC	Local controller
MCM	Markov chain model
NRT	Non-real-time
QoS	Quality of service
RSSI	Received signal strength indicator
RT	Real-time
SD-WiFi	Software defined WiFi network
SVM	Support vector machine
VoIP	Voice over IP
WOA	Whale optimization algorithm

Efficiency aware switch migration (EASM) is presented for load balancing in SDN [24]. EASM operates to provide fairness in load balancing by switch migration. Load at the controller is determined from routing overhead, state synchronization, and data interaction overhead. Load at the controllers is balanced using the trigger factor. The probability distribution is significant to decide the migration of a particular switch. Multiple mathematical formulations are used in this work which leads to complexity in the system.

2.2. Load Balancing in SD-WiFi. SD-WiFi requires mobility management support to balance the load among the IEEE 802.11 users. The seamless mobility of devices forces to use handover mechanisms. A scheme is proposed which uses the logical access points (L_{AP}) for providing continuous connectivity [25]. The received signal strength indicator (RSSI) is the only significant parameter estimated for making handover decisions. Traffic load at APs is determined and balanced. Single metric based handover decision is not efficient as it creates crowded hotspots [26]. Load balancing in SD-WiFi is achieved by flow prioritization and clustering of local controllers [27]. The packets are prioritized by an analytical hierarchical process and the high priority packets are processed first by the local controllers.

The two-tier architecture is presented to balance the load in SD-WiFi [28]. The controller computes the fairness index to identify the underloaded and overloaded access points. The status reports of the access points enable the controller to estimate the fairness index, load level, and control level. The fairness index is estimated from the load which includes CPU utilization, memory utilization, and average packet error rate. Load level is predicted from the maximum load with respect to the number of base stations. System-oriented parameters are given more consideration whereas the user-

oriented parameters are neglected. Type-2 fuzzy particle swarm optimization along with support vector machine (SVM) techniques are used to balance the load among multicontrollers in SD-WiFi [29]. The SVM differentiates between the high priority packets and low priority packets and forwards these packets to the control plane where the Markov chain model (MCM) predicts the future states of the local controllers and in case of any overloaded controllers, the packets are transferred to an underloaded controller, chosen by the fuzzy and particle swarm optimization techniques. Complex and long mathematical formulation contributes to the latency factor.

A load balancing scheme is proposed in SD-WiFi to optimize the load among the WAPs [30, 31]. An adaptive connection and hand-off (ACH) load balancing algorithm is used to bring fairness of load among the WAPs. Load is introduced by the network parameters, which are not discussed. Load balancing and traffic prioritization are achieved for SD-WiFi in healthcare [32]. Probability distribution is used to categorize healthcare traffic into emergency traffic, administrative traffic, and normal Internet traffic. Load among the OpenFlow enabled APs is optimized using SDN through a testbed [33]. The quality of service (QoS) is enhanced for access selection through throughput, packet loss rate, and received signal strength indicator (RSSI) values. The scalability of the control plane is not taken into account.

3. Problem Definition

Dynamic and adaptive load balancing (DALB) in SDN controllers is proposed over a distributed architecture [34]. DALB estimates the load using the number of flow table entries, average message arrival rate at each switch, and round trip time between the switch and the controller. Self-decisions are made by any of the controllers deployed. The absence of a centralized controller leads to a high load in the data plane. The artificial intelligence algorithm and optimization algorithm are also used to address the load imbalance issues in SDN [35, 36]. Backpropagation based artificial neural network (BP-ANN) is presented for load balancing. Features such as bandwidth utilization ratio, packet loss rate, transmission latency, and transmission hop are taken into account for preferring a switch to balance the load. Only the link-based parameters are considered to mitigate load at the switches.

Ant colony optimization algorithm is used to perform dynamic load balancing. Link bandwidth is considered as the only perimeter for load balancing. Load status and dynamic weight coefficients are used which lead to the construction of an additional load balancing plane [37]. Packets are prioritized and first-in-first-out (FIFO) procedure is followed for prioritizing similar packets resulting in higher processing time. Multiple SDN controller design is developed on the basis of hierarchical structure, which detects the loaded controllers and rearranges the controllers and switches every time after the allocation of packets [38]. SD-WiFi also focuses on minimizing the load at the APs in the data plane [39]. The controller is responsible for making decisions on

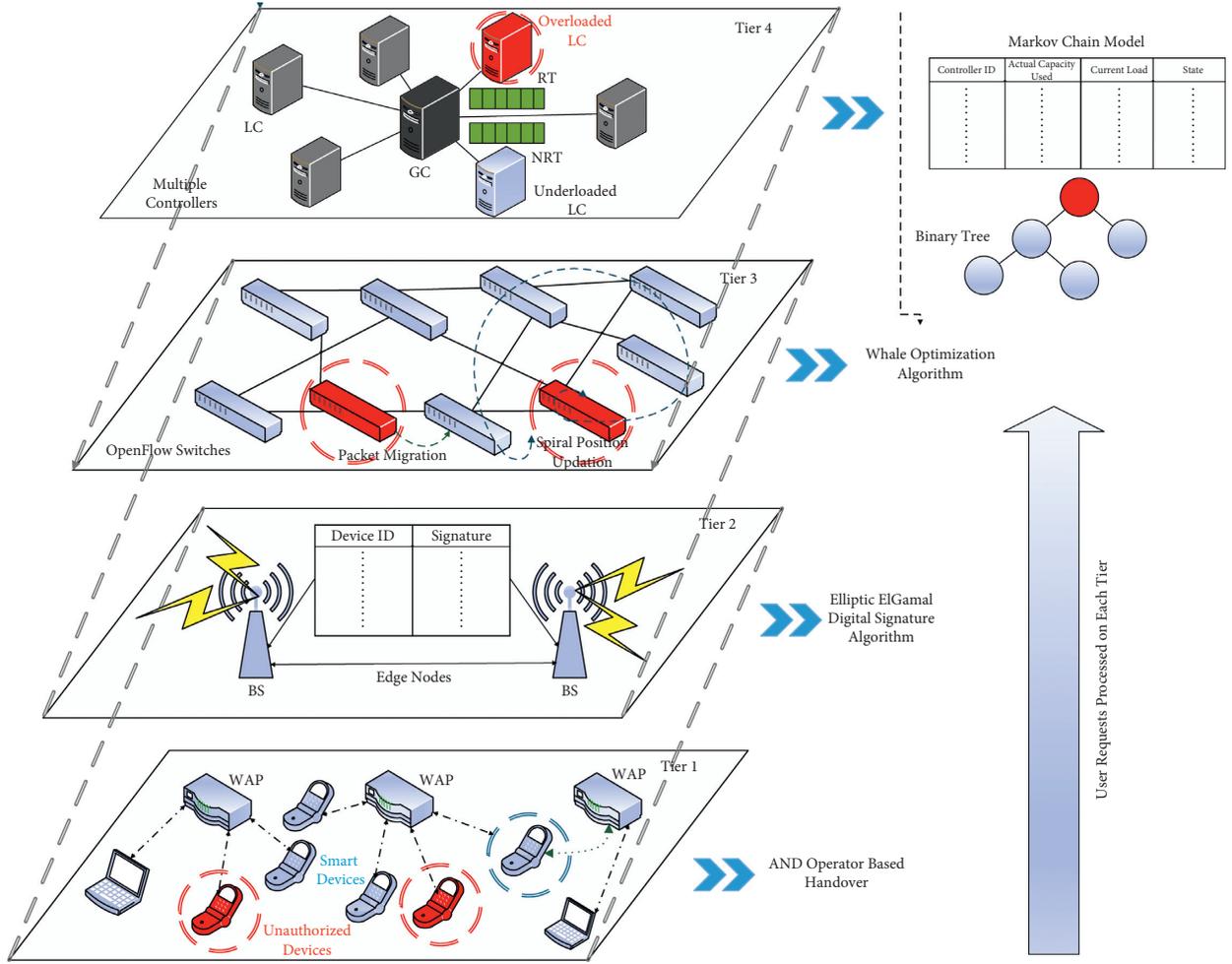


FIGURE 1: Four-tier software defined WiFi edge architecture.

handover which contribute towards significant jitter as unnecessary messages are exchanged between the controller and the OpenFlow enabled APs.

The major problems defined in previous research works considered SDN focusing on only a specific tier. Packet priorities, traffic patterns, flow table size, and routing methods are not utilized efficiently. Link and system parameters are considered separately. There is still a need for research to be conducted on load balancing at each tier considering the data plane and the control plane in SD-WiFi using the edge architecture.

4. Proposed System Model

The proposed FT-SDWE is developed in this paper to overwhelm the problem of load imbalance in each tier. FT-SDWE is a combination of SDN, WiFi network, and edge computing which is split into a four-tier architecture as shown in Figure 1. Load balancing is the major focus in each tier designed in FT-SDWE. Tier 1 is a WiFi network in which the wireless devices get access via WAPs. Tier 2 comprises edge nodes that are base stations (BSs) which authenticate devices to mitigate load from unauthorized devices. Tier 3 is a data plane with OpenFlow switches and tier 4 consists of

multiple SDN controllers. FT-SDWE is comprised of n dynamic mobile devices $Dv = \{d_1, d_2, d_3, \dots, d_n\}$, N WAPs, that is, $WAP = \{ap_1, ap_2, ap_3, \dots, ap_N\}$, k edge nodes $EN = \{bs_1, bs_2, bs_3, \dots, bs_k\}$, l switches as $S = \{sw_1, sw_2, sw_3, \dots, sw_l\}$, and m controllers $C = \{cn_1, cn_2, cn_2, \dots, cn_m\}$.

Table 2 depicts the major contributions of the proposed FT-SDWE for load balancing in the SDN designed environment. Load at each tier is significantly focused and minimized effectively. Figure 1 illustrates the proposed FT-SDWE design comprised of four tiers for load balancing. In tier 1, AND operator based handover decision is made to balance the number of wireless devices connected to each WAP, tier 2 reduces the load of unauthorized devices by authenticating authorized devices using EEDSA, tier 3 involves selecting an optimal switch using WOA subjected to overloading of flow entries. In tier 3, the packets are also classified into real-time (RT) and non-real-time (NRT) classes. The packets classified by switches in the data plane reach GC where they are placed into corresponding queues for further processing by LCs in the control plane. Finally, in tier 4, the GC monitors the load of each LC by MCM and allocates the packets to a least loaded controller using the binary search tree

TABLE 2: Contribution of load balancing in FT-SDWE.

FT-SDWE	Load balancing methods	Algorithm used
Tier 1 (mobile devices and WAPs)	Handover	AND operator based handover
Tier 2 (edge nodes)	Authentication	Elliptic ElGamal digital signature algorithm
Tier 3 (OpenFlow switches)	Packet migration	Whale optimization algorithm
Tier 4 (controllers, global and local)	Monitoring and allocation	Markov chain model and binary tree algorithm

algorithm. GC manages two queues for serving both RT and NRT packets from mobile devices. The RT packets are given higher priority as compared to NRT packets. Load balancing in individual tier exhibits better performance for different significant metrics.

4.1. Tier 1: Handover. Load is optimized among the WAPs by efficient handover in tier 1. An AND operator based handover is performed in tier 1. The handover decisions depend upon the received signal strength indicator (RSSI), distance, and mobility.

The association between WAPs and mobile devices takes place through probe messages or beacons. The mobile devices send the probe request message while in a scanning phase and the active WAPs already in the association phase reply with a probe response message. The connection takes place after a series of successful handshakes. At this point, the WAP may get overloaded as a large number of mobile devices connect to it. At this point, even the IEEE 802.11 standards do not include any mechanism to instruct the mobile device to make a handover under this loaded condition. In the proposed FT-SDWE, the inconvenience is overcome by an AND operator based handover algorithm. Each WAP maintains a logical truth table for a handover decision to be made. The RSSI, distance, and mobility values are updated periodically.

The truth table contains all the possible combinations of RSSI, distance, and mobility. Table 3 shows the parameters with the possible combinations leading to the corresponding handover decision.

AND operator based handover for mobile device d_1 is mathematically expressed as

$$HD_{w_{ap_1}} = R_i(d_1) \wedge D_t(d_1) \wedge M_b(d_1). \quad (1)$$

$HD_{w_{ap_1}}$ denotes the handover decision at w_{ap_1} for d_1 based on RSSI as $R_i(d_1)$, distance as $D_t(d_1)$, and mobility as $M_b(d_1)$ provided from AND operator \wedge . The binary high for handover means that the handover must take place and vice versa.

The binary values for each parameter specify low and high as 0 and 1, respectively. RSSI values are detected by the mobile devices and the WAP with the highest RSSI value is selected for association. The values of RSSI in the range of (-70 dBm to -90 dBm) result to be poor values which are given 0 binary value, whereas the RSSI values ranging (-30 dBm to -67 dBm) have higher efficiency hence classified with the binary value of 1.

The distance metric is defined between d and the wap. Distance is estimated using Euclidean distance formulation expressed as

TABLE 3: Tier 1 handover decision.

RSSI	Distance	Mobility	Handover decision
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$D_t(wap, d) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (2)$$

The wap and d are defined into x, y coordinates such as (x_1, y_1) and (x_2, y_2) , respectively. The measured distance between 1 m and 10 m is provided high binary value as 1 and distance more than 10 m is associated with a low binary value of 0.

Mobility of the device is important in determining the decision for handover. Slow-moving devices are given low binary values such as 0 and fast-moving devices are given high binary values of 1. Mobility plays a key role in WiFi networks since the devices are wireless and are enabled to connect to different WAPs.

Using equation (1), the handover time is reduced so that the load is optimized quickly among the WAPs. In the phase of active scanning, the mobile device sends a probe request frame to all available WAPs, and hence the WAPs reply with the probe response frame saving the MAC address of the mobile device which sent the probe request frame, for future communication. Let us suppose the mobile device d_1 connects to w_{ap_1} . Through the periodic update of the truth table in all WAPs, a handover decision is made. The w_{ap_1} dissociates d_1 and hands it over to any nearby wap when the RSSI value of w_{ap_1} detected by d_1 is low and the d_1 is far away from w_{ap_1} . The proposed FT-SDWE focuses mainly on dynamic mobile devices which have a mobility rate from 1 m/s to 20 m/s.

4.2. Tier 2: Authentication. Tier 2 comprises edge devices, that is, BSs which are responsible for authenticating the incoming packets from a particular mobile device. EEDSA is utilized for authentication. Tier 2 is constructed as a bridge between the WiFi network and SDN to minimize the load which is caused by the participation of unauthorized devices in the network. The digital signature is provided to authenticate the mobile devices. Most of the unauthorized mobile devices enter into the network as freeloaders for degrading the network performance.

ElGamal's method of key distribution consists of three phases including the setup phase, the encryption phase, and the decryption phase. The setup phase is responsible for public key sharing and the remaining two phases help in the successful delivery of information between two entities making it difficult for the intruders to play their role. Digital signature algorithms have been widely employed in the cryptographic research areas as they ensure data integrity, nonrepudiation, and data origin authentication. Elliptic curve cryptography is a very powerful mathematical tool used for encryption and decryption services. It uses a shorter encryption key which means less storage and faster computation. Reverse engineering is also impossible with large values involved due to the computational complexity of logarithm problems. Tier 2 uses the power of ElGamal digital signature and elliptic curve techniques for authentication of authorized mobile devices. EEDSA has three basic phases for its successful operation which are: 1) key generation; 2) signature generation, and signature verification.

Keys are generated for the signing and verification of a particular device. A user using a mobile device d_1 chooses a random number R_n , where R_n ranges $[1, N - 1]$ and N is the order of the curve. $R_n = P_k$, where P_k is the private key. The public key is generated from the multiplication of R_n and a point on the elliptic curve. A point doubling method is used to find the required point. Let (P, Q) be the two points present on the elliptic curve C . Assume that $P = (x_p, y_p)$, $Q = (x_Q, y_Q)$ where $P, Q \neq \infty$. The value of $P + Q$ is expressed as $Z = (x_Z, y_Z)$, so does the point doubling method yield expressed as $2P = Z = (x_Z, y_Z)$ which is algebraically expressed into

$$x_Z = \left(\frac{3x_p^2 + A}{2y_p} \right)^2 - 2x_p \pmod{p}, \quad (3)$$

$$y_Z = \left(\frac{3x_p^2 + A}{2y_p} \right) (x_p - x_Z) - y_p \pmod{p}. \quad (4)$$

The term p denotes a prime number and A is the integer obtained from the curve equation $y^2 = x^3 + 4x + 20$. The public key curve point is therefore $Q_A = P_k * Z$.

Signatures are generated when d_1 first selects the digest value z for the request it is willing to make. The digest value must be of the same bit size as N . d_1 then uses a random number u between 1 and N . To find the point U on the curve, u is multiplied by the generator point P . The point U results in two coordinate values where the x coordinate of U represents the first half of the signature represented as r and r serves as a reference value used to verify the signature. d_1 then finds the other half of the signature value using equation (6). The signature generated is denoted as (r, s) , where

$$r = (x_Z) \pmod{N}, \quad (5)$$

$$s = u^{-1} (z + rR_n) \pmod{N}. \quad (6)$$

If r is not equal to 0, that is, $r \neq 0$, then cryptographic hash function for device d_1 is given as

$$e = H(d_1). \quad (7)$$

The ENs are responsible for verifying the signature (r, s) generated through equations (5) and (6) and also ensure that the range for r and s is between 1 and $N - 1$; otherwise, the signature remains invalid. e is calculated using the same HASH function used in signature generation. The value w is calculated as $w = s^{-1} \pmod{N}$. Two multipliers j_1 and j_2 are calculated where $j_1 = zw \pmod{N}$ and $j_2 = rw \pmod{N}$. Finally j_1 is multiplied by P and j_2 is multiplied by Q_A . Summing up these values, we calculate S having the same x coordinates as r .

If the generated signature for device d_1 is $\{d_1(r, s)\}$ and signature at bs_1 is similar, then the authentication is successful; otherwise, access will be denied. In the proposed FT-SDWE, the ENs are assumed to maintain the signatures of each legitimate mobile device and hence it matches the signature in accordance with the user identity. The procedure of EEDSA is presented in Algorithm 1.

In tier 2, authentication is performed to minimize the unnecessary load from unauthorized devices. ENs are assumed to be trusted and hence are deployed to mitigate the load from illegitimate devices.

4.3. Tier 3: Packet Migration. Tier 3 is the data plane which is comprised of OpenFlow switches. The switches can become overloaded anytime due to the increased amount of packets arrival. The loaded and underloaded switches are enabled to classify the incoming traffic into two categories such as real-time traffic and non-real-time traffic. Each packet is verified with respect to data type, deadline, protocol, and port numbers. A legitimate packet should match all the fields and only after this, the packets are forwarded to the next tier. Once the fields are matched, the packets are classified and directed to the control plane.

OpenFlow switches have the capability of pushing the packets towards the controllers and they can also discard the packets if the flow rules are not matched. Each individual packet entering into the data plane is supposed to have matching fields, flow priorities, counters, and processing instructions. Switches only have limited flow entries; hence, intolerable packet flows lead towards an overloaded switch causing an immediate packet migration to take place to an optimal switch. Table 4 depicts the properties of traffic used and the processing done by the switches. The incoming packet information is matched with the fields stored in the OpenFlow switch. If the information matches and the packet verification is confirmed, the desired action is taken by the switch. The proposed FT-SDWE is capable of supporting both real-time traffic such as voice over IP (VoIP) and non-real-time traffic such as file transfer protocol (FTP).

If the OpenFlow switch is filled with flow entries, then the WOA comes into play for selecting an optimal switch to migrate the packets. The selected optimal switch either is underloaded or has a partially filled flow table. WOA defines the current best solution as optimal or nearly optimal. [40]. The following equations represent the behavior of whales, that is, packets,

```

(1) Begin
(2) Generate  $P_k$ 
(3) Compute  $Z = (x_Z, y_Z)$  from equations (3) and (4)
(4) Estimate  $r$ 
(5) if ( $r = 0$ ) then
(6)   go to step 1
(7) else
(8)   Compute  $s$  and  $e$  from equations (6) and (7);
(9) end if
(10)  $d_1 \rightarrow \{d_1(r, s)\}$  to  $bs_1$  // Signature sent to EN
(11)  $bs_1$  verifies  $\{d_1(r, s)\}$ 
(12) if ( $(r, s)d_1 = (r, s)bs_1$ ) // Verifying the signature then
(13)   Access granted
(14) else
(15)   Access denied
(16) end if
(17) End.

```

ALGORITHM 1: Elliptic ElGamal digital signature algorithm.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|, \quad (8)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}. \quad (9)$$

From equations (8) and (9), \vec{A} and \vec{C} are computed from the following formulation:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad (10)$$

$$\vec{C} = 2\vec{r}. \quad (11)$$

The terms \vec{A} and \vec{C} denote the coefficient vectors, \vec{X} is the position, X^* is the best solution, and the value of a in equation (10) decreases gradually, whereas r denotes a random vector value ranging $[0, 1]$. For updating the positions, spiral updating is performed, which is capable of updating the positions of the surrounding switches. Spiral updating is handled by estimation of distance metric between the current switch and its neighboring switch. Let (X, Y) and (X^*, Y^*) be the position of the currently overloaded switch and neighboring switch, respectively. The spiral equation for distance estimation is expressed as

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi q) + \vec{X}^*(t). \quad (12)$$

From equation (12), b is a constant that defines the logarithmic spiral, q is a random number ranging $[-1, 1]$. The fitness value of each neighboring switch is predicted from the following significant parameters as the number of packets in buffer, packet loss rate, latency, number of flow entries, and number of packets processed in history. All the parameters considered for the evaluation of fitness function are significant, based on which the switch load is effectively identified. Data plane in SDN is one of the major areas that causes load imbalance due to several technical reasons including malicious packets involvement.

The number of packets in buffers N_{bf} is defined as the remaining number of packets that are yet to be processed and are still in the waiting state. Packet loss rate P_{ls} at switch S_l is determined from the number of packets transmitted and cumulative packets received. Mathematically, P_{ls} is expressed as

$$P_{ls}(S_l) = \frac{N_{Txp}(S_l) - N_{Rxp}(S_l)}{N_{Txp}(S_l)}. \quad (13)$$

Packet loss P_{ls} is determined from the number of packets transmitted N_{Txp} by the switch and the number of packets received N_{Rxp} by the switch. Here, N_{Rxp} is the cumulative packets received by the switch. N_{Rxp} is expressed as

$$N_{Rxp}(S_l) = \sum_0^n (Dv(Rxp)) + \sum_0^l (S(Rxp)). \quad (14)$$

The cumulative number of received packets is estimated from $Dv(Rxp)$ that denotes the total number of packets received from devices such as edge nodes and $S(Rxp)$ denotes the total number of packets received from overloaded switches. The latency of individual switch is formulated as follows:

$$L(S_l) = \frac{B_{tx}}{T_{xR}}. \quad (15)$$

Latency at switch S_l is estimated from the ratio of the number of bytes transmitted B_{tx} and the transmission rate T_{xR} . The total number of flow entries is represented as N_{fe} , which defines the packets arrived and considered for processing. The number of packets processed in history N_h is the packets processed and transmitted to the control plane. The fitness value is dependent on the aforementioned parameters such as N_{bf} , P_{ls} , L , N_{fe} , and N_h .

The value of the fitness function determines the optimal switch for migration. WOA is used due to the power of the spiral position updating mechanism. The immediate

TABLE 4: Properties of traffic.

Real-time traffic	
Data type	VoIP
Deadline	Low
Protocol	UDP
Port number	5060, 5061, 5062, 5063, 5064, 5065
Action	Accept or deny packets
Non-real-time traffic	
Data type	FTP
Deadline	Medium or high
Protocol	TCP
Port number	20 and 21
Action	Accept or deny packets

migration of packets minimizes the latency and ensures quick load balancing in the data plane.

Figure 2 illustrates the working flow of the proposed WOA in tier 3 for load balancing among the switches. Population refers to the number of switches participating. The process is repeated until an optimal switch is determined depending on the fitness value. Tier 3 is supported to prefer the best switch for migration. The migration process is performed only if the switch is occupied with more numbers of flow entries.

4.4. Tier 4: Monitoring and Flow Allocation. Tier 4 is comprised of GCs and LCs. The GC plays a major role in monitoring LCs load based on MCM. By the load information of LCs, the GC generates a binary search tree which is used to assign packets to LCs that arrive from the mobile devices. The GC maintains two queues one for real-time traffic and one for non-real-time traffic. Real-time traffic (VoIP) is given priority in processing as compared to non-real-time traffic (FTP). MCM is used to identify the states of LCs. States are defined as the load at the individual controller. LCs are categorized as underloaded LC and overloaded LC. The incoming packets from the data plane are allocated to the underloaded LC based on the decision of GC which maintains the current state of each individual LC present in the controller plane. Figure 3 demonstrates the MCM used in the proposed FT-SDWE, to determine the current states of the LCs.

MCM is enabled to detect all the possible states based on the identification of LCs transitions. MCM is also applicable to detecting future load states. In this work, four transitions are possible as represented in Table 5.

In MCM, the arrived requests, that is, packets from the data plane are modeled using the Bernoulli p distribution, where X_s represents the state of a particular LC after completion of s number of transitions. Initially, the LC is in the state i at time $t = 0$. The transition probability $Ts_{i,j}$ is expressed as

$$Ts_{i,j} = P(X_s = j | X_0 = i). \quad (16)$$

X_0 represents the initial state of the controller and j denotes the final state, that is, either underloaded or overloaded. The single transition probability $Ts_{i,j1}$ is given as

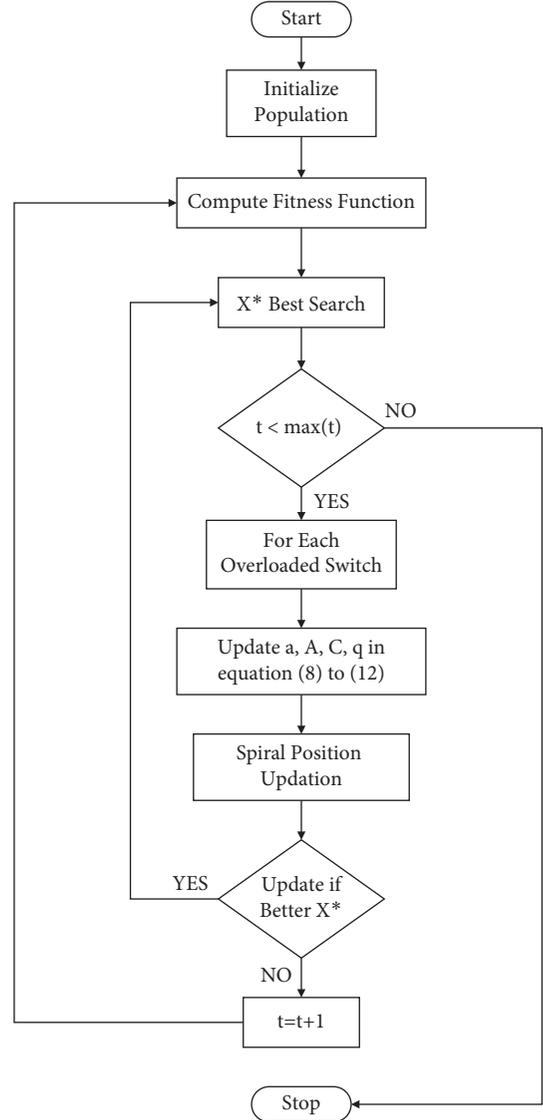


FIGURE 2: Whale optimization algorithm flowchart.

$$Ts_{i,j(1)} = P_{ij}. \quad (17)$$

If the transition probability of the controller stays in the same state after a certain time period, then its state is expressed as

$$Ts_{ii} = 1. \quad (18)$$

On periodical updating of load states in GC, it is efficient to support load management in the control plane. Using the load states of the LCs, the GC creates a BST. An incoming packet is analyzed in GC and is assigned to the available least loaded LC for further processing.

The probability values range [0-1], which lead to the construction of the BST. The BST is initiated with the root node, that is, parent and followed by left and right child nodes. The MCM results show that the load values below 0.5 represent an underloaded LC and the load values above 0.5 represent an overloaded LC. For instance, a set of sample

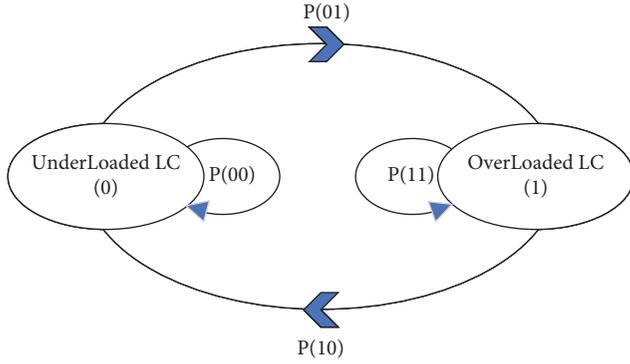


FIGURE 3: Markov chain model used in the proposed FT-SDWE.

TABLE 5: Transitions in the Markov chain model.

Load status	Probability	Possible transitions
Underloaded LC	$P(00)$	To underload
	$P(01)$	To overload
Overloaded LC	$P(11)$	To overload
	$P(10)$	To underload

values is considered as $\{0.4, 0.6, 0.7, 0.2, 0.5, 0.3, 0.1\}$ and a BST is constructed in Figure 4. The BST follows in-order traversal in which the minimum loaded LC is assigned with packets first followed by the ascending order of load values as shown by the dotted arrow lines in Figure 4. The least loaded LC having a load value of 0.1 is assigned for packet processing first, next comes the LC with a load value of 0.2, and so on. BST aims to direct the packets towards an underloaded LC first and simultaneously the load at an overloaded LC is processed.

FT-SDWE is supported to balance load at each tier and finally attains better performance than the previous research works.

5. Performance Evaluation

This section is composed of three subsections as simulation setup, FT-SDWE evaluation, and comparative results. The sections discuss the effective achievements of the proposed FT-SDWE after being tested on simulation software.

5.1. Simulation Setup. FT-SDWE is developed in OMNeT++ 4.6 simulator and JDK 1.8 that is installed on Windows 10 (64-bit). The proposed scheme is designed with OpenFlow switches, global controller, local controllers, wireless access points, edge nodes, and mobile devices. INET is employed in OMNeT++ supporting the OpenFlow standards. POX controller is used to performs load balancing in the control plane and data plane.

The significant parameters involved to develop FT-SDWE are listed in Table 6. The proposed scheme is not limited to only these parameters.

The simulation topology of FT-SDWE designed in OMNeT++ is depicted in Figure 5. The data packets from 80 mobile devices are aggregated and processed at WAPs.

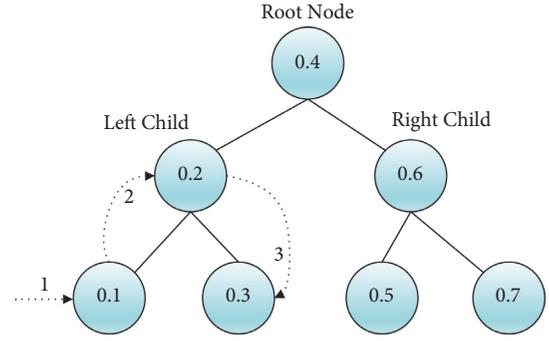


FIGURE 4: Binary search tree in GC.

Using OMNeT++, a four-tier setup is made and load balancing is applied over each tier as discussed in Section 4. Based on the simulation topology, the performance of FT-SDWE is evaluated in the next section.

5.2. FT-SDWE Evaluation. In this section, we report the balanced load at controllers for three time periods T_1 , T_2 , and T_3 . Table 7 illustrates the controller ID and its corresponding load which is obtained from MCM. The following entities are managed by the global controller for constructing a binary tree to balance the incoming data packets.

The load of each LC is initially calculated using its CPU, memory, and disk values. While these entities are being used by the LC during the packet processing and according to the number of packets being processed, the load values of LCs vary. The capacity of each LC is defined as

$$L(C_i) = \sum_{i=1}^m (C_{i(\text{CPU})}, C_{i(\text{Mem})}, C_{i(\text{Disk})}). \quad (19)$$

Let m be the total number of controllers; $C_{i(\text{CPU})}$, $C_{i(\text{Mem})}$, $C_{i(\text{Disk})}$ are the CPU, memory, and disk capacity of controller i , respectively. In this evaluation, the number of incoming packets is varied in each time period such that, in T_1 , nearly 2000 packets arrive per second; in T_2 , 4000 packets arrive per second; and T_3 deals with 6000 packets per second. However, in FT-SDWE, if the controllers are overloaded initially, they are still capable enough of processing the packets. The overloaded controllers are taken into consideration by BST which helps in load balancing among all LCs in the control plane.

5.3. Comparative Results. This section discusses the comparative analysis of the proposed FT-SDWE to the software defined WiFi networks (SD-WiFi) load balancing method [39], efficiency aware switch migration (EASM) [24], GAME-switch migration [23], and load information strategy [22]. Better performance is achieved due to the impact of the proposed algorithms in each tier. The major parameters that are studied to show the superiority of the proposed scheme are bandwidth, jitter, migration time, response time, and throughput.

TABLE 6: Simulation specifications.

Parameters	Range/value
Simulation area	800 m*800 m
Number of mobile devices	80
Number of OpenFlow switches	10
Network environment	Number of GCs 1
	Number of LCs 4
	Number of edge nodes 2
	Number of WAPs 8
Transmission rate of WAPs	100 Mbps
Traffic	VoIP FTP
Device mobility	1–20 m/s
Mobility model	Rectangle mobility
Packet transmission interval	0.5 s
Packet size	56 bytes
Simulation time	30 minutes

5.3.1. Bandwidth. Bandwidth plays a major role in wireless communication. In the proposed FT-SDWE, the wireless mobile devices connecting to WAPs have an impact on bandwidth, as they have to perform handover in order to balance the load. Higher bandwidth leads to quicker transfer of higher amount of data between two entities. Bandwidth also depends upon the capacity of the link.

In SD-WiFi, the handover was performed only after the controller decision was made. Multiple messages were exchanged before performing the handover. Handover decisions lead to higher delays which caused higher bandwidth consumption. Figure 6 demonstrates the comparison of bandwidth for two cases which are after handover and before handover. In both cases, FT-SDWE performs better than SD-WiFi as the association decisions are made by the WAPs, hence reducing the extra overheads caused by unnecessary message exchange between switches and the controller. The bandwidth before handover is approximately 20–30% higher in FT-SDWE, whereas the bandwidth after the handover is 10–15% higher than SD-WiFi. SD-WiFi is tested with only 4 mobile devices whereas 80 mobile devices are used in FT-SDWE and still the FT-SDWE performance supersedes SD-WiFi in terms of high-density networks.

5.3.2. Jitter. Jitter is defined as the variation of the time delay occurring between the reception of the packets. For instance, assume a sequence of packets being transmitted over a channel, jitter will be caused due to some disruption delay between the two continuous packets. Heavy jitter in a network tends to increase the degradation of network parameters. Jitter is a significant parameter that is used to analyze the handover algorithm. Measurement of jitter is essential during handover, as the packets being forwarded from one WAP to another will be altered.

The best handover algorithm guarantees minimum occurrence of jitter during the handover. In FT-SDWE, handover is performed for load balancing at tier 1.

Jitter parameter is measured in two cases, that is, before handover and after handover as shown in Figure 7. This metric is plotted with respect to simulation time, in which

the jitter in FT-SDWE is minimum as compared to SD-WiFi. In SD-WiFi, the longer time taken for handover decision increases jitter, whereas in FT-SDWE, the WAPs perform simple AND operation for handover decisions. Before and after the handover, FT-SDWE shows reduced jitter for 80 devices.

5.3.3. Migration Time. This metric is measured to analyze the amount of time consumed to transfer the packets from an overloaded switch to an underloaded switch. EASM was proposed to balance the load in SDN by performing switch migration. In EASM, migration time increases due to the use of multiple mathematical formulations. Table 8 presents the migration time for FT-SDWE compared to two previous research works which are EASM and GAME-SM.

FT-SDWE is specially designed to solve the issue of load balancing. In this work, tier 3 concentrates on load balancing via packet migration. Migration time is minimized by selecting an optimal switch to which the packets are migrated. Without selecting an optimal switch it is complex to migrate packets and higher time is consumed for migration. FT-SDWE reduces the migration time by using WOA, which ensures the selection of an optimal switch for packet migration. This algorithm is enabled to tolerate a huge number of packets.

5.3.4. Response Time. The response time for the proposed FT-SDWE is compared to EASM and load information strategy as shown in Figure 8. The reduction of response time intimates the better performance of the proposed method.

Response time should be minimized as much as possible to enhance the network performance. Response time depends on the performances of the algorithms involved in the network. FT-SDWE exhibits less response time as compared to EASM and load information strategy methods. Increased mathematical computations consume higher time and resources which tend to increase the response time. As the number of controllers increases, the number of packets also

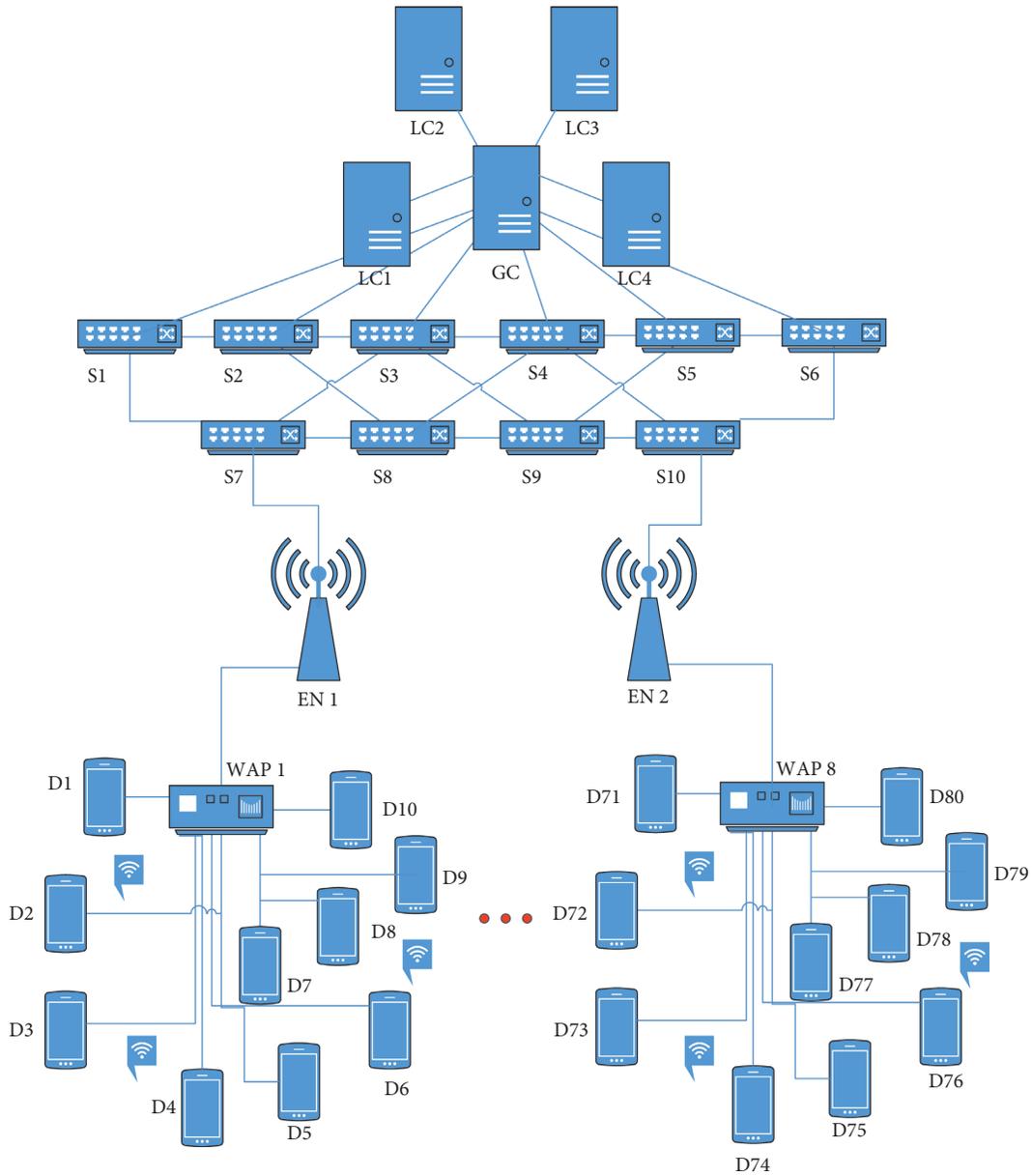


FIGURE 5: FT-SDWE simulation topology.

increases gradually as 1000 packets per second, 2000 packets per second, 4000 packets per second, and 6000 packets per second. From the result, it is clear that approximately 10–15% of response time is minimized in the proposed FT-SDWE as compared to EASM and load information strategy method.

5.3.5. Throughput. The comparative analysis includes the parameter throughput which is one of the most significant metrics to be discussed in any type of network evaluation. Throughput is defined as the measure of the total number of packets received in accordance with the total number of

packets transmitted. Figure 9 demonstrates the comparative analysis for throughput which compares FT-SDWE with EASM and load information strategy methods.

Due to higher mathematical computations and improper load balancing methods, throughput decreases, intimating the degradation of the network performance in EASM and load information strategy methods. On increasing the number of controllers, the throughput changes. As the number of controllers increases, the throughput is also enhanced due to the arrival of increased numbers of packets. Load balancing at each tier is supported to achieve higher throughput, due to the quick processing of packets arriving into the network.

TABLE 7: Global controller in tier 4.

Time period	Controller ID	Actual capacity used (%)	Current load	Status
T1	LC1	40	0.4	Underloaded
	LC2	70	0.7	Overloaded
	LC3	20	0.2	Underloaded
	LC4	10	0.1	Underloaded
T2	LC1	50	0.5	Overloaded
	LC2	40	0.4	Underloaded
	LC3	30	0.3	Underloaded
	LC4	40	0.4	Underloaded
T3	LC1	30	0.3	Underloaded
	LC2	60	0.6	Overloaded
	LC3	40	0.4	Underloaded
	LC4	20	0.5	Overloaded

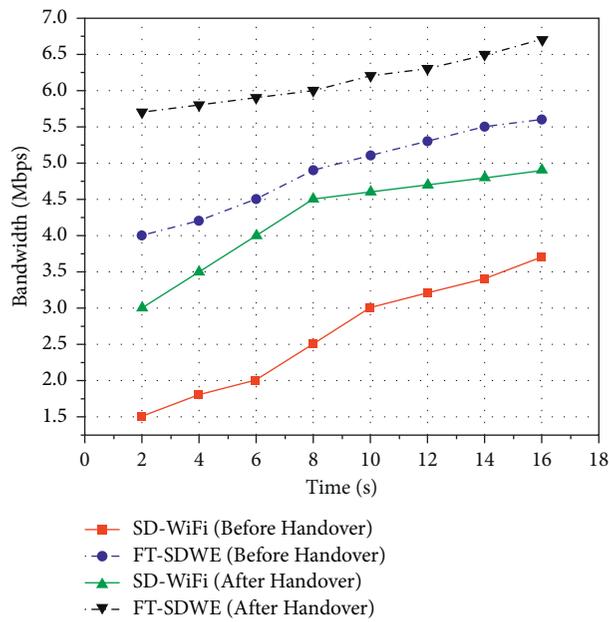


FIGURE 6: Performance of bandwidth.

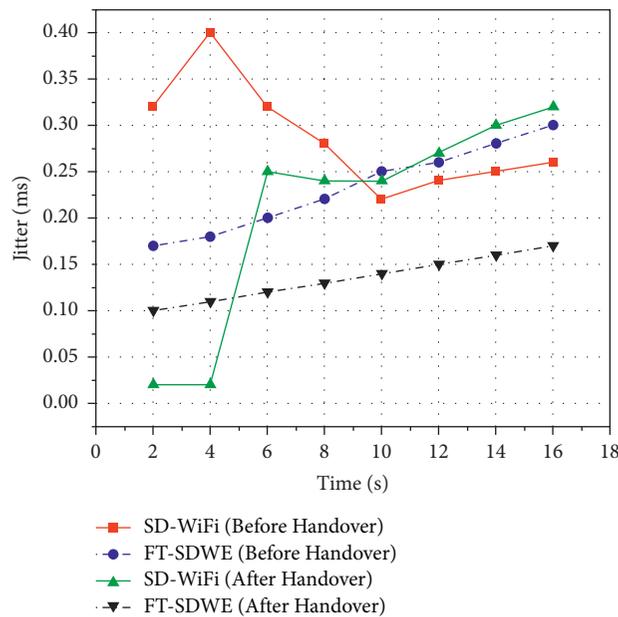


FIGURE 7: Performance of jitter.

TABLE 8: Migration time comparison.

Method used	Migration time
EASM	13 s
GAME-SM	19 s
FT-SDWE	61 ms

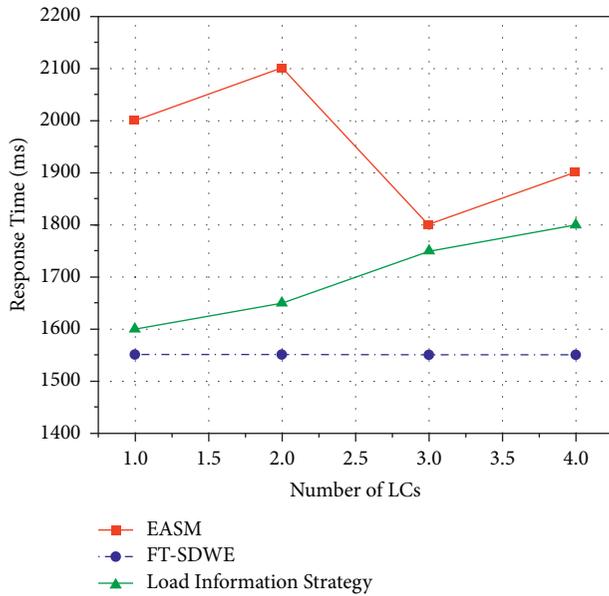


FIGURE 8: Response time.

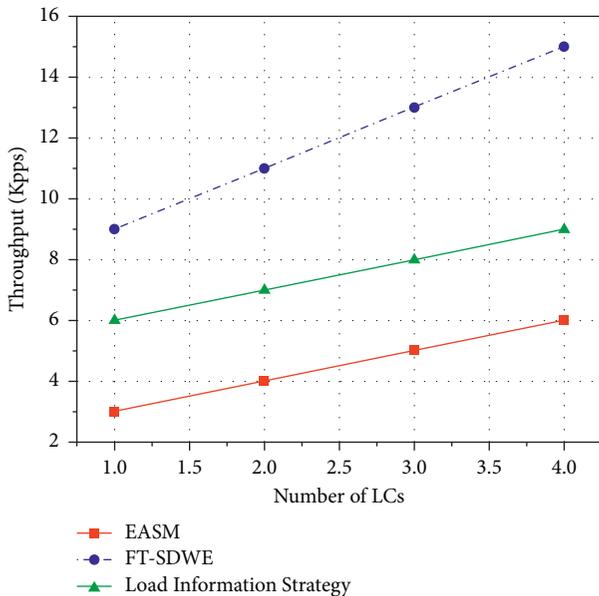


FIGURE 9: Performance of throughput.

6. Conclusion

In this paper, a novel FT-SDWE is proposed for effective load balancing. FT-SDWE is comprised of four tiers with WAPs, edge nodes, switches, and controllers. Many previous research works have been studied, leading to the issue that load balancing is only focused on any tier and plane.

However, in the proposed scheme, the load is balanced on all planes and tiers resulting in better performance attributes. Individual tiers in the proposed FT-SDWE use algorithms such as AND operator based handover, elliptic ElGamal digital signature algorithm, Whale optimization algorithm, Markov chain model, and binary tree search algorithm. Tier 1 balances the load by improving the handover process, tier 2 authenticates the devices to mitigate unnecessary load from unauthorized devices, tier 3 migrates the packets among switches to manage load imbalance, and tier 4 maintains the list of load of all LCs and provides efficient packet allocation. Load balancing in each tier tends to achieve higher performance when compared with previous research works. The proposed method outperforms SD-WiFi, EASM by, GAME-SM, and load information strategy methods in bandwidth, jitter, migration time, response time, and throughput.

In future, FT-SDWE performance is planned to be evaluated in a large-scale environment through a software/hardware codesign. Further, the execution time will be measured on an individual tier using hybrid algorithms to increase the SD-WiFi efficiency in high-density environments.

Data Availability

The data that support the findings of this study are openly available in Achieving Load Balancing in High Density Software Defined Wi at <http://cloud.eic.hust.edu.cn:8084/~sohaib/>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Manzoor, C. Zhang, X. Hei, and W. Cheng, "Understanding traffic load in software defined WiFi networks for healthcare," in *Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1-2, IEEE, Yilan, Taiwan, May 2019.
- [2] S. Manzoor, H. Manzoor, M. Manzoor, and A. Mahmood, "A load balancing mechanism for a multi-controller software defined WiFi network," in *Proceedings of the International Conference on Artificial Intelligence and Security*, pp. 3-14, Springer, Dublin, Ireland, July 2021.
- [3] S. Manzoor, Y. Yin, Y. Gao, X. Hei, and W. Cheng, "A systematic study of IEEE 802.11 DCF network optimization from theory to testbed," *IEEE Access*, vol. 8, pp. 154114-154132, 2020.
- [4] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network monitoring in software-defined networking: a review," *IEEE Systems Journal*, vol. 12, no. 4, 2018.
- [5] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, 2017.
- [6] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: a survey," *IEEE Access*, vol. 6, pp. 15980-15996, 2018.
- [7] S. Kanwal, Z. Iqbal, A. Irtaza et al., "Head node selection algorithm in cloud computing data center," *Mathematical*

- Problems in Engineering*, vol. 2021, Article ID 3418483, 12 pages, 2021.
- [8] K. Afzal, R. Tariq, F. Aadil et al., "An optimized and efficient routing protocol application for IoV," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9977252, 32 pages, 2021.
 - [9] S. Fatima, N. A. Aslam, I. Tariq, and N. Ali, "Home security and automation based on internet of things: a comprehensive review," *IOP Conference Series: Materials Science and Engineering*, vol. 899, Article ID 012011, 2020.
 - [10] N. Bilandi, H. K. Verma, and R. Dhir, "AHP-neutrosophic decision model for selection of relay node in wireless body area network," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 222–229, 2020.
 - [11] D. Tu, Z. Zhao, and H. Zhang, "Isd-wifi: an intelligent sdn based solution for enterprise wlangs," in *Proceedings of the 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1–6, IEEE, Yangzhou, China, October 2016.
 - [12] C. A. Wang, B. Hu, S. Chen, D. Li, and B. Liu, "A switch migration-based decision-making scheme for balancing load in sdn," *IEEE Access*, vol. 5, pp. 4537–4544, 2017.
 - [13] R. Jang, D. Cho, Y. Noh, and D. Nyang, "Rflow+: an sdn-based wlan monitoring and management framework," in *Proceedings of the INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, Atlanta, GA, USA, May 2017.
 - [14] P. Song, Y. Liu, T. Liu, and D. Qian, "Flow stealer: lightweight load balancing by stealing flows in distributed sdn controllers," *Science China Information Sciences*, vol. 60, no. 3, Article ID 032202, 2017.
 - [15] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for iot," *Wireless Personal Communications*, vol. 92, no. 1, pp. 181–196, 2017.
 - [16] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in sdn-based industrial internet of things with edge computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, 2018.
 - [17] A. Hakiri, B. Sellami, P. Patil, P. Berthou, and A. Gokhale, "Managing wireless fog networks using software-defined networking," in *Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1149–1156, Hammamet, Tunisia, October 2017.
 - [18] H. Xu, H. Huang, S. Chen, G. Zhao, and L. Huang, "Achieving high scalability through hybrid switching in software-defined networking," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 618–632, 2018.
 - [19] Y. Lu, B. Fu, X. Xi, Z. Zhang, and H. Wu, "An sdn-based flow control mechanism for guaranteeing qos and maximizing throughput," *Wireless Personal Communications*, vol. 97, no. 1, pp. 417–442, 2017.
 - [20] Y. Yu, X. Li, and C. Qian, "Sdlb: a scalable and dynamic software load balancer for fog and mobile edge computing," in *Proceedings of the Workshop on Mobile Edge Communications*, ACM, pp. 55–60, Los Angeles, CA, USA, August 2017.
 - [21] H. Zhong, Y. Fang, and J. Cui, "Lbbsrt: an efficient sdn load balancing scheme based on server response time," *Future Generation Computer Systems*, vol. 68, pp. 183–190, 2017.
 - [22] J. Yu, Y. Wang, K. Pei, S. Zhang, and J. Li, "A load balancing mechanism for multiple sdn controllers based on load informing strategy," in *Proceedings of the Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific*, pp. 1–4, IEEE, Kanazawa, Japan, October 2016.
 - [23] G. Cheng, H. Chen, H. Hu, and J. Lan, "Dynamic switch migration towards a scalable sdn control plane," *International Journal of Communication Systems*, vol. 29, no. 9, pp. 1482–1499, 2016.
 - [24] T. Hu, Z. Guo, J. Lan, J. Zhang, and W. Zhao, "Easm: efficiency-aware switch migration for balancing controller loads in software-defined networking," 2019, <https://arxiv.org/abs/1711.08659>.
 - [25] S. M. M. Gilani, T. Hong, W. Jin et al., "Mobility management in ieee 802.11 wlan using sdn/nfv technologies," *EURASIP Journal on Wireless Communications and Networking*, vol. 67, 2017.
 - [26] A. Raschella, F. Bouhafs, M. Seyedbrahimi, M. Mackay, and Q. Shi, "Quality of service oriented access point selection framework for large Wi-Fi networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 441–455, 2017.
 - [27] S. Manzoor, X. Hei, and W. Cheng, "Towards dynamic two-tier load balancing for software defined wifi networks," in *Proceedings of the 2017 2nd International Conference on Communication and Information Systems*, pp. 63–67, ACM, Wuhan China, November 2017.
 - [28] Y.-D. Lin, C. C. Wang, Y.-J. Lu, Y.-C. Lai, and H.-C. Yang, "Two-tier dynamic load balancing in SDN-enabled wi-fi networks," *Wireless Networks*, vol. 24, pp. 1–13, 2018.
 - [29] S. Manzoor, X. Hei, and W. Cheng, "A multi-controller load balancing strategy for software defined wifi networks," in *Proceedings of the 2018 4th International Conference on Cloud Computing and Security*, pp. 622–633, Haikou, China, June 2018.
 - [30] Z. Chen, S. Manzoor, Y. Gao, and X. Hei, "Achieving load balancing in high-density software defined wifi networks," in *Proceedings of the 2017 International Conference on Frontiers of Information Technology (FIT)*, pp. 206–211, IEEE, Islamabad, Pakistan, December 2017.
 - [31] T. Zahid, X. Hei, W. Cheng, A. Ahmad, and P. Maruf, "On the tradeoff between performance and programmability for software defined WiFi networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1083575, 12 pages, 2018.
 - [32] S. Manzoor, P. Karmon, X. Hei, and W. Cheng, "Traffic aware load balancing in software defined WiFi networks for healthcare," in *Proceedings of the Information Communication Technologies Conference (ICTC)*, pp. 81–85, IEEE, Nanjing, China, May 2020.
 - [33] S. Manzoor, Z. Chen, Y. Gao, X. Hei, and W. Cheng, "Towards QoS-aware load balancing for high density software defined Wi-Fi networks," *IEEE Access*, vol. 8, pp. 117623–117638, 2020.
 - [34] Y. Zhou, M. Zhu, L. Xiao et al., "A load balancing strategy of sdn controller based on distributed decision," in *Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 851–856, IEEE, Beijing, China, September 2014.
 - [35] C. Chen-Xiao and X. Ya-Bin, "Research on load balance method in sdn," *International Journal of Grid and Distributed Computing*, vol. 9, no. 1, pp. 25–36, 2016.
 - [36] W. Lin and L. Zhang, "The load balancing research of SDN based on ant colony algorithm with job classification," in *Proceedings of the 2016 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA-16)*, pp. 472–476, Atlantis Press, Dalian, China, May 2016.
 - [37] N. T. Hai and D.-S. Kim, "Efficient load balancing for multi-controller in sdn-based mission-critical networks," in *Proceedings of the 2016 IEEE 14th International Conference on*

- Industrial Informatics (INDIN)*, pp. 420–425, IEEE, Poitiers, France, July 2016.
- [38] Y.-W. Ma, J.-L. Chen, Y.-H. Tsai, K.-H. Cheng, and W.-C. Hung, “Load-balancing multiple controllers mechanism for software-defined networking,” *Wireless Personal Communications*, vol. 94, no. 4, pp. 3549–3574, 2017.
- [39] K. Nahida, C. Yin, Y. Hu et al., “Handover based on ap load in software defined wi-fi systems,” *Journal of Communications and Networks*, vol. 19, no. 6, pp. 596–604, 2017.
- [40] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.