IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 August 2023, accepted 5 September 2023, date of publication 8 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3313249

RESEARCH ARTICLE

Achieving QoS in Smart Cities Using Software Defined Wi-Fi Networks

SOHAIB MANZOOR[1], NAEEM IQBAL RATYAL[1], AND HEBA G. MOHAMED[2]
[1]Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, Pakistan
[2]Department of Electrical Engineering, College of Engineering, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Heba G. Mohamed (hegmohamed@pnu.edu.sa)

This work was supported by Princess Nourah Bint Abdulrahman University Researchers Supporting Project PNURSP2023TR140, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT The execution of smart cities around the globe has risen due to the steady connectivity and increased number of wireless devices. Due to low-cost network construction and simple technical implementation, Wi-Fi networks have become a dominant wireless technology to enable the connectivity of Internet-of-Things (IoT) in smart cities. There are a number of services and applications running in smart cities with different demands of the quality of service (QoS). The paper focuses to address the latency problem, which is a key performance metric regarding QoS, in time sensitive applications in smart cities. The emerging paradigm, Software-defined Networking (SDN) is extended for Wi-Fi networks to ensure fairness of traffic load among the access points (AP). We propose three algorithms based on service time, M/G/1 analysis and AP selection to determine the packet transmission delay, packet latency rates and choosing a least loaded destination AP respectively. The optimization of load among the APs ensures a reduced packet latency factor, when a communication link is formed between the smart city IoT devices and the APs. A symmetric load index and a reduced packet latency rate is maintained between the IoT devices and the OpenFlow enabled APs using three software-defined algorithms designed in this study. A Linux based software-defined testbed is developed to ensure the credibility of the algorithms developed. Extensive experimentation using the hardware devices confirm that the proposed algorithms are efficient enough to reduce the latency rate and enhance the throughput rate by 17%, 13% and 9% when compared to received signal strength indicator scheme (RSSI), Po-Fi scheme and aggregated Wi-Fi scheme respectively, by shifting the wireless traffic load from a higher packet latency IoT device to a least loaded AP.

INDEX TERMS QoS, Wi-Fi, SDN, smart city, testbed.

I. INTRODUCTION

Wi-Fi networks have become common in humans life day by day due to its facile connectivity anytime and anywhere [1]. Wi-Fi networks can be easily seen in shopping malls, airports, campus networks and smart homes due to their low-cost network construction and simple technical implementation [2]. The APs constitute the major components of the Wi-Fi network as they provide access to the Internet. Spontaneous deployment of the APs lead to unbalanced network resource utilization and variable AP densities [3]. Research shows that almost 70 percent of the data traffic in the future will be dependent on the Wi-Fi networks outnumbering the wired networks due to rapid increase in portable Wi-Fi associated devices [4]. High usage of smart wearables that are connected to wireless smart phones through Wi-Fi networks instead of cellular connections are making outdoor usage of Wi-Fi networks inevitable [5]. In order to provide a satisfied QoS for voice over IP (VoIP) applications Wi-Fi load balancing still remains a hot topic to explore.

In order to achieve near to optimal throughput especially after COVID-19 pandemic, Wi-Fi networks are commonly used for Internet access in uploads and downloads for applications such as video meetings, video conferences, online classes and healthcare [6]. In order to support functionalities of many IoT applications, Wi-Fi networks are

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

used as the main candidate to design the communication network technologies for smart cities [7]. The fruits offered by the smart cities are however not very attractive for the ordinary people because of many issues arising in today's world of big data such as availability of data, reliability, latency and privacy [8]. In smart cities the hurdle is to transmit and process the real time data sent from the IoT devices to the APs. Some delay sensitive applications need a priority-based service for data processing else the application is worthless. Applications such as health care require service differentiation and load balancing among the APs in terms of wireless traffic load for efficient monitoring of patients and emergency reporting [9].

A massive sensor network comprising of thousands of nodes is required to cover the entire smart city region [10]. The sensor nodes differ in types and are deployed in random locations. Due to the specific functionality of the nodes, some nodes are difficult to replace such as nodes used as regenerative repeaters to support long range communications due to obstacles, nodes which are used to support low power consumption in structure monitoring or nodes which are deployed in remote areas [11]. The huge sensor network requires huge data rates thus outperforming many wireless network technologies such as LoRa or Zigbee [12]. The implementation of Wi-Fi networks with interconnected IoT devices become inevitable. The sensor data from the IoT devices is transferred over the Wi-Fi networks to specific servers or management entities. The IoT devices need to ensure a threshold for packet latency overcoming the fact that they may lie in overlapped Wi-Fi regions [13]. In the basic service set (BSS), some APs continuously receive normal administrator traffic while other APs become overloaded with delay sensitive traffic received from health-care applications or VoIP. In such scenarios an adaptive network configuration is required that connects the IoT device sending delay sensitive information to an AP which is underloaded [14]. The fairness of load among the APs cannot be achieved in standard IEEE 802.11 networks and hence non standardize protocols/hardware makes it almost impossible to achieve load balancing among the APs in smart cities.

A new network architecture, SDN is recently introduced to address the problems in wireless networks [15]. In SDN, the data plane is separated from the control plane giving the network administrator an overall view of the network to install applications easily in wireless networks without changing the hardware. There are almost very few studies that use the functionality of SDN to upgrade the network infrastructure of smart cities. SDN has the power to dynamically configure the wireless network resources according to the network conditions [16]. SDN simplifies the complexities of wireless networks such as future 6G networks or Wi-Fi networks. Due to the abstraction of the control plane and data plane many vendors such Cisco, Microsoft, HP and Google are supporting OpenFlow standards [17]. By inducing SDN into Wi-Fi networks, we believe that we can achieve optimal load balancing among the APs in software defined Wi-Fi networks (SD-WiFi) deployed in smart cities.

In this paper, network resource utilization is achieved by using the functionalities of SDN into the W-Fi networks deployed in smart cities. An SDN controller is programmed to associate the IoT devices to the least loaded AP for optimal packet latency performance. In traditional Wi-Fi networks the association decisions are made by the wireless devices based on the received signal strength indicator (RSSI). The wireless device-based association decisions do not guarantee throughput as the AP showing highest RSSI could be overloaded providing the least access to the Internet. In the proposed work the SDN controller computes the reports received by the APs to make network wide association decisions that which IoT should be connected to which AP in order to achieve the lowest packet latency rate. The smart city design incorporates multiple APs with overlapped coverage making a dense Wi-Fi network scenario with mobile IoT devices. The IoT devices connect to the OpenFlow enabled AP to access the internet. The SDN controller has the load balancing applications installed on it which collect the AP reports and make network wide computations to ensure fairness of load among the APs by performing handoffs. The proposed research introduces three algorithms based on service time estimation, M/G/1 analysis and choosing the least loaded AP. The purpose is to maintain the load symmetry among the APs and at the same time reduce the end-to-end delay for a satisfied QoS. Service time-based algorithm choses the IoT device with the highest end-to-end packet delay to be de-associated and then re-associated to a least loaded AP. The algorithm runs on a constant network traffic rate and efficiently handoffs the IoT device. The M/G/1 based algorithm also shifts the IoT device with highest end-to-end delay to a least loaded AP. The algorithm makes use of the random traffic distribution using Poisson process. The SDN controller ensures the fairness of load among the APs deployed in the smart city. The major contributions of the proposed work are:

- An algorithm based on service time is designed to find the highest end-to-end packet transmission delay for an IoT device without changing any hardware configuration. The computations are performed by the SDN controller using the packet arrival time information.
- Highest end-to-end packet latency for an IoT device is calculated using M/G/1 analysis. The computations are performed regardless of any hardware change.
- A third algorithm is designed to reassociate the IoT devices with highest end-to-end packet latency rate to the AP with the least load.
- A Linux based software defined Wi-Fi networks (SD-Wi-Fi) testbed is designed to verify the credibilities of the aforementioned algorithms.

The remainder of this paper is organized as follows. Section II explores the related work. Network model is explained in Section III. Section IV explains the proposed algorithm designs. Section V describes the experimental setup. Section VI discusses the results gained from the experimental setup. The conclusion is presented in Section VII. Tables 1 show the list of abbreviations used in this study.

TABLE 1. List of abbreviations.

Abbreviations	Description
AHP	Analytical Hierarchal Process
AI	Artificial Intelligence
AP	Access point
BSS	Basic service set
IoT	Internet of things
LVAP	Light virtual access point
MAC	Medium access control
OAP	OpenFlow enabled AP
OVS	Open virtual switch
PC	Personal computer
PK	Pollaczek-Khinchine
PTP	Precision time protocol
QoS	Quality of service
RSSI	Received signal strength indicator
SDN	Software defined networking
SD-Wi-Fi	Software defined Wi-Fi network
SDWN	Software defined Wi-Fi network
SNMP	Software network management protocol
SSID	Service set ID
TCP/IP	Transmission control protocol/Internet protocol
UDP	User datagram protocol
Wi-Fi	Wireless fidelity
WLAN	Wireless local area network

II. RELATED WORK

The centralized controller in the SDN framework is responsible to make network wide decisions [18]. A lot of research work has been done where SDN is chosen as the main paradigm to choose the AP. SDN is used to address the scalability issues at the control plane [19]. A two-tier arrangement of controllers is used to balance the load among the controllers. Analytical hierarchal process (AHP) is employed as an application to prioritize the most sensitive traffic packet to be processed first. The latency factor is not considered. Data rate base fittingness function is used by the SDN controller to choose the most least loaded AP [20]. The research does not take into account the end-to-end packet latency. Healthcare scenarios are taken into account using SDN to address the packet delay ratio through simulations [21]. A high-density Wi-Fi network is proposed with SDN capabilities to ensure load balancing [22]. Fairness of load among the APs is discussed without discussing the jitter factor. The load on the APs is balanced through a learning-based algorithm using software defined W-Fi networks (SDWN) [23]. The work takes into account the throughput for TCP connections. A QoS aware study for Wi-Fi network is done using SDN [24]. A same radio channel is used for all APs and wireless stations. A traffic aware load balancing based simulation research is performed [25]. The research considers the service differentiation while ignoring the end-to-end delay.

An algorithm is designed for Wi-Fi networks which is capable of choosing a least loaded AP among the loaded ones [26]. The study considers the throughput as the main metric for the performance evaluation. Multi-criteria association-based study makes use of SDN in the Wi-Fi networks [27]. Smooth handoffs are presented considering multi metrics. The load matrix among APs and packet latency rates are ignored. Virgil is proposed to choose an AP with the best connection quality [28]. The paper discussed the AP load and the traditional received signal strength indicator (RSSI) methods. The research demanded the hardware to be IEEE 802.11 k/v supported. The IoT devices are used in many applications and have become the main part of the smart city designs [29]. A deterministic load balancing algorithm is designed for the IoT devices based on the game theory approach [30]. The game theory approach inculcates extra nonlinear complex mathematical equations to the system model. In multi hop networks, video load balancing method is proposed [31]. The performance of video applications in the mesh network is improved. The method relies on routing algorithms and fully ignores the packet delays.

The wireless traffic load is modeled in an SDN supported Wi-Fi network [32]. Four tier network model is used to achieve load balancing. Meta heuristic techniques are employed to balance the load among the OpenFlow switches. The performance metrics consider the fairness of load among the controllers and throughput achieved. Data offloading schemes are presented in 5G networks using multiple channels [33]. The work focuses on the bandwidth allocation ignoring the latency metrics. A distributed coordination function study uses dense Wi-Fi networks [34]. The main aim of the study is to evaluate the access modes performance for different system parameters. The study focuses on optimal throughput performance for different load conditions. QoS award load balancing is achieved in Wi-Fi networks using SDN [24]. The OpenFlow protocol is modified to support the function of load balancing among the APs. The scheme focuses on the performance attributes of semi centralized and fully centralized load balancing options. An SDN based handoff reduction algorithm is designed [35]. The scheme uses the simple network configuration manager to help reduce the detection and discovery times. The load on the APs is not the main focus neither the latency rate. A detailed comparison table is shown in Table 2. The table compares the proposed scheme with the previous research work conducted in the field of load balancing for Wi-Fi networks.

To the best of our knowledge this is the first standardize study to minimize the latency rate of packets transmitted from the IoT devices to multiple APs in a smart city environment using a testbed. The APs installed work on multiple radio channels. The SDN controller is responsible to maintain the threshold of latency for the packets transmitted from IoT devices towards the APs. The SDN controller having the overall view of networks, reduces the packet latency by handing over the highest end-to-end latency IoT device to the least loaded AP.

TABLE 2. Comparison table for load balancing methods in Wi-Fi networks.

S.No	Authors	Main findings (research gaps)	Proposed solutions
1	Deb et al. [18]	SDN controller is used as a distributed entity. Routing policies are focused without considering End-to-End delay metrics.	SDN controller is used as a centralized entity. End-to-End packet delays are considered.
2	Manzoor et al. [19], [24], [25], [32], [34]	Controllers load is considered. No testbed is made. Latency factor is not considered.	Load on APs is balanced. Testbed is made. Packet transmission delays are calculated.
3	Gomez et al. [20]	Least loaded AP determined. No latency factor considered.	Service time and latency factor are monitored.
4	Barka et al. [21], Chen et al. [22]	Throughput performance monitored. No mathematical modeling is performed. Network jitter is neglected.	Mathematical model is designed to optimize load on APs. Network jitter is considered.
5	Al-Jawad et al. [23]	Heuristic based algorithm is used for AP load balancing. No handoff consideration.	Concrete algorithms are developed for monitoring latency. Analysis is done before and after the handoffs.
6	Zhong et al. [26]	Least loaded AP is determined. Approximation methods are used. No testbed is formed.	Least loaded AP is determined by SDN. Mathematical modeling is performed. Testbed is developed.
7	Sohaib et al. [27], Bhatti et al. [28]	Multi-criterion chosen for handoffs. Simulation based study is used.	SDN used for handoffs. Testbed based study is used.
8	Islam et al. [29], Babar et al. [30]	Complex mathematical equations are formed. Game theory approach is used. Simulation based study is used.	Concrete mathematical modeling is performed. Real time study is done. Testbed is developed.
9	Kumar et al. [31],	AP load balancing is considered. No packet delay analysis is performed.	AP load balancing is performed using mathematical modeling and testbed. Packet delays are considered.
10	Raja et al. [33]	Bandwidth allocation is focused. No packet latency metric is analyzed.	QoS is focused. Latency factor is calculated.

III. NETWORK MODEL

The proposed network model takes into account a smart city design where the APs are deployed to constitute a city-wide Wi-Fi network. The APs form a dense Wi-Fi network where all the APs have an overlapped coverage. The Wi-Fi network supports all city-wide wireless applications communication over the IoT devices as depicted in the Figure 1. In the smart city-wide Wi-Fi network design two types of data traffic are incorporated. The first type of traffic relates to the human generated traffic such as from the smart wireless devices and the second type of traffic relates to the machines generated traffic such as from the IoT devices that support various technologies such as Bluetooth, LoRa, Zigbee etc. All the traffic is routed through OpenFlow enabled APs which are connected to the SDN controllers. The application plane in the SDN takes control of load balancing application. The load balancing application ensures fairness of load among the APs which in return reduces the packet latency. The control plane has the SDN controller or several controllers which communicate to all APs through the OpenFlow protocol. The OpenFlow acts as a bridge of information exchange between the APs and the controller. The data plane has the forwarding devices which generate all wireless traffic. The wireless traffic is routed through APs to the SDN controller. The OpenFlow protocol is used as the communication protocol standard in SDN where as in some previous research related to e-Health other protocols and architectures are used for specific applications. A simple network management protocol (SNMP) is deployed in the application plane to collect all the packet information from the OpenFlow enabled APs. On receiving these reports, the SDN controller makes the computations related to load balancing. During the beacon frame reply and response procedure the media access control (MAC) address of the devices are also communicated to the SDN controller through the OpenFlow protocol hence allowing the SDN controller to de-associate a wireless device from an AP and then re-associate with the least loaded AP.

IV. ALGORITHM DESIGNS

This section describes the design of three proposed algorithms. The two algorithms aim to find the IoT devices with the highest end-to-end latency rate. After the IoT devices are selected they undergo a handoff as instructed by the SDN controller. The design of first algorithm is based on packet delivery time estimation and the second algorithm design is based on the queuing analysis theory. After the IoT device with the highest end-to-end latency rate is selected through the first two algorithms, it is handed over to the least loaded

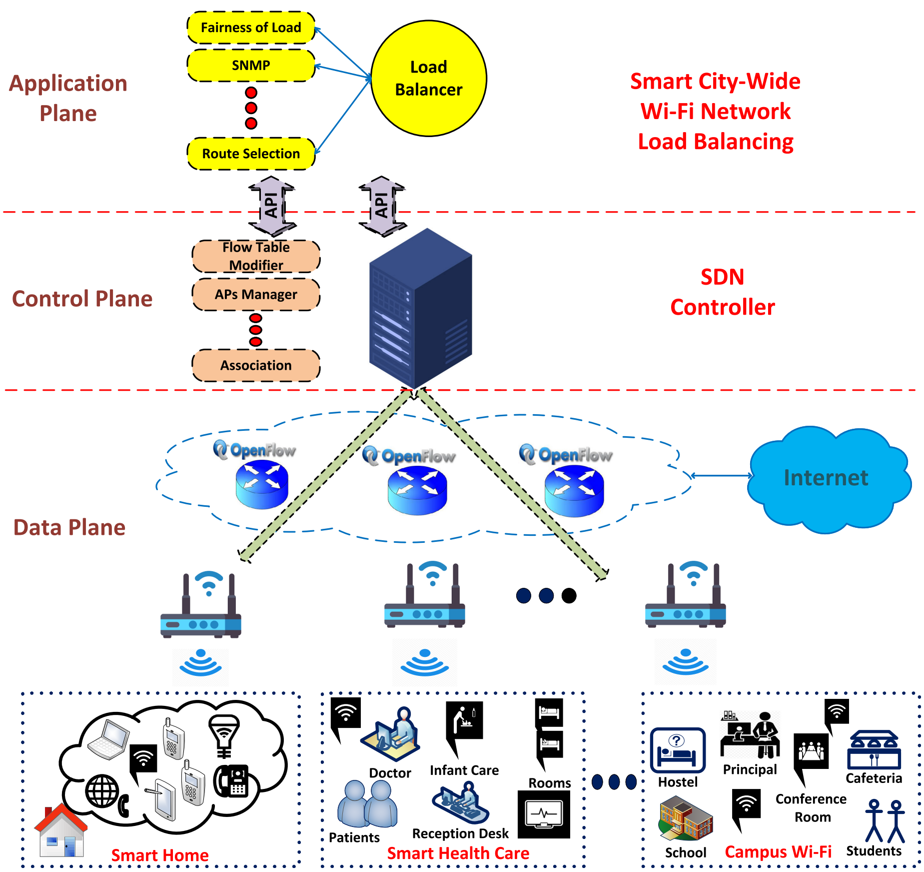


FIGURE 1. The proposed architecture.

AP. The selection of the least loaded AP is performed through the third algorithm.

A. ESTIMATION OF SERVICE TIME

The end-to-end packet delay ETE_d for an IoT device x is calculated by the summing the queuing delay QD_{tx} and the service time ST_{tx} as shown in Eq.1.

$$ETE_d = QD_{tx} + ST_{tx}. \tag{1}$$

The queuing delay varies with the service time and packet arrival rate. In the proposed scheme the arrival rates for all the arrival processes are kept same and it is assumed that all the arrival processes are similar. When the network operation region is far away from the saturation region, the queuing delay in comparison to the service time is negligible. Hence the end-to-end delay is directly dependent on the service time, the longer the service time the longer the end-to-end delay. In Wi-Fi networks the average service time is calculated as shown in Eq.2.

$$[ST_{tx}]_{avg} = \sum_{a=1}^{N-1} \emptyset_u \left[\partial t_{tu} + \frac{\partial c_{tu}}{2}\,\frac{\rho_c}{1-\rho_c}\right] + \beta\left(\rho_c\right) + \partial t_{tx} + \frac{\partial c_{tx}}{2}\,\frac{\rho_c}{1-\rho_c}. \tag{2}$$

The node u, queue utilization is calculated as $\emptyset_u = ST_{tu}\Lambda$, where Λ is the packet arrival rate, the probability of collision is represented as ρ_c and the backoff interval time is calculated in Eq.3.

$$\beta\left(\rho_c\right) = \frac{1-\rho_c-\rho_c(2\rho_c)^{bu}}{1-\rho_c}\,\frac{CW_{minimum}}{2}. \tag{3}$$

where ∂t_{tu} and ∂c_{tu} are the transmission and collision times for node u, $CW_{minimum}$ is the minimum contention window size. The transmission and collision times are mathematically expressed as $\partial t_{tu} = pt/\Upsilon_u + \partial_{N1}$ and $\partial c_{tu} = pt/\Upsilon_u + \partial_{N2}$ respectively. pt is the packet size, $N1$ and $N2$ are the constant times as prescribed in IEEE 802.11 standard protocol and the channel transmission rate for node u is presented as Υ_u. The working principle of the proposed algorithm is as follows:

1) The Packet_In events are initiated when the packets reach the OpenFlow enabled APs and are forwarded to the SDN controller. The recorded arrival times of the Packet-In events are used by the SDN controller to find the standard deviation and average standard deviation of the interarrival times of the Packet_In event. The information is used for the packet service time calculations.
2) The highest value of the average interarrival time will decide a handoff to be taken. If there are number of higher interarrival time values then based on the highest

standard deviation value, the specific IoT device will be chosen for the handoff.

3) The monitoring of the Packet_In events will be done continuously. If the interarrival time is not reduced for the IoT device that was re-associated, the handoff to another AP will take place.
4) On the addition of a new IoT device to the network and on the load imbalance of the OpenFlow enabled APs, the SDN controller monitors the interarrival times of the Packet_In messages. If the average value of the packet interarrival times exceed the threshold values, step 2 is repeated.

The steps are summarized in algorithm 1.

Algorithm 1 Estimation of Service Time

Receive the packets.
Calculate the standard deviation and average of interarrival times for each IoT device.
Select the IoT device with the highest average interarrival time.
if more then one IoT device have the maximum average interarrival time **then**
  Select the IoT device with the highest standard deviation.
  Hand it over to the least loaded OpenFlow enabled AP.
else
  Hand it over to the least loaded OpenFlow enabled AP.
  if The average interarrival time is not minimized **then**
    .
    Calculate the standard deviation and average of interarrival times for each IoT device.
  else
    if Any new IOT device joined the network and the APs load is imbalanced **then**
      .
      Calculate the standard deviation and average of interarrival times for each IoT device.
    else
      Check if any new IOT device joined the network and the AP load is imbalanced.
    end if
  end if
end if

B. M/G/1 ANALYSIS

The wireless traffic is generated from a number of IoT devices. The aggregated wireless traffic is modeled for approximation using a Poisson arrival process. Pollaczek-Khinchine (PK) formulae is used to calculate the end-to-end delay with the help of M/G/1 queuing analysis. Using the PK formula, the mean waiting time for the packet in the queue is calculated through Eq.4.

$$Que_{tx} = \Lambda * \frac{(\vartheta_{sx}^2 + [ST_{tx}]_{avg}{}^2)}{2 * (1 - \Lambda * [ST_{tx}]_{avg})}. \tag{4}$$

where ϑ_{sx} represents the standard deviation of the service time at the node x, Λ is the packet arrival rate, $[ST_{tx}]_{avg}$ is the average service time. Using Eq.1, the average end-to-end packet delay of node x is calculated using Eq.5.

$$[ETD_d]_{avg} = Que_{tx} + [ST_{tx}]_{avg}. \tag{5}$$

The working principle of the proposed algorithm is as follows:

1) The SDN controller after saving the Packet_In events, computes the standard deviation and average standard deviation for the interarrival times.
2) The SNMP reports the SDN controller regarding the packet arrival rate. The PK formula is used by the SDN controller to compute the end-to-end delay for each IoT device. From here the service time of each IoT device is deduced.
3) The IoT device with the maximum end-to-end delay is selected for handoff.
4) The IoT device which underwent a handoff is monitored to see if its end-to-end packet delay is lowered. If the delay is not lowered then step 2 is repeated for another least loaded AP.
5) When ever a new IoT device is associated to an AP or there is an imbalance of the AP load, the end-to-end delay for all IoT devices is rechecked by the SDN controller. When the end-to-end delay exceeds in value, step 2 is repeated.

The steps are summarized in algorithm 2.

Algorithm 2 M/G/1 Analysis

Receive the packets.
Calculate the standard deviation and average of interarrival times for each IoT device.
Using the PK formula from Eq.4, approximate the end-to-end delay for each IoT device.
Select the IoT device with the highest end-to-end delay.
Hand it over to the least loaded OpenFlow enabled AP.
if more then one IoT device have the maximum average interarrival time **then**
  .
  Calculate the standard deviation and average of interarrival times for each IoT device.
else
  if Any new IOT device joined the network and the APs load is imbalanced **then**
    .
    Calculate the standard deviation and average of interarrival times for each IoT device.
  else
    Check if any new IOT device joined the network and the AP load is imbalanced.
  end if
end if

C. FINDING THE LEAST LOADED AP

The algorithms find the least loaded AP. Once the least lead AP is found the SDN controller initiates the handoff. The IoT devices with the maximum average interarrival time as calculated in section IV-A or the IoT devices with the maximum end-to-end delay as calculated in section IV-B, are handed over to the least loaded AP. The working principle of the proposed algorithm is as follows:

1) The SDN controller is responsible to receive all the flows from the OpenFlow enabled APs. When ever a packet is received the Packet_In event is triggered.
2) The SDN controller on receiving the flows makes the computations such as extraction of number of packets received form a specific AP. The computations are performed periodically.
3) The least loaded AP chosen by the SDN controller is the one with the minimum number of Packet_In messages.
4) SDN controller computes the Packet_In messages continuously. Step 2 is repeated for extraction of packets coming from a specific AP to choose a least loaded AP.

The steps are summarized in Algorithm 3.

Algorithm 3 Finding the Least Loaded AP

```
Receive the packets.
Increment the packet count received from the AP, also
keep the record of the AP from where the packet came from.
Calculate the standard deviation and average of interarrival times for each IoT device.
Select the IoT device with the highest average interarrival time.
if more then one IoT device have the maximum average interarrival time then
    .
    Select the IoT device with the highest standard deviation.
    Select the AP with the minimum packet count.
    Hand the IoT device to the chosen OpenFlow enabled AP.
else
    Select the AP with the minimum packet count.
    Hand it over to the chosen OpenFlow enabled AP.
    if The average interarrival time is not minimized then
        .
        Calculate the standard deviation and average of interarrival times for each IoT device.
    else
        if Any new IOT device joined the network and the APs load is imbalanced then
            .
            Calculate the standard deviation and average of interarrival times for each IoT device.
        else
            Check if any new IOT device joined the network and the AP load is imbalanced.
        end if
    end if
end if
```

V. EXPERIMENTAL PLATFORM

A real-time Linux based software defined testbed as depicted in Figure 2 is built to verify the credibilities of the three proposed algorithms. Extensive emulation runs are carried out on the experiment setup build with a real hardware.

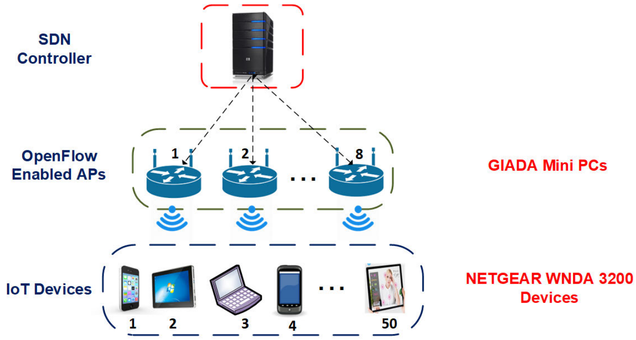


FIGURE 2. The experimental setup with the IoT devices and OpenFlow enabled APs.

The OpenFlow enabled APs are emulated using GIADA Mini PCs. In the experimental setup 8 OpenFlow Enabled APs are used. The transmission range of each OpenFlow enabled AP is 1-33m. NETGEAR WNDA 3200 devices are used as the IoT devices. 50 IoT devices are used which continuously generate data traffic at the transmission rate of 400 to 600 packets per second. The transmission rates of 400 to 600 per seconds are used to design a high density SD-Wi-Fi. In the performance evaluation, the network is initially loaded with a transmission rate of 400 packets per second to verify the latency rate after load optimization and later the network is loaded with a transmission rate of 600 packets per seconds to monitor the latency rate after load optimization. IEEE 802.11 "n" standard is used in the experimental setup. The personal computer (PC) used in the setup, runs 19.10 Ubuntu operating system. The wired connections are established through Gigabit Ethernet ports. A single run took about 30 minutes of test time.

The step wise implementation details of the testbed are:

1) The USB wireless cards are used to function as APs and the IoT devices. The Ubuntu PCs are also used as the IoT devices. The tools used to configure the wireless USB cards are *iw* and *hostapd* so that the USB cards act as IoT devices and APs respectively. In the proposed research a low cost test bed is formed with the help of the wireless USB cards which are configured with a unique IP address.
2) The testbed architecture is shown in Figure 3. To support the management provided by the SDN controller the APs connect to the SDN switch. The USB wireless port is added to the Open Vswitch (OVS) using the $ovs - vsctl$ command that manipulates the wireless messages. The IP layer is managed by the bridge. The bridge does not provide any additional support for the power control, transmission rate and channel utilization. *iw* and *hostapd* are used to modify the wireless parameters. The SDN controller is connected to the APs using the transmission control protocol (TCP) link. In the control plane the information between the wireless controller and OpenFlow controller is shared with the help of the extensible markup language *XML* which support the parameter files.

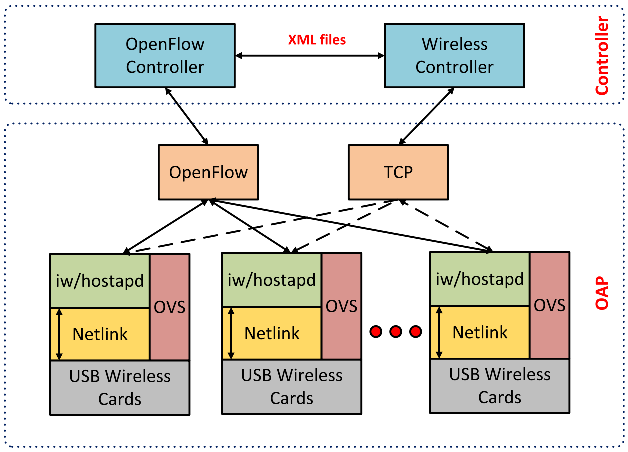


FIGURE 3. The architecture of testbed.

3) The information of the IoT devices in the coverage area of the APs is collected through the *iw AP_Namestation* dump command line interface (CLI). Similarly, the IoT devices collect the discoverable AP information through the *iw STA_Name* scan CLI. The information collected is forwarded the RYU SDN controller.

In order to make a smooth functionality of the testbed many software packages are also used such as Floodlight, Empower-5G framework and Open vSwitch (OVS), which support the hardware. Floodlight package is a component-based framework for software-defined networking. It is used for the proper functioning of the SDN controller. The floodlight package is installed on a dedicated PC using the OpenWRT which supports the OpenFlow standards. The APs are OpenFlow enabled mean that the OVS switch is preinstalled in them using the OVS software. The OpenFlow protocol acts as a bridge for communication between the SDN controller and the OpenFlow enabled AP.

Empower-5G framework which is an SDN based radio access network, enables the seamless handovers of IoT devices from one AP to another. The frame work is installed in the PC and run through the SDN controller. Using Empower-5G, all the APs registered in the Wi-Fi network obtain a unique service set identifier (SSID). The Empower-5G framework establishes a light virtual access point (LVAP) for each IoT device that associates to the OpenFlow enabled AP. During the handovers, instead of physical device re-associations, LVAPs are shifted from one AP to another making the handovers smooth and seamless and thus maintaining the throughput. The LVAPs can be easily installed and deleted on the APs by the SDN controller. The RUDE/CRUDE software is used to emulate the traffic generated by the IoT devices. The RUDE/CRUDE software has the functionality of usergram data protocol (UDP) traffic generator which follows the server/client pattern. The background traffic can also be generated at different data rates by the same software. In order to measure the end-to-end delay between the IoT devices and the destination device, a precision time protocol (PTP) is used.

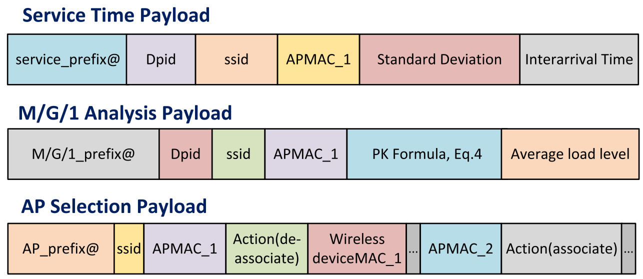


FIGURE 4. Payloads used to achieve QoS in the smart cities environment.

VI. PERFORMANCE EVALUATION

The performance evaluation of the three proposed algorithms is explained in this section. The performance of the algorithms is validated through extensive experimentations. The test time for each experiment has been keep for approximately 30 minutes.

A. SD-WI-FI IMPLEMENTATION

A real time testbed differs from an emulation platform due to a number of reasons. In the emulation platform signal fading and channel interference are usually ignored. These factors have a key importance when dealing with a testbed platform. More development efforts are involved while developing a prototype testbed. It is costly to make a testbed in comparison to the simulation/emulation platform especially when designing a high density SD-Wi-Fi where there are a large number of IoT devices involved.

B. CHALLENGES AND LIMITATIONS

While creating the testbed for the three algorithms proposed a number of challenges arose. The challenges and experimental details are listed below:

- In this study the OpenFlow standard has been extended to the wireless networks where it has only been used for the wired networks in the past. The service time, M/G/1 analysis and finding the least loaded APs in the proposed study required wireless management and mobility. The extended OpenFlow format for wireless networks is illustrated in Figure 4.

  In order to achieve the QoS in the smart city environment, the OpenFlow extension message formats are depicted in the Figure 4. The payloads illustrate the working of the three proposed algorithms. The service time payload used for the service time, transports the standard deviation and interarrival times information from the APs to the SDN controller. The fields also carry the service set ID (ssid), daemon process ID (Dpid) and the media access control (MAC) address of the AP. The M/G/1_prefix@ is similar to the payload used for service time with the exception that it carries the PK formula information along with average load levels. The last payload AP_prefix@ is used to find the least loaded AP. It carries the information related to association and dissociation. The fields hold the information related to MAC address of the overloaded AP and least loaded AP. The payload also carries the MAC addresses of IoT

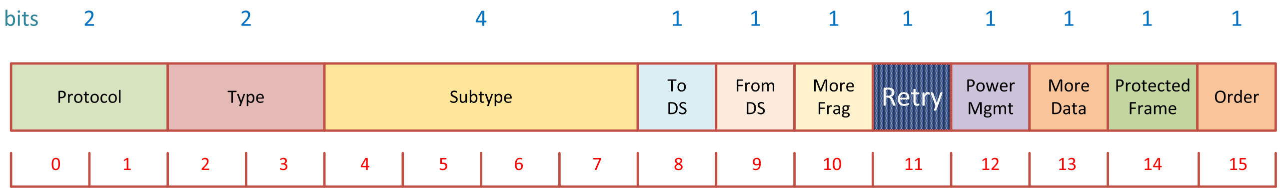


FIGURE 5. The MAC header frame control domain.

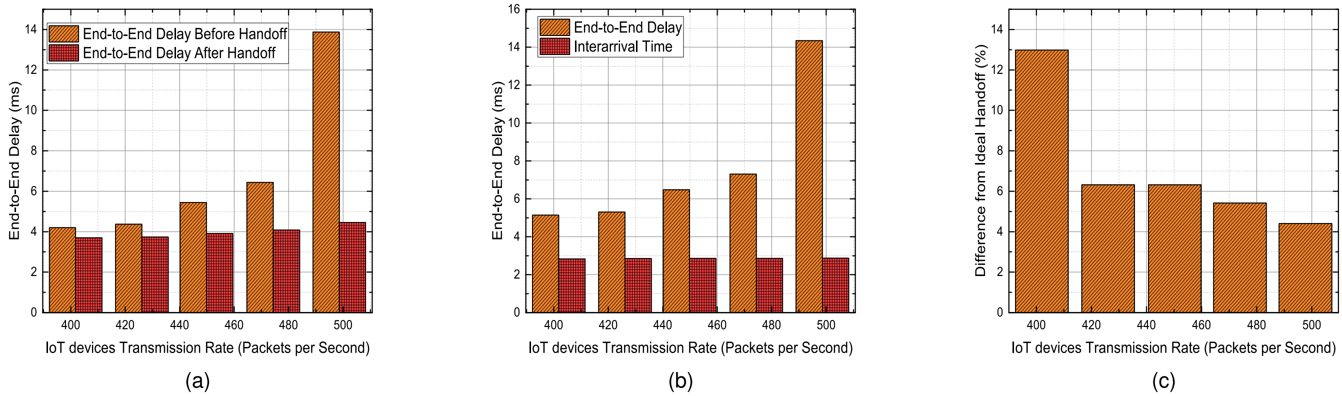


FIGURE 6. (a) End-to-End packet delay before and after the handoffs. (b) Comparison of End-to-End packet delay with interarrival time events. (c) Comparison of proposed algorithm with the ideal handoff.

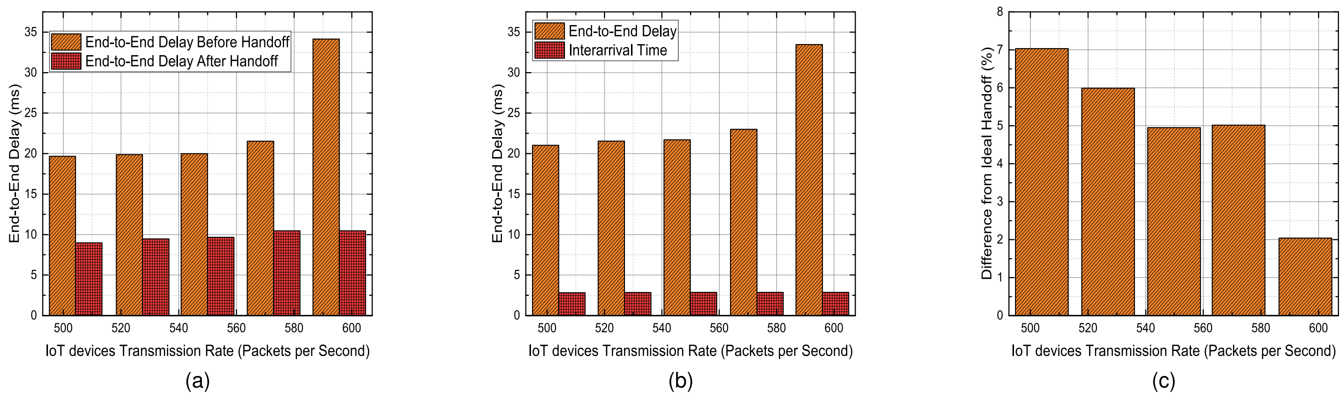


FIGURE 7. (a) End-to-End packet delay before and after the handoffs for increased transmission rates. (b) Comparison of End-to-End packet delay with interarrival time events for increased transmission rates. (c) Comparison of proposed algorithm with the ideal handoff for increased transmission rates.

devices to be deassociated from an overloaded AP and reassociated to an underloaded AP.

- The testbed development consumes a lot of cost and coding as high density SD-Wi-Fi involves large number of IoT devices which are connected to the APs. Each wireless device needs an extra lines of code in the OpenWRT_emulation platform.
- We have used the packet transmission rates from 400 to 600 packets per second to ensure a loaded SD-Wi-Fi scenario but in real testbed platform the IoT devices not always transmit a packet so additional sensing functions are needed to be devised for the IoT devices. The MAC header frame control domain is depicted in the Figure 5. The 11 bit is the depiction of the retransmission indicator. The retransmission indicator is switched to 1, whenever the frame is retransmitted. The APs determine the packet retransmissions by checking the retransmission indicator of the received frames. The retransmission indicator is also added to the OpenFlow standard payloads to support the additional sensing functions of the IoT devices.

C. EFFECT OF SERVICE TIME ON HANDOFFS

1) TRANSMISSION RATE OF 400 TO 500 PACKETS/S

In order to study the performance evaluation of the first algorithm presented in section IV-A, we have used three performance metrics. The first performance metric makes the comparison for handoffs regarding the end-to-end packet delay with a variable traffic rate. The Figure 6 (a) show that the end-to-end packet delay decreases after the handoff specifically when the traffic rate is much higher as explained in Eq.2. The second performance metric makes a comparison between the end-to-end packet delay and the interarrival time of the Packet_In events received by the SDN controller.

Figure 6(b) shows the comparison for variable traffic rates. The third performance metric validates the application of the service time algorithm for a real time handoff to be made. The end-to-end delay for the handoff is recorded by the end user using a smart device (ideal situation). The credibility of the algorithm is seen in Figure 6(c) and it is observed that when the proposed algorithm is used, the percentage of difference reduces with the increase in traffic rate.

2) TRANSMISSION RATE OF 500 TO 600 PACKETS/S

Another set of experimentation is performed to validate the credibility of the proposed algorithm. In the first set of experimentation the IoT traffic was kept at the transmission rate from 400 to 500 packets per second. By increasing the packet rate we have made the network loaded. In the second set of experimentation where the transmission rates for IoT traffic is increased from 500 to 600 packets per seconds, a significant difference is observed in the end-to-end delay before and after the handoff as shown in Figure 7(a). The significant difference in the end-to-end to delay is observed due to changing the interarrival times of the packet_In events received at the SDN controller as depicted in Figure 7(b) and calculated from Eq 3. It is also observed from Figure 7(c), that when the network load is increased the proposed algorithm handoff almost matches the ideal handoff scenario.

It is evident from Figure 6 (a) and Figure 7 (a) that the proposed algorithm running on the SDN controller and estimating the service time by counting the Packet_In events significantly reduces the end-to-end packet delay for the IoT devices after the handoffs are performed. This is due to the fact that the re-transmissions are lowered. Another deduction from Figure 6 (b) and Figure 7 (b) is when the traffic rate is constant, the end-to-end packet delay at the destination IoT device can be calculated from the interarrival time of the Packet_In events received at the SDN controller. By looking at Figure 6 (c) and Figure 7 (c), the handoffs are even credible when the traffic rate is increased which validates the functionality of the proposed algorithms. The increase in network traffic pushes the network towards saturation with higher backoff times, making the end-to-end delay more dependent on the variations in service time.

D. EFFECT OF M/G/1 ANALYSIS ON HANDOFFS

1) TRANSMISSION RATE OF 400 TO 500 PACKETS/S

The performance evaluation of the algorithm as presented in section IV-B is checked in this section. The end-to-end delay variation is observed for the change in packet rates. The Figure 8 (a) depicts the credibility of the algorithm as the end-to-end packet delay decreases rapidly after the handoff. The case is not the same when the interarrival time is varied for the Packet_In events received by the SDN controller using the Poisson distribution of the traffic as observed from Figure 8 (b). Thus, there is not much of a change in the end-to-end packet delay as depicted in Figure 6 (b) where the arrival rate was kept fixed.

2) TRANSMISSION RATE OF 500 TO 600 PACKETS/S

On loading the network in terms of increased packet rates, the performance of the proposed algorithm is evaluated in terms of the end-to-end packet delay as shown in Figure 9(a) and Figure 9(b). The end-to-end packet delay decreases after the handoff as observed from Figure 9(a). On increasing the network traffic, the difference between the end-to-end packet delay before and after the handoff is comparatively small. The ensures the smoothness of handoffs by the proposed algorithms. By changing the interarrival times for Packet_In events received by the SDN controller, the effect on the end-to-end packet delay is negligible as depicted in Figure 9(b). The estimated end-to-end delay calculated by the SDN controller is compared to the end-to-end delay before the handoff as shown in Figure 8(c). The difference in the time arises due to the time taken by the operating system to fix a time-stamp for outgoing and incoming packets. This delay is system oriented and cannot be tackled.

While using the M/G/1 analysis for the handoffs, we can observe a similar pattern for performance evaluation as obtained in service time estimation algorithm. The end-to-end packet delay reduces in the proposed M/G/1 based algorithm after the handoff. On increasing or decreasing the network traffic there is no effect on the packet interarrival time events because due to the nature of the Poisson traffic, the queuing delay in the shared IEEE 802.11 channel affects the end-to-end delay more than the service time.

E. HANDOFF TO A LEAST LOADED AP

The Figure 10 (a) and Figure 10 (b) show the performance evaluation of the algorithm proposed in section IV-C, to choose the least loaded AP. The performance comparison is made between the proposed algorithm and the algorithm that measures the end-to-end delay for the IoT devices before and after the handoff by choosing a destination AP randomly. The traffic load is varied for both comparisons and the end-to-end packet delay is calculated. It is obvious from the findings that the end-to-end delay for the proposed algorithm before and after the handoff is significantly then the algorithm which choses the destination AP randomly.

The least loaded AP is chosen by the SDN controller by calculating the wireless traffic load on each AP. The decisions made by the SDN controller hence has an impact on the end-to-end packet delay and channel capacity. The IoT devices are mobile and continuously reporting their data to the SDN controller, hence the proposed algorithm is not affected by the mobility of the IoT devices.

F. AGGREGATE THROUGHPUT

The comparison of the aggregate throughput is made with four schemes, the traditional RSSI scheme, Po-Fi scheme [36] and aggregated Wi-Fi scheme [37]. The aggregate throughput performance is shown in Figure 11. As the number of IoT devices increase the aggregate throughput increases initially and then tends to slow down gradually. In the Wi-Fi network the distributed coordination function (DCF) states that when the traffic load is increased the throughput reaches a saturation limit. The RSSI based

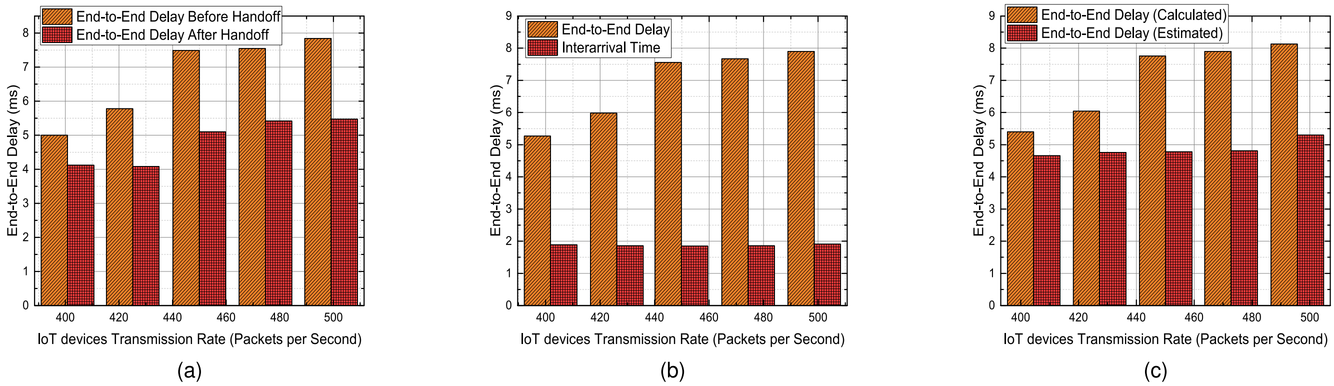


FIGURE 8. **(a) End-to-End packet delay before and after the handoffs. (b) Comparison of end-to-end packet delay with interarrival time events. (c) Calculated and estimated end-to-end delay.**

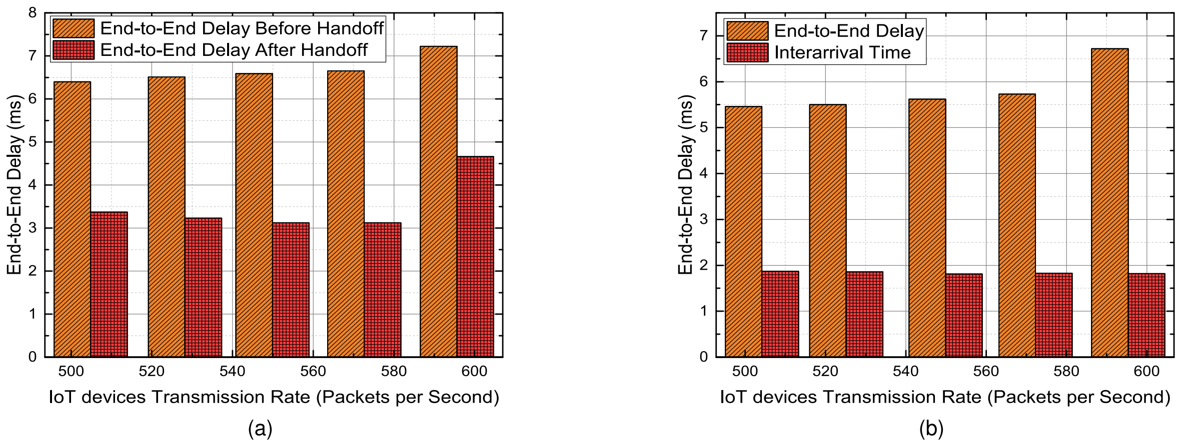


FIGURE 9. **(a) End-to-End packet delay before and after the handoffs with increased transmission rates. (b) Comparison of End-to-End packet delay with interarrival time events with increased transmission rates.**

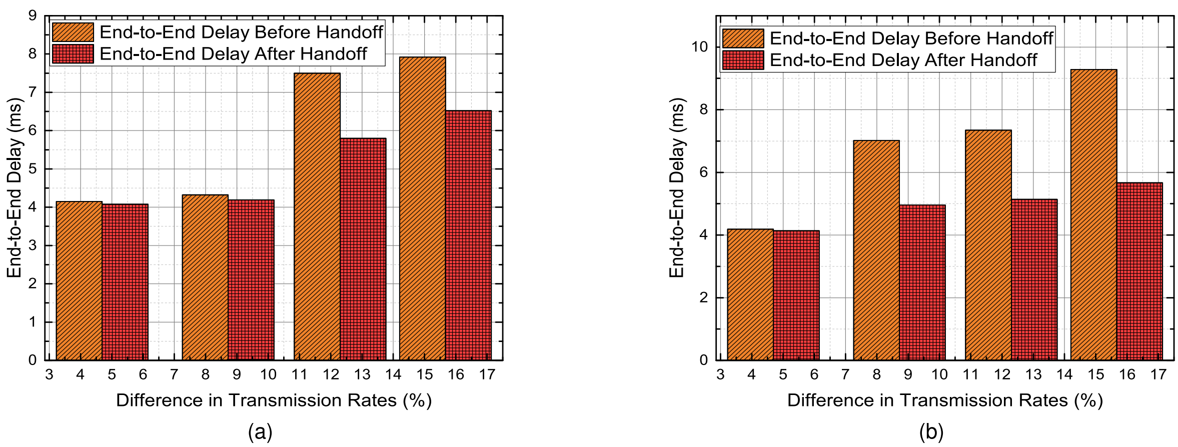


FIGURE 10. **(a) End-to-End packet delay before and after the handoffs with algorithm chosing random APs. (b) End-to-End packet delay before and after the handoffs with the proposed algorithm for choosing least loaded AP.**

throughput shows the poor performance as only one criterion is chosen for AP association and this does not guarantee the fairness of load among all the APs. Due to non-fairness of load a single AP may get overloaded resulting in a degraded throughput performance. In the traditional Wi-Fi networks, the worst fact is that the wireless device keeps its association with the AP unless it moves far away or the RSSI values diminish due to some AP failure. In the traditional Wi-Fi networks where only RSSI is used for association, the total throughput contribution is just 40% of the total throughput. In the proposed scheme the load among the APs is balanced through the centralized SDN controller and destination AP is chosen only when it is underloaded. In this way the contention is removed and the load fairness among the APs is enhanced.

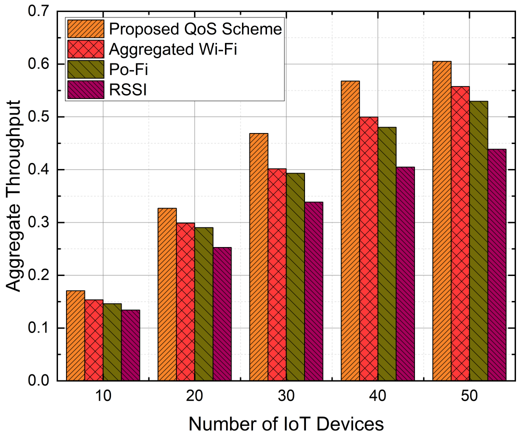


FIGURE 11. Aggregate throughput performance.

The packet delivery rate becomes better in the proposed scheme as the IoT device gets connected to an underloaded AP.

In comparison to RSSI scheme and Po-Fi scheme the aggregated Wi-Fi shows better throughput performance. The Po-Fi scheme is fully centralized which means unnecessary probing frames are used for the southbound APIs and this creates additional overheads leading to time delays. Additional wireless functionalities and forwarding rules induce extra times. Po-Fi relies on light virtual access points (LVAPs) which need extra programmability and time efforts to be created and then shifted alongside the user mobility. In aggregated Wi-Fi scheme the throughput performance is slightly better due to the OpenStack functionality which allows the users to monitor, control and customize the network resources. The use of service orchestrator (XOS) incurs extra costs and time delays for packet processing. The proposed scheme relies not only on the RSSI values but also on the load of the APs. The proposed algorithms are designed specifically to monitor the latency performance and only those destinations APs are chosen for handoffs using SDN which have better throughput performance and guarantee access to the Internet. In the proposed scheme the aggregate throughput is improved by 17%, 13% and 9% when compared to RSSI, Po-Fi and aggregated Wi-Fi schemes respectively.

VII. CONCLUSION

The proposed research makes use of the capabilities of SDN to ensure fairness of load among the OpenFlow enabled APs in the smart city design while reducing the end-to-end packet delay and enhancing the throughput rate by 17%, 13% and 9% when compared to received signal strength indicator scheme (RSSI), Po-Fi scheme and aggregated Wi-Fi scheme respectively. The smart city design incorporates multiple APs with overlapped coverage making a dense Wi-Fi network scenario with mobile IoT devices. The IoT devices connect to the OpenFlow enabled AP to access the internet. The SDN controller has the load balancing applications installed on it which collect the AP reports and make network wide computations to ensure fairness of load among the APs by performing handoffs. The proposed research introduces three algorithms based on service time estimation, M/G/1 analysis and choosing the least loaded AP. The purpose is to maintain the load symmetry among the APs and at the same time reduce the end-to-end delay for a satisfied QoS. Service time-based algorithm choses the IoT device with the highest end-to-end packet delay to be de-associated and then re-associated to a least loaded AP. The algorithm runs on a constant network traffic rate and efficiently handoffs the IoT device. The M/G/1 based algorithm also shifts the IoT device with highest end-to-end delay to a least loaded AP. The algorithm makes use of the random traffic distribution using Poisson process. Extensive experimentation reveal that the algorithm decreases the end-to-end packet delay. The third algorithm is proposed to chose the least loaded AP. The SDN controller based on the traffic load on the AP choses the least loaded AP and while comparing the proposed algorithm to the algorithm that chooses the destination AP randomly, it is deduced that the end-to-end packet delay is significantly reduced. We look forward to introduce artificial intelligence (AI) in the OpenFlow enabled APs to make handover decision themselves till certain load levels. The AI will help in service differentiation and would prioritize packets to the controller which are more delay sensitive hence improving the latency factor.

REFERENCES

[1] Y. Gao and L. Dai, "Random access: Packet-based or connection-based?" *IEEE Trans. Wireless Commun.*, vol. 18, no. 5, pp. 2664–2678, May 2019.

[2] T. Zahid, X. Hei, W. Cheng, A. Ahmad, and P. Maruf, "On the tradeoff between performance and programmability for software defined WiFi networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–12, Apr. 2018.

[3] M. Salman, J.-H. Son, D.-W. Choi, U. Lee, and Y. Noh, "DARCAS: Dynamic association regulator considering airtime over SDN-enabled framework," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20719–20732, Oct. 2022.

[4] S. Manzoor, C. Zhang, X. Hei, and W. Cheng, "Understanding traffic load in software defined WiFi networks for healthcare," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, May 2019, pp. 1–2.

[5] E. Ganesan, I.-S. Hwang, A. T. Liem, and M. S. Ab-Rahman, "SDN-enabled FiWi-IoT smart environment network traffic classification using supervised ML models," *Photonics*, vol. 8, no. 6, p. 201, 2021.

[6] A. Rahman, C. Chakraborty, A. Anwar, M. R. Karim, M. J. Islam, D. Kundu, Z. Rahman, and S. S. Band, "SDN-IoT empowered intelligent framework for Industry 4.0 applications during COVID-19 pandemic," *Cluster Comput.*, vol. 25, pp. 2351–2368, Jul. 2021.

[7] S. Lahlou, Y. Moukafih, A. Sebbar, K. Zkik, M. Boulmalf, and M. Ghogho, "TD-RA policy-enforcement framework for an SDN-based IoT architecture," *J. Netw. Comput. Appl.*, vol. 204, Aug. 2022, Art. no. 103390.

[8] M. Ibrar, L. Wang, N. Shah, O. Rottenstreich, G.-M. Muntean, and A. Akbar, "Reliability-aware flow distribution algorithm in SDN-enabled fog computing for smart cities," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 573–588, Jan. 2023.

[9] S. Misra, S. Pal, N. Ahmed, and A. Mukherjee, "SDN-controlled resource-tailored analytics for healthcare IoT system," *IEEE Syst. J.*, vol. 17, no. 2, pp. 1777–1784, Jun. 2023.

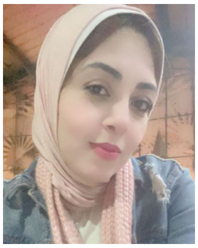
[10] A. M. Farooqi, M. A. Alam, S. I. Hassan, and S. M. Idrees, "A fog computing model for VANET to reduce latency and delay using 5G network in smart city transportation," *Appl. Sci.*, vol. 12, no. 4, p. 2083, Feb. 2022.

[11] A. R. Javed, F. Shahzad, S. U. Rehman, Y. B. Zikria, I. Razzak, Z. Jalil, and G. Xu, "Future smart cities requirements, emerging technologies, applications, challenges, and future aspects," *Cities*, vol. 129, Oct. 2022, Art. no. 103794.

[12] Z. Latif, C. Lee, K. Sharif, and S. Helal, ''SDBlockEdge: SDN-blockchain enabled multihop task offloading in collaborative edge computing,'' *IEEE Sensors J.*, vol. 22, no. 15, pp. 15537–15548, Aug. 2022.
[13] M. Aslam, D. Ye, A. Tariq, M. Asad, M. Hanif, D. Ndzi, S. A. Chelloug, M. A. Elaziz, M. A. A. Al-Qaness, and S. F. Jilani, ''Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT,'' *Sensors*, vol. 22, no. 7, p. 2697, Mar. 2022.
[14] V. M. Martinez, R. S. Guimaraes, R. C. Mello, A. P. do Carmo, R. F. Vassallo, R. Villaca, M. R. Ribeiro, and M. Martinello, ''Make before degrade: A context-aware software-defined wifi handover,'' in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2023, pp. 562–572.
[15] J. Leon, A. Aydeger, S. Mercan, and K. Akkaya, ''SDN-enabled vehicular networks: Theory and practice within platooning applications,'' *Veh. Commun.*, vol. 39, Feb. 2023, Art. no. 100545.
[16] P. P. Ray and N. Kumar, ''SDN/NFV architectures for edge-cloud oriented IoT: A systematic review,'' *Comput. Commun.*, vol. 169, pp. 129–153, Mar. 2021.
[17] M. Kumar and A. Bhandari, ''DDoS detection in ONOS SDN controller using snort,'' in *ICT With Intelligent Applications*, vol. 1. Berlin, Germany: Springer, 2022, pp. 155–164.
[18] R. Deb and S. Roy, ''A comprehensive survey of vulnerability and information security in SDN,'' *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108802.
[19] S. Manzoor, X. Hei, and W. Cheng, ''Towards dynamic two-tier load balancing for software defined WiFi networks,'' in *Proc. 2nd Int. Conf. Commun. Inf. Syst.*, Nov. 2017, pp. 63–67.
[20] B. Gómez, E. Coronado, J. M. Villalón, R. Riggio, and A. Garrido, ''WiMCA: Multi-indicator client association in software-defined Wi-Fi networks,'' *Wireless Netw.*, vol. 27, no. 5, pp. 3109–3125, Jul. 2021.
[21] E. Barka, S. Dahmane, C. A. Kerrache, M. Khayat, and F. Sallabi, ''STHM: A secured and trusted healthcare monitoring architecture using SDN and blockchain,'' *Electronics*, vol. 10, no. 15, p. 1787, Jul. 2021.
[22] Z. Chen, S. Manzoor, Y. Gao, and X. Hei, ''Achieving load balancing in high-density software defined WiFi networks,'' in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2017, pp. 206–211.
[23] A. Al-Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, ''An innovative reinforcement learning-based framework for quality of service provisioning over multimedia-based SDN environments,'' *IEEE Trans. Broadcast.*, vol. 67, no. 4, pp. 851–867, Dec. 2021.
[24] 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.
[25] S. Manzoor, P. Karmon, X. Hei, and W. Cheng, ''Traffic aware load balancing in software defined WiFi networks for healthcare,'' in *Proc. Inf. Commun. Technol. Conf. (ICTC)*, May 2020, pp. 81–85.
[26] H. Zhong, J. Xu, J. Cui, X. Sun, C. Gu, and L. Liu, ''Prediction-based dual-weight switch migration scheme for SDN load balancing,'' *Comput. Netw.*, vol. 205, Mar. 2022, Art. no. 108749.
[27] S. Manzoor, H. Manzoor, S. Rubab, M. A. Khan, M. Alhaisoni, A. Alqahtani, Y. J. Kim, and B. Chang, ''WiMA: Towards a multi-criterion association in software defined Wi-Fi networks,'' *Comput., Mater. Continua*, vol. 75, no. 2, pp. 2347–2363, 2023.
[28] C. Bhatt, V. Sihag, G. Choudhary, P. V. Astillo, and I. You, ''A multi-controller authentication approach for SDN,'' in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2021, pp. 1–4.
[29] M. J. Islam, A. Rahman, S. Kabir, M. R. Karim, U. K. Acharjee, M. K. Nasir, S. S. Band, M. Sookhak, and S. Wu, ''Blockchain-SDN-based energy-aware and distributed secure architecture for IoT in smart cities,'' *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3850–3864, Mar. 2021.
[30] H. Babbar, S. Rani, D. Gupta, H. M. Aljahdali, A. Singh, and F. Al-Turjman, ''Load balancing algorithm on the immense scale of Internet of Things in SDN for smart cities,'' *Sustainability*, vol. 13, no. 17, p. 9587, Aug. 2021.
[31] R. Kumar, V. Venkanna, and V. Tiwari, ''Opt-ACM: An optimized load balancing based admission control mechanism for software defined hybrid wireless based IoT (SDHW-IoT) network,'' *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107888.
[32] S. Manzoor, K. B. Bajwa, M. Sajid, H. Manzoor, M. Manzoor, N. Ali, and M. I. Menhas, ''Modeling of wireless traffic load in next generation wireless networks,'' *Math. Problems Eng.*, vol. 2021, pp. 1–15, Aug. 2021.
[33] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja, and A. K. Bashir, ''Intelligent reward-based data offloading in next-generation vehicular networks,'' *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3747–3758, May 2020.
[34] 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.
[35] H. Manzoor, S. Manzoor, N. Ali, M. Sajid, M. I. Menhas, and X. Hei, ''An SDN-based technique for reducing handoff times in WiFi networks,'' *Int. J. Commun. Syst.*, vol. 34, no. 16, p. e4955, Nov. 2021.
[36] Z. Shi, Y. Tian, X. Wang, J. Pan, and X. Zhang, ''Po-Fi: Facilitating innovations on WiFi networks with an SDN approach,'' *Comput. Netw.*, vol. 187, Mar. 2021, Art. no. 107781.
[37] K. Abbas, M. Afaq, T. A. Khan, A. Rafiq, J. Iqbal, I. Ul Islam, and W. Song, ''An efficient SDN-based LTE-WiFi spectrum aggregation system for heterogeneous 5G networks,'' *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, p. e3943, Apr. 2022.

SOHAIB MANZOOR received the B.S. degree in electrical engineering from the Mirpur University of Science and Technology, Pakistan, in 2011, the M.S. degree in electrical and electronics engineering from Coventry University, U.K, in 2014, and the Ph.D. degree from the Huazhong University of Science and Technology (HUST), China, in 2020. Since May 2012, he has been a permanent Faculty Member with the Department of Electrical Engineering, Mirpur University of Science and Technology. His current research interests include software-defined networking, wireless LANs, and programming. He was a recipient of the best paper award and the best presentation award on numerous international venues. He has won the best final year design project all over Pakistan by Pakistan Engineering Council (PEC). He has a number of competitive research grants to his name.

NAEEM IQBAL RATYAL received the M.S. degree in electrical engineering from the University of Engineering and Technology Taxila, Pakistan, in 2008, and the Ph.D. degree in electrical engineering from the Capital University of Science and Technology Islamabad, Pakistan, in 2016. He is currently a Professor with Electrical Department, Mirpur University of Science and Technology (MUST). His research interests include wireless networking, image processing, and biometrics.



HEBA G. MOHAMED was born in Alexandria, Egypt, in 1984. She received the B.Sc. and M.Sc. degrees in electrical engineering from Arab Academy for Science and Technology, in 2007 and 2012, respectively, and the Ph.D. degree in electrical engineering from the University of Alexandria, Egypt, in 2016. In 2016, she was an Assistant Professor with the Alexandria Higher Institute of Engineering and Technology, Ministry of Higher Education, Egypt. Since 2019, she has been an Assistant Professor with the Faculty of Engineering, Communication Department, Princess Nourah Bint Abdulrahman University, Saudi Arabia. In 2022, she was an Associate Professor in Egypt. Her research interests include cryptography, wireless communication, mobile data communication, the Internet of Things, and computer vision.