Journal of Engineering Research

Volume 6 | Issue 1 Article 9

2022

On Boosting Integrated WLAN & ZigBee Network Performance via Load Balancing

Shahinaz Elkasrawy

On Boosting Integrated WLAN & ZigBee Network Performance via Load Balancing

Shahinaz Helmy Elkasrawy[1], Mohamed E. Nasr [2], Heba Ali El-Khobby[3], Sameh Atef Napoleon [4]

[1] , Communications and Electronics Department, Faculty of Engineering, Tanta University, Tanta, Egypt, eng.shahinaz@gmail.com
[2] Professor, Communications and Electronics Department, Faculty of Engineering, Tanta University, Tanta, Egypt, mohamed.nasr@f-eng.tanta.edu.eg
[3] Associate. Professor, Communications and Electronics Department, Faculty of Engineering, Tanta University, Tanta, Egypt, H.El-khobby@f-eng.tanta.edu.eg
[4] Associate Professor, Communications and Electronics Department, Faculty of Engineering, Tanta University, Tanta, Egypt, s.napoleon@f-eng.tanta.edu.eg

Abstract—**network traffic and overload are constantly increasing. This situation leads to congestion and packet losses at bottlenecks and across the different parts and devices of the network. Luckily, network technologies and techniques are developing rapidly. This paper is dedicated to applying and testing the impact of load balancing mechanisms on network performance. Two networking scenarios are considered: "server on-premise" and "server on cloud."The research takes place in a vast scale network where two of the most popular technologies are spotted in an integrated multiprotocol scenario of Wireless Area networks (WLAN) with the Internet of Things (IoT) ZigBee. Previous studies were concerned by the challenges present due to the very different natures of IoT ZigBee and WLAN networks. This paper presents a better quality of service (QoS) by applying load balancing to these integrated scenarios. Not just that, it also introduces an even better Qos by deploying the rapidly growing popular technology of cloud computing to the same scenario of integrated networks with load balancing. By applying the same data rates with the same timers and networking parameters, network performance is measured and compared to show the difference between previous work without load balancing, and this papers work after deploying load balancing. The research shows whether load balancing has a positive or a negative effect on network performance or does not affect some cases. The network performance parameters under consideration are traffic dropped; traffic received, delay and throughput. Load balancing is tested regarding two different server positions: "on-premise" and "on the cloud."**

Keywords: IoT; Load Balancing; ZigBee; Wireless networks; Cloud computing; Wi-Fi; IEEE802.11; IEEE802.15.4; WLAN

I. INTRODUCTION

The concept of integrated networks and services is widely used and present. Many studies were made to improve the implementation and deployment of devices and data storage and transfer between devices and servers. Some widely adopted trust assessment methods for evaluating the trustworthiness of cloud services have been proposed from different perspectives. A sensor is deployed at the selected remote site and is connected to the gateway router. Communication establishes through the gateway router in either WAN mode or the Ethernet mode. WAN mode helps the gateway device interact directly with the internet, whereas; Ethernet mode helps the gateway device interact with the locally deployed devices. When the communication mode is set, and the cloud server is configured, the communication initiates between the sensors and the cloud server. The established communication allows monitoring the data from any location at ease. IoT Data Server is a "Data Integration Controller" consisting of highly reliable industrial computers and non-programming data integration software. It equips standard data management functions developed especially for data collection, process, saving, notice, and publishing. These functions will help the data management in various scenes from the production cell system to the production line, factory, and cloud system. The servers gather the information that reaches a massive amount in a short time. In such cases, IoT applications can face the challenge of real-time managing/displaying/extracting helpful client information from the whole data stored on servers. Especially in critical situations, a client's database query can take too long. A distinct layer of data processing is used to "cache" fields based on selected or most frequent database queries.

In this paper, two scenarios are being introduced depicting previous work. The networks are about an integrated network of wireless LAN and IoT networks. The data of the said network must be processed and stored by a server. The server in the first scenario is an on-premise server connected to the network via an Ethernet connection through a router. A wireless LAN bridge is used for wireless users, and a Zigbee router is used for IoT devices. The server of the second scenario is an on-cloud server connected via a WAN internet link. In both scenarios, there is a single server and a single router. Previous works and studies always took care of the client-side and the users' clustering and the study of signals between coverage areas. Here, this paper is concerned with the idea of load balancing. The load balancing routing is used under constraints of the quality of transmission (LBRCQT) algorithm [1]. The algorithm was used before but for wireless meshed networks only. In this paper some modifications were made to the original algorithm, it is used for integrated wireless LAN and ZigBee IoT networks. Two load balancing solutions are being taken care of. First, there is a whole server farm to deal with the data of both the WLAN and the IoT users. Second, many routers connect the different users to the server farm for load balancing and redundancy. The results section will introduce a comparison to show the difference between deploying load balancing in the network, the paper's main idea, and the network's performance without using any load balancing like previous related work.

The Internet of Things (IoT) explains the network of objects—"things"—embedded with sensors, programs, and other various technologies to connect and send data between devices and systems on the internet Figure 1[2].

Zigbee is a wireless technology that is an open global standard to deal with the needs of low-cost, low-power wireless IoT networks. The Zigbee standard works on the IEEE 802.15.4 standard and operates in unlicensed frequency bands, 2.4 GHz, 900 MHz, and 868 MHz Zigbee architecture Figure 2 [3]

Message Queue Telemetry Transport (MQTT) protocol is an open-source application layer protocol based on Transmission Control Protocol (TCP). It works as publish/subscribe model with asynchronous connections. It has a minimal overhead (2 bytes header), comparable to the client/server model. It is simple to implement and recommend smart applications like smart hospitals, smart homes, smart schools, etc. MQTT improves the operation of high delay and low bandwidth networks. The Broker component of the MQTT is needed to allow connections between clients (publishers and subscribers) [4].

A WLAN allows users to roam through the coverage area. Wi-Fi or WLAN uses high-frequency unlicensed radio waves for linking the nodes. There are various standards of IEEE 802.11 WLANs. The most popularly used are 802.11, 802.11a, 802.11b, 802.11g, 802.11n, and 802.11p. They all use carrier-sense multiple-access collision avoidance (CSMA/CA) [5]. Figure 3 shows a sample wireless topology with its components.

Fig. 1. IoT things interaction

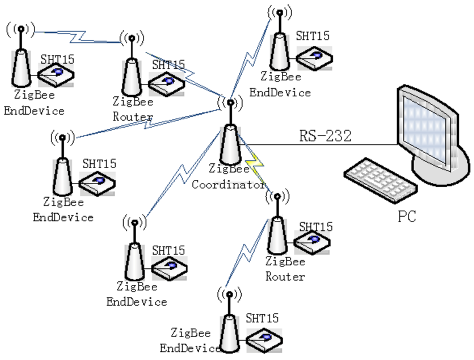

Fig. 2. Sample Zigbee topology

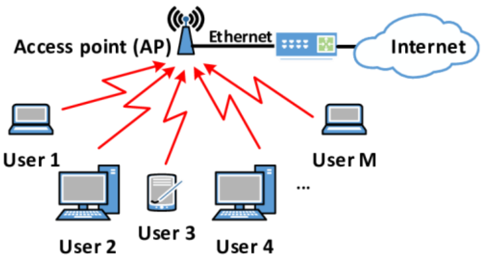

Fig. 3. Sample WLAN network

Load balancing is the process of enhancing performance by distributing traffic among several server pools, Figure 4. There are only some load balancers that support the MQTT protocol. High Availability Proxy (HAProxy) open-sourced was used in this paper. This section clarifies two main load balancers used in this paper: HAProxy and Server Load Balancing (SLB) router.

The HAProxy is open source software used to distribute load among multiple servers based on TCP. HAProxy consists of two lists: a Back-end list that contains servers that receive messages from the front-end list. This list can be defined by: the IP address, port, and load balancing technique used. The other is the Front-end listing that represents messages from clients. This listing can be defined by: IP address and port numbers of clients and Access Control List (ACL). ACL is applied to provide some rules to permit or deny messages arriving from clients to servers [6]. HAProxy uses health checks to monitor the availability of back-end servers. Health check works by establishing a TCP connection between the HAProxy and the back-end servers. If one of the servers fails to process messages from the front-end, then HAProxy removes the server from the back-end list [7]. HAProxy provides redundancy by adding two HAProxys and acts as an active or passive model using Keepalives to prevent the load balancer gets overwhelmed with a massive number of messages because of a single point of failure. Keepalives is software used for routing messages; it works by creating a shared virtual IP address between two HAProxys. Figure 4 shows the main load balancing concept.

In this paper, server load balancing (SLB) is used because of its suitability for the present solution, which is integration between internet of things (IoT) and wireless local area networks (WLAN) networks with an attempt to optimize the network performance. These load balancers issue messages from multiple clients to several servers simultaneously according to a specific technique to reduce the traffic on a single server. Load balancing techniques can be classified into several types [8]; the most popular techniques are as follows:

1) Round Robin (RR) This method distributes the traffic among the servers sequentially.
2) Weighted Round Robin (WRR) is similar to RR; however, some servers have more capabilities than others. Therefore, weights can be distributed to the servers. For example, if there are two servers: the first server receives five messages (because it is weighted to

5), the second server receives one message (because it is weighted to 1).

3) Least Connection (LeastConn) method chooses the server with the least number of active connections. Some servers have more overloads than others because clients connect to servers much longer than others. So, server 3 receives three messages. Both servers 1 and 3 now have the same number of active connections. Then, the load balancer performs RR on servers 1 and 3 until the number of active connections in both servers reaches 20. Thence, the load balancer performs RR on three servers 1, 2, and 3, and so on.

Cloud networking is a form of information technology (IT) infrastructure in which some or all of an establishment's network capabilities and resources are present in a public or private cloud platform, managed on-premise or by a service provider, and available on demand[9]. Figure 5 shows cloud deployment and communications with the things network.

The research also presents different solutions and scenarios based on different network requirements and challenges. The first two scenarios represent previous work with its drawbacks. The last two represent the proposed solution.

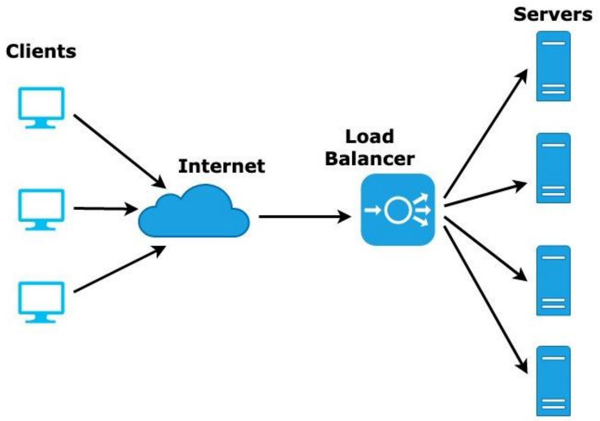

Fig. 4. Load balancing Concept

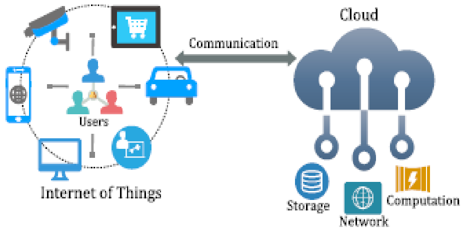

Fig. 5. Cloud to things deployment

- IoT and WLAN integrated network with server on-premise without load balancing
- IoT and WLAN integrated network with a server on the cloud without load balancing
- IoT and WLAN integrated network with server on-premise with load balancing
- IoT and WLAN integrated network with a server on the cloud with load balancing

The paper presents previous implementations and introduces the newly developed implementations by adding the load balancing techniques to the scenarios and compares the old topologies to the new ones. It also compares the new topologies deploying the server on-premise to deploying the server on the cloud using load balancing in each case.

First, previous and related work is discussed. Then, the system model is introduced with the current work improvement and equations. The results section comes next with detailed representations and tables. Then paper conclusions and references. Throughout the paper, the Riverbed simulator is used to implement and measure the network results and shows the difference between previous implementations and this paper's introduced implementation.

II. RELATED WORK

The topic of integrating ZigBee with WLAN networks was handled before. The main concern of previous studies was to show the benefits of such integration and try to overcome challenges and drawbacks. ZigBee is low power and low-rate network, and WLAN is for high rate networks. It was found that integrating those yields better network performance as some devices and network parts need ZigBee, and some work better with WLANs. They proposed using traffic prioritization of 802.15.4 nodes by implementing two groups of 802.15.4 node classes that differ by the minimum contention window value [10] [11].

This paper's primary concern is huge enterprises as vast amounts of data are sent and received around the clock. Handling low rate and high rate data is a good thing. Nevertheless, large enterprises do not accept bottlenecks, collisions, drops, delays, and jitters. Hence came the addition of load balancing to the topology. First, several servers and routers were added inside the enterprise for load balancing, called an on-premise solution. This addition improved the network performance but increased the cost considerably. The better solution was using several servers and routers for load balancing but through the cloud. It was the best solution concerning network parameters. However, internet and cloud connections must be considered besides the added cost for using several servers and connections.

As shown in Figure 6, the first problem in this scenario is that no load balancing is used. The router and the server represent single points of failure. They also represent bottlenecks in the network, which may increase the delay in the network and data loss. Server on-premise topologies suffer from many problems. The least are cost, monitoring, troubleshooting, and limited resources. On the other hand, the deployment and configuration are relatively simple.

As shown in Figure 7, Single points of failure include the single on-premise router connected to the internet. The presence of the server on the cloud instead of on-premise has the advantage of fewer IT skills required and less cost for technical requirements and resources. However, also it has some disadvantages like delay and loss; especially it is a single server for the whole process. The IoT paradigm has expanded the possibility of using sensors universally, particularly if connected to a cloud service for data sharing.

There are many ways to connect sensors to the cloud: wearable or moveable devices often lean on a Smartphone that performs as a gateway, while other sensors, such as smart sensors for constant monitoring (e.g., fall detectors) are connected through wireless networks covering a limited area (e.g., ZigBee or Wi-Fi).

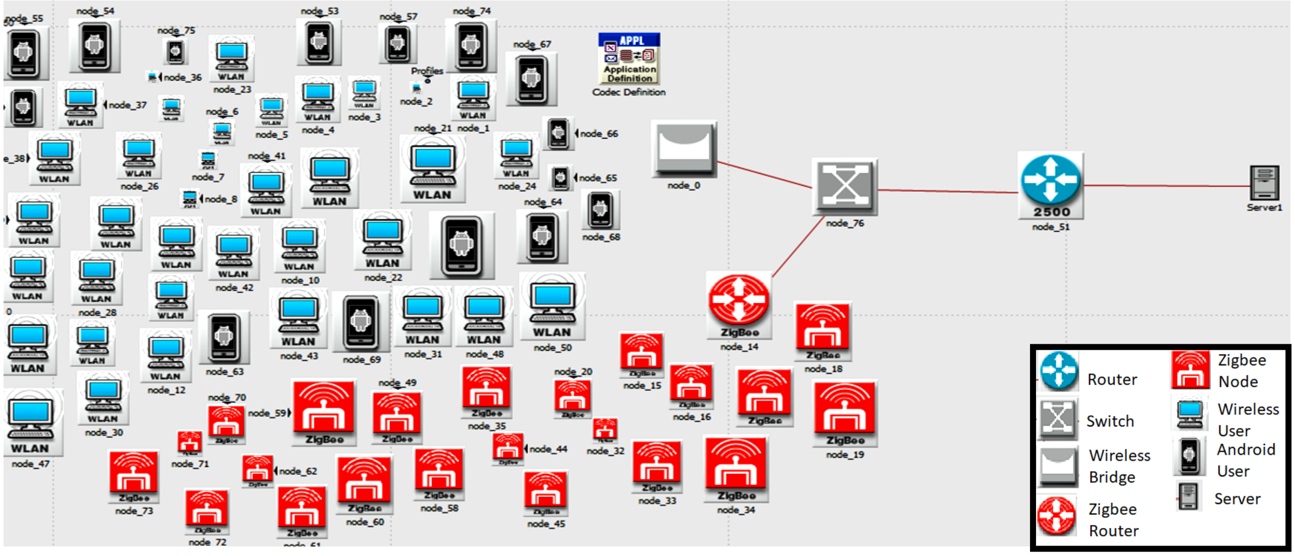

Fig. 6. Server on-premise, no load balancing

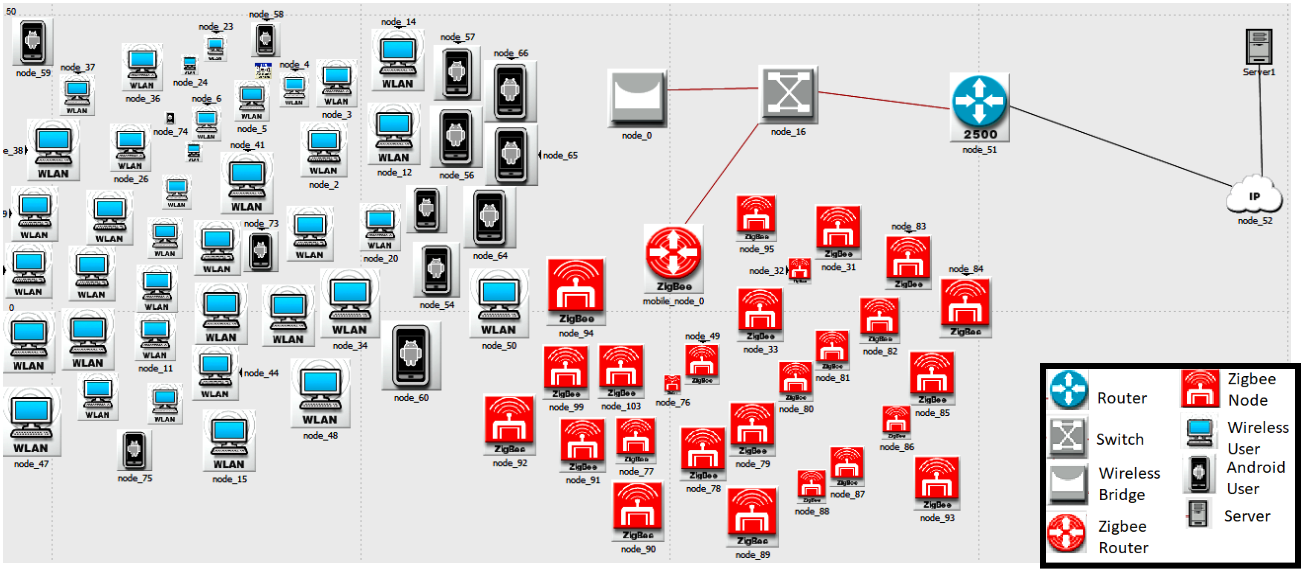

Fig. 7. The server on the cloud, no load balancing

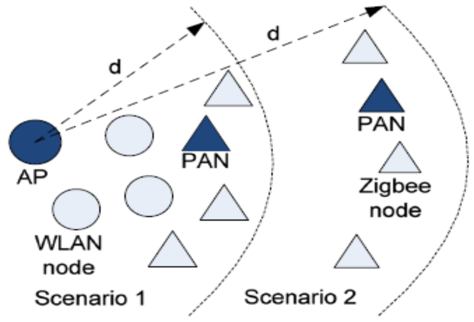

Fig. 8. Coexistence scenarios between WLAN and ZigBee network

Since the transmission power of the IEEE 802.11 WLAN node is much larger than that of IEEE 802.15.4 ZigBee nodes, one network affects the other would depend on the relative distance between points in the two networks. Two scenarios are considered. In the first scenario, any node of either network can perfectly sense and detect the transmissions of any other nodes. It would be the case as the distance between WLAN and ZigBee nodes is small (distance d is small). For the second scenario ZigBee points can sense and detect the flows of WLAN nodes, but the WLAN nodes cannot detect the flows of ZigBee nodes. It would happen as the distance d between WLAN nodes, and ZigBee nodes are sufficiently large [12].The considered coexistence scenarios are illustrated in Figure 8.

By representing the position of each node in the (WLAN ,ZigBee) pair by a single variable, a much more manageable model is attained. Modelling the IEEE 802.15.4 MAC protocol using a one-dimensional model has been adopted; however, analyzing the coexistence act of the two networks is far more challenging and still an open problem [13].

ZigBee performance under Wi-Fi interference is simulated. A model has-been introduced which reflects the ZigBee and Wi-Fi coexistence. Results show that ZigBee may be severely interfered with by Wi-Fi and that a Safe Distance and Safe Offset Frequency can be identified to guide ZigBee deployment. It is shown that a distance of 8 m between ZigBee and Wi-Fi is a safe distance that can guarantee reliable ZigBee performance regardless of the offset frequency. It is also shown that 8 MHz is a safe offset frequency even when the distance is 2 m. The algorithm improves the ZigBee performance to provide reliable service in coexistence with Wi-Fi networks [14].

For telecommunications networks in general, load balancing routing is often used to reduce bottlenecks in the network. Based on the number of used routing paths, load balancing routing algorithms are classified into two main types: single-path routing and multiple-path routing [15]. For the single path routing, the route cache of each node stores only one path to the destination node. This path is used for data transmission. Therefore, the load balancing must be done during route discovery. For this method, the authors of [16] have proposed a load balancing routing algorithm for the network by modifying the discovery principle of the ad hoc on-demand distance vector (AODV) protocol. To discover a new route by AODV protocol, the source node broadcasts the route request packet (RREQ) to all its neighbors. At each node receiving RREQ, if this RREQ has already been received, delete the RREQ. Otherwise, return the route reply packet (RREP) if its route cache has a route to the node destination; else, it forwards the RREQ packet to its neighbors, except the origin node.

This process repeats until a route has been found. To balance the traffic load, the authors of [16] have improved the process of processing RREQ packets. When an intermediate node receives an RREQ packet from another node, it will check its current load based on the status of its buffer. If the current load exceeds the defined maximum, the node discards RREQ. Otherwise, RREQ is forwarded to the following nodes. The found routes will not go through nodes with heavy traffic for this method. By simulation methods, it was found that this method improves network performance regarding delay and traffic drops but decreases the Quality of Transmission (QoT).

Another method often used for single-path load balancing routing is to build load-aware metrics. The authors of [17] have proposed a routing metric for this method, namely weighted cumulative expected transmission time with load balancing (WCETT-LB) for wireless networks. The simulation results have shown that this method outperforms the previous ones regarding packet transmission ratio, average end-to-end delay, and congestion level. However, these methods do not present any diversity concerning the position of the servers regarding control, management, or storage. Also, previous studies are concerned with one topic at a time. It is either studying the effect of integration on network performance or load balancing on network performance.

This paper introduces the challenges of integrating multiprotocol in the same network and deploying different load balancing techniques to enhance integrated networks' performance through load balancing as multiprotocol networks are widely used and developed nowadays.

In the results section, a comparison is made to show the difference in performance between the related work and the proposed work after applying load balancing to the scenarios.

III.SYSTEM MODEL AND PROPOSED WORK

In order to improve the performance of the network in terms of the QoT and congestion probability, a route routing algorithm is proposed, namely load balancing routing under constraints of quality of transmission (LBRCQT) [1]. This method uses the principle of SDN to collect the information of QoT and traffic load of the links in the network.

The primary purpose of the proposed modified algorithm is to choose a load balancing route under the constraints of QoT, where the load balancing is done by selecting a route in which a data packet is transmitted over that route, the blocking probability is minimal. The constraints of QoT include Signal to Noise Ratio (SNR) and End to End Delay (EED).

A. Simulation Scenarios

A WLAN Ethernet bridge connects the wireless hosts and users in all scenarios. The WLAN users are a group of fixed PCs and mobile android users. IoT Zigbee end devices are connected to a Zigbee router in all scenarios. The number of users in all scenarios is nearly the same to ensure the validity of the results. The Ethernet Bridge and the Zigbee router are connected to an Ethernet switch. The Ethernet switch connects the whole network of different technologies to the router(s) and then to the rest of the network, depending on the scenario discussed later. The IoT and WLAN devices are stationary in the network plane. The deployment of IoT and WLAN devices is random. The simulation runtime is 1 hour. The time in the graphs is shown in minutes.

It is considered an 802.11-based network co-located with an802.15.4-based network and shares the same spectrum band. The most critical assumptions and approximations are herein summarized: (1) 802.11 nodes(resp. 802.15.4 nodes) can detect each other transmissions if they are in their detecting range; (2) 802.15.4 nodes packets are of the same size, and 802.11 packets are also of the same size; (3) all the 802.15.4nodes are time-synchronized with the coordinator's beacon; it is considered only direct transmission and the coordinator does not acknowledge the reception of the packets; (4) it is considered that nodes always have a packet ready for transmission; (5) It is considered only one type of priority class of 802.15.4 nodes for each analysis; (6) It is assumed ideal channel conditions, i.e., a failure transmissionoccurs only upon collisions.

As shown in Figure 9, a server farm was deployed to ensure load balancing, redundancy, and failover. Several routers are present for load balancing and backup. This model decreases delay and loss as there are many points of backup and redundancy. The main problem here is the cost of hardware and deployment. Also, maintenance and management will be an issue. The routers and servers add many advantages concerning throughput, reliability, delay, loss, and network utilization but the cost and management need enhancements.

As shown in Figure 10, a server farm is used but from the cloud. The servers are used for load balancing, backup, and failover. Several routers are used to connect to the cloud. Multiple wan links are connecting the enterprise to the cloud. All the enterprise devices can be managed through the cloud. The cloud simplifies management, maintenance, and migration. The network performance is enhanced concerning delay, throughput and loss. There are no single points of failure in this scenario.

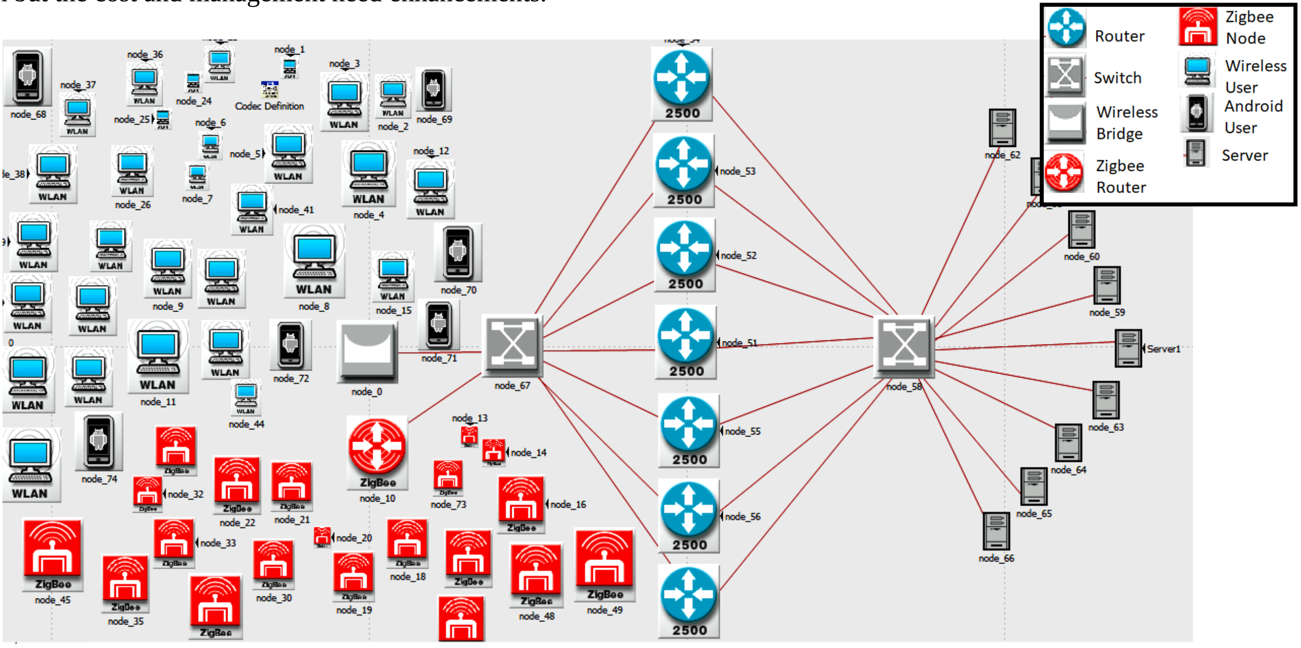

Fig. 9. Server on-premise, with load balancing

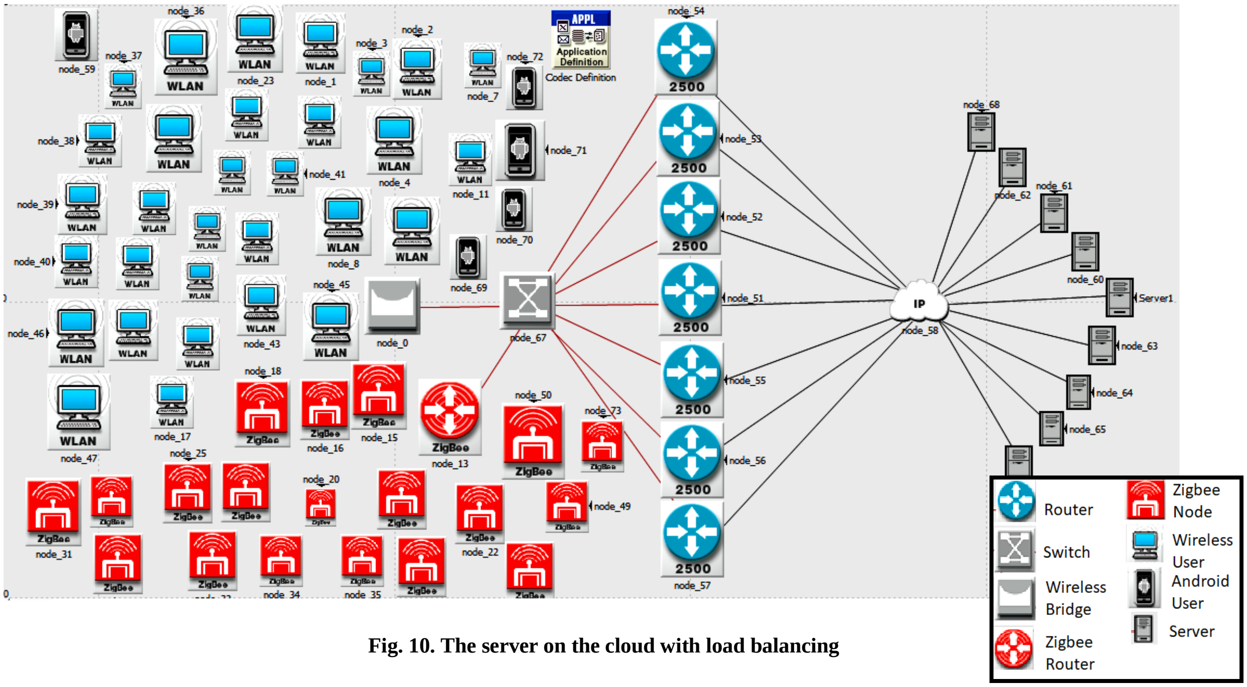

Fig. 10. The server on the cloud with load balancing

The network performance and the interference influence are calculated in Packet loss ratio (PLR), defined in (1).

$$\mathbf{PLR = L_r / L_t} \quad \mathbf{(1)}$$

Where, L_r is number of lost packets and L_t is Total number of transmitted packets.

The PHY layer of IEEE 802.15.4 at 2.4 GHz uses Offset quadrature phase-shift keying (OQPSK) modulation. For an additive white Gaussian noise (AWGN) channel, the BER can be calculated by the following equation (2) [22]:

$$BER = Q\left(\sqrt{\frac{E_b}{N_o}}\right) \qquad (2)$$

Equation 1 shows the BER (Bit Error Rate), Where E_b is the signal energy per user data bit. N_o is the noise spectral density. E_b/N_oindicates the power efficiency of the system without considering modulation type, error correction coding, or signal bandwidth.

The throughput attained by each ZigBee and WLAN node can be expressed as shown in equations 3 and 4, respectively [13].

$$\boldsymbol{S_z = \frac{T_z n_z \tau_z (1-\tau)^{n-1}}{E[S_{t_w}]}} \qquad \mathbf{(3)}$$

$$\boldsymbol{S_w = \frac{T_s n_w \tau_w (1-\tau)^{n-1}}{E[S_{t_w}]}} \qquad \mathbf{(4)}$$

Where $S_z$the throughput of ZigBee node is, S_w is the throughput of WLAN nodes. The probability that one node (WLAN or ZigBee node) tries to transmit in a generic backoff slot τ. The settle time of the states in which the ZigBee nodes is in the transmission states T_z, and the settle time of the states in which the WLAN node is in the transmission states T_s.The probability that a WLAN node tries to transmit in a generic slot τ_w. The probability that the ZigBee node tries to transmit in a generic slotτ_z. n = n_w + n_z where n_w and n_z denote thenumber of WLAN and ZigBee nodes, respectively. The estimated length of a generic WLAN slot E[S_{t_w}].

Packet loss is the number of packets dropped to reach the destination server when travelling through the network. Packet loss is calculated using equation (5). Where P_L is Packet Loss, O_L is the Offered Load of packets sent by things, and P_R is Packets Received by destination [14].

$$\mathbf{P_L = O_L - P_R} \quad \mathbf{(5)}$$

LBRCQT algorithm [1]:

```
Input: A network topology with current load traffic of all; a request of discovering a new path from the source node (s) to the destination node (d).
Output: A load balancing route satisfying the constraints of QoT from s to d or rejection of the request if a new route is not found.
Method:
(1) Determine packet blocking probability of all links.
(2) Determine all possible paths from node s to node d (K routes);
(3) Determine SNR of all possible routes $\beta_{sd}^{(r_k)}$ , $k = 1..K$.
(4) Determine EED of all possible routes $\tau_{sd}^{(rk)}$, $k = 1..K$
(5) Determine the route from s to d.
(6) send the control packets to the points to update the new path into the flow switch table;
(7) Else, Send the control packets to the nodes to reject the request of the data transmission;
(8) End
```

Some modifications were made to the LBRCQT algorithm to make it more suitable for the integration of IoT and Wi-Fi networks over the cloud.

The proposed modified algorithm's main objective is to choose a load-balancing route while maintaining Quality of Transmission QoT standards [1]. Software Defined Networks (SDN) and Ternary Content Addressable Memory (TCAM) were used for route storage. Also, Time Division Multiple Access (TDMA) was used for slot allocation for the different node types.

The proposed modified algorithm steps are given below:

Proposed Modified algorithm:

```
Input: a network architecture with all links' active load traffic $\rho = [\rho_{ij}]_{nxn}$; a request to find an alternative route connecting the source node(s) to the destination node (d).
Output: A load-balancing route that satisfies the QoT restrictions from s to d or, in the absence of a new route, denial of the request.
Method:
Stage 1: In case of no Multipath Determine the path without Load Balancing.
Stage 2: If there are multipath and Load Balancing the following steps are applied.
    (1) Determine whether the communication is from a ZigBee or Wi-Fi node.
    (2) Prioritize ZigBee transmission over Wi-Fi transmission.
    (3) Use Time Division Multiple Access (TDMA) to allocate a suitable time slot.
    (4) Determine the likelihood of packet blocking $B_{ij}$ on all links (Bij) according to equation (6) [18]
```

$$\mathbf{B_{ij}} = \begin{cases} \dfrac{\boldsymbol{\rho^k{}_{ij}\ (1-\rho_{ij})}}{\boldsymbol{1-\rho^{k+1}{}_{ij}}} & \textbf{if } \boldsymbol{\rho_{ij} \neq 1} \\[2ex] \dfrac{\mathbf{1}}{\mathbf{k+1}} & \textbf{otherwise} \end{cases} \qquad (6)$$

```
        Where k is the number of packets in the queue and $\rho_{ij}$ is the traffic density.
    (5) Find every route that could lead from node s to node d (K routes);
    (6) Find the Signal to Noise Ratio (SNR) for each potential route. $\beta_{sd}^{(r_k)}$ , $k = 1..K$ according to equation (7) [19,20]
```

$$\boldsymbol{\beta_{sd}{}^{(r)}} = \begin{cases} \min\limits_{\forall l_{ij} \in r_{sd}} \left(\beta_{ij}{}^{(l)}\right) & \textbf{if DF is used} \\[2ex] \left(\sum\limits_{\forall l_{ij} \in r_{sd}} \dfrac{1}{\beta_{ij}{}^{(l)}}\right)^{-1} & \textbf{otherwise} \end{cases} \qquad (7)$$

```
    Where $l_{ij}$ is for the link, $\beta_{sd}$ is the SNR of the route $r_{sd}$ and DF is for decode and forward.
    (7) Identify the EED for each potential path. $\tau_{sd}^{(rk)}$, $k = 1..K$according to equation (8) [21]
```

$$\boldsymbol{\tau_{sd}{}^{(r)} = \sum_{\forall l_{ij} \in r_{sd}} \tau_{ij}{}^{(l)}} \qquad (8)$$

```
    Where $\tau_{sd}^{(r)}$ and $\tau_{ij}^{(l)}$ are the EED of the route $r_{sd}$ and the link $l_{ij}$ , respectively.
    (8) Map out the path from s to d.
    (9) Store the routing strategy information in SDN TCAM.
    (10) Else if TCAM is full, use MAC with a virtual time slotted schedule.
    (11) Else, Send the node control packets to refuse the data transfer request;
    (12)End
```

V. PERFORMANCE EVALUATION

In Figure 11, large amounts of traffic represent different protocols at once that simulate a real-life scenario. In the upcoming figures, a set of results will be displayed different comparisons between server on-premise and server on the cloud with and without load balancing, server on-premise with and without load balancing, and server on the cloud with and without load balancing. Then the pros and cons of every scenario will be discussed.

Table 1 shows the traffic rates in gigabytes and terabytes.

A Simulation model has been configured, showing the ZigBee and Wi-Fi coexistence. Simulation results show that ZigBee may be affected by Wi-Fi and that a certain distance and specific offset frequency can be adjusted to guide ZigBee deployment. It is shown that a distance of 8 m between ZigBee and Wi-Fi is a safe distance that can guaranteereliable ZigBee performance regardless of the offset frequency. It is also shown that 8 MHz is a safe offset frequency even when the distance is just 2 m. The presence of a load balancing server on the cloud improves the overall network performance compared to on-premise.

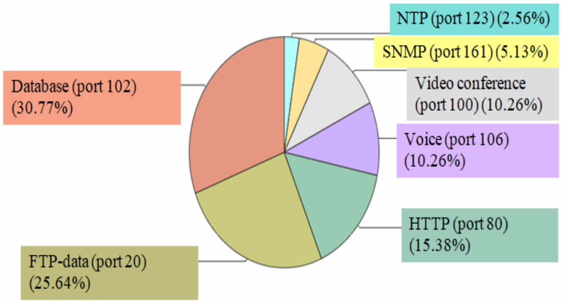

Fig. 11. Various protocols at once results

Table 1. Flow Details

Objects with traffic	
Number of rows	504
Traffic volume statistics	
Average volume per-flow	11.675 GB
Total volume on flows	5.746 TB
Average volume	11.675 GB

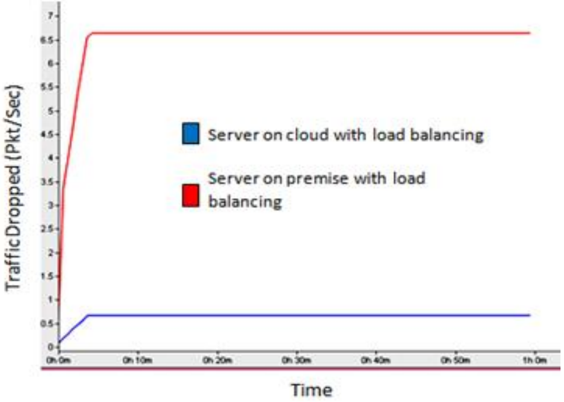

Fig. 12. Comparison between traffic dropped results with respect to time in minutes in case of server on cloud and on-premise with load balancing applied.

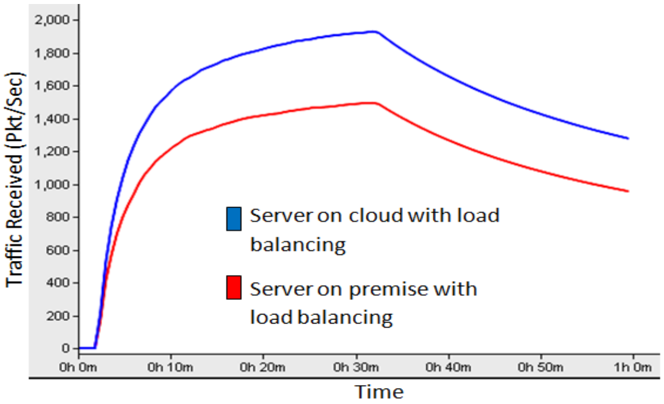

Fig. 13. Comparison between Traffic received by one of the load balancing servers with respect to time in minutes in case of server on a cloud vs. server on-premise with load balancing applied.

Traffic dropped in the case of server on-premise is much larger than traffic dropped in the server on-premise, and load balancing is used in both cases.

Figure 12 shows that the traffic dropped in the case of server on cloud in case of load balancing is 0.75 Pkt/sec, while in case of server on-premise without load balancing; it is 6.6 Pkt/sec. In the case of servers on the cloud, the traffic dropped decreased by 88.63%, a significant improvement in network performance.

Figure 13 shows that traffic received in the case of server on-premise with load balancing is 1700 Pkt/sec, while in the case of server on the cloud with load balancing, it is 2400 Pkt/sec. Load balancing and server on cloud deployment increased the received traffic by 29.1%.

Traffic received by a server residing on the cloud is much improved than in the case of the server on the enterprise location.

The obtained results in Fig.13 show the traffic received of all data transmission channels for the case that the Proposed Modified algorithm used. According to the simulation scenarios presented in the minimum required traffic received for ensuring QoT is 600 Pkt/sec.

Table 2 shows a significant difference between traffic received in server on-premise and server on the cloud after deploying load balancing.

Calculating the packet delivery ratio PDR is very important. The ratio between the total numbers of packets received by the server, whether on-premise or on the cloud, and the total number of packets transmitted by all the nodes, whether they were Zigbee or Wi-Fi nodes. Traffic is sent on average of 11GB per flow, and the total volume on flows is 5TB. The duration of flow is 3600 sec.

Next, the packet delivery ratio (PDR) is analyzed. It is an important performance parameter of the network system. The difference in the PDR proposed modified algorithm is shown in Figures 13, 14, 16, and 18, where PDR to time is plotted, expressed in pps. It is observed that, for the proposed modified algorithm, PDR has increased significantly for servers on the cloud compared with that of servers on-premise.

Figure 13 shows that traffic received in the case of server on-premise with applying load balancing is 80000 Pkt/sec, and in case of server on the cloud, it is 50000 at the same point in time. So, to achieve better results without deploying an on-cloud server, load balancing on-premise is a perfect solution with even better results. The delay in the server on cloud and server on-premise with load balancing in both cases is nearly identical. However, the delay is considerably better with load balancing than without load balancing. The load balancing algorithm enhances the delay results by 37.5% in the case of server on cloud and by 25% in the case of server on-premise. Also, the results show that the delay has better overall results in the case of server on-premise than for server on cloud. Also, if load balancing was removed from the equation, traffic received in the case of server on cloud is better than that received by an on-premise server. One attribute for that is the specs and qualities of on-cloud servers compared to on-premise servers.

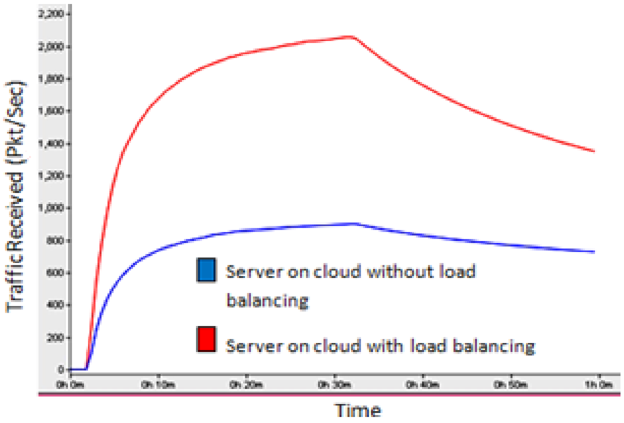

Fig. 14. Comparison between traffic received by one of the load balancing servers with respect to time in minutes in case of server on the cloud once without load balancing and then after applying load balancing.

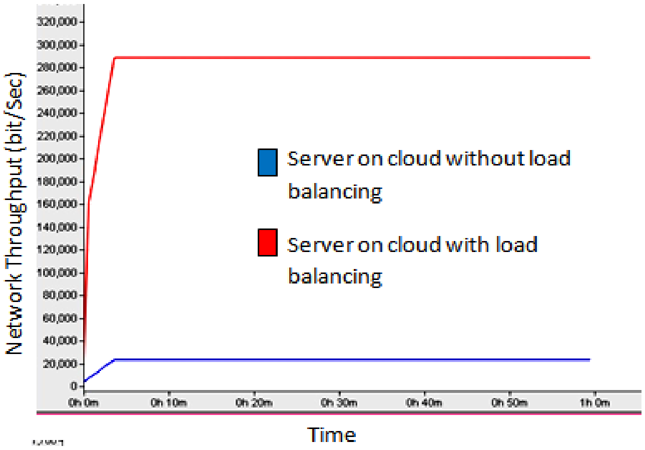

Fig. 15. Comparison between Network throughput results with respect to time in minutes in case of server on the cloud without load balancing and then after applying load balancing.

Figure 15 and Figure 17 show the network throughput in server on-premise and server on the cloud with and without load balancing. It is noticed that the server's location does not affect the throughput as in both cases, on cloud and on-premise, the throughput is 30000 bit/sec, and after deploying load balancing, it increases to be 290000 bit/sec. While the server's location does not affect the network throughput, load balancing affects the performance massively as an increase of 89.65% is present. Before applying load balancing, deploying the server on the cloud increases the traffic received by 32%. After applying load balancing, traffic received increases again by 61.4%. That is a substantial overall improvement in network performance, referring to Tables 3 and 4. In some scenarios, deploying a server on the cloud is not available or applicable.

A considerable difference is noted in the network throughput when comparing load balancing results to without load balancing results. Load balancing adds various servers to the topologies, so many servers share and process the network data instead of one server carrying the whole load.

Also, the same is noted when comparing server on-premise results in cases with and without load balancing. Traffic received is much improved when deploying load balancing in the network. The throughput in the case of using the proposed modified algorithm also increases for cloud servers compared to that without load balancing. It is more clearly visible from Figures 15 and 17.

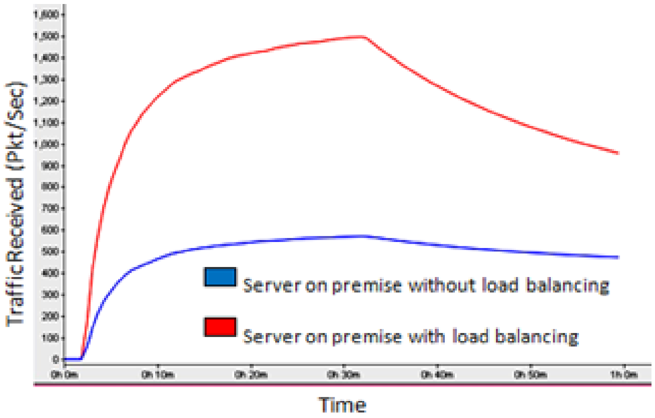

Fig. 16. Comparison between traffic received results with respect to time in minutes in case of server on cloud and on-premise without load balancing applied and then after applying load balancing.

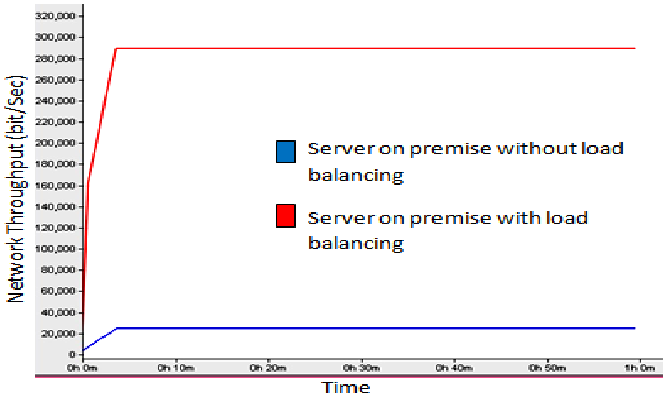

Fig. 17. Comparison between Network throughput results with respect to time in minutes in case of server on-premise without load balancing applied and then after applying load balancing.

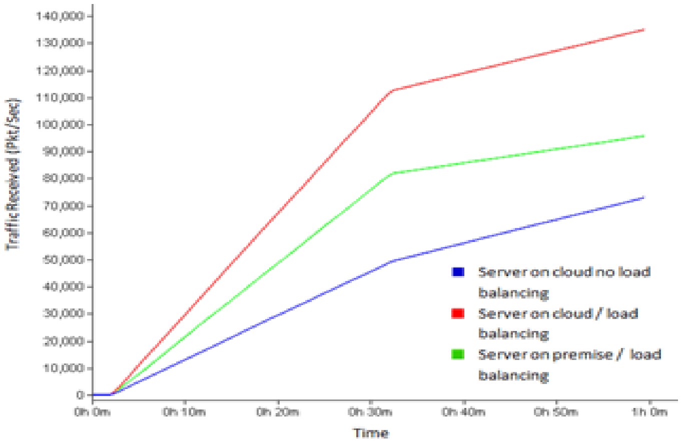

Fig. 18. Comparison between Networks received traffic results with respect to time in minutes in case of server on-premise with load balancing applied and server on the cloud with and without load balancing applied.

In the next section, we compare the average EED of the proposed modified algorithm for an integrated network. The obtained results are shown in Figures 19 and 20; the EED depends mainly on the transmission delay because the queue delay is minimal. In the case of the heavy traffic load, the average EED of proposed modified algorithm reduces significantly compared with that without load balancing.

The proposed modified algorithm causes it has reduced the queue delay at each node due to the load balancing.

Thus, the simulation results on average EED have shown that when the traffic load in the network is heavy, the proposed modified algorithm performs more efficiently than without load balancing in terms of the average EED.

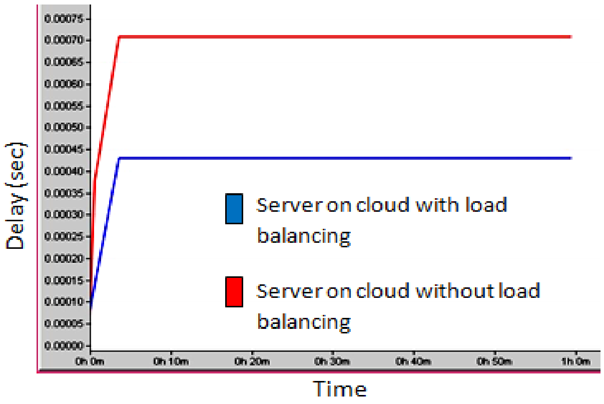

Fig. 19. Comparison between network delay results with respect to time in minutes in case of server on the cloud with and without load balancing applied

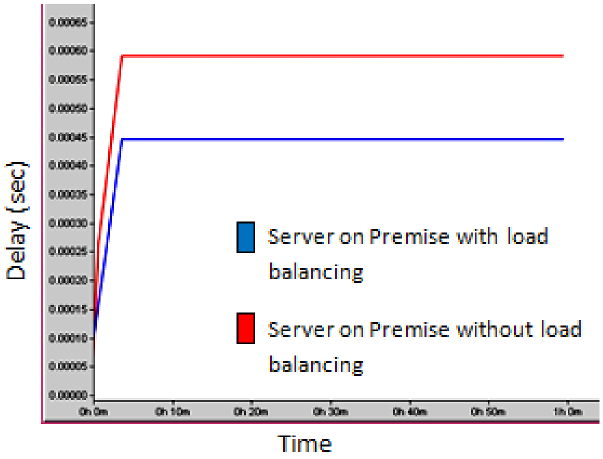

Fig. 20. Comparison between network delay results with respect to time in minutes in case of server on-premise with and without load balancing applied

Based on the simulation results presented above, we can conclude that the proposed modified algorithm significantly improves the network performance in terms of SNR, PDR, and throughput, mainly applied while the server is on the cloud. The cause of the improvement in network performance is that since the proposed modified algorithm has chosen load balancing routes, the bottleneck in the network is reduced. In addition, the constraint conditions of QoT have also been considered, reducing the blocking probability of the data packets due to QoT being unguaranteed.

Table 2. Comparison between network performances in case of server on-premise and server on the cloud with the deployment of load balancing

Point of comparison	Server on-premise (Load Balancing)	Server on cloud (Load Balancing)
Traffic Dropped at 30m of flow(Pkt/sec)	6.6	0.75
Traffic Received at 30m of flow(Pkt/sec)	1700	2400
Delay (sec)	0.00045	0.00045

Table 3. Comparison between network performances in case of server on the cloud with and without the deployment of load balancing

Point of comparison	Server on cloud (Load Balancing)	Server on cloud (Without LB)
Traffic Received at 30m of flow(Pkt/sec)	2100	810
Throughput (bits/sec)	300000	29000
Delay (sec)	0.00045	0.00072

Table 4. Comparison between network performances in case of server on-premise with and without the deployment of load balancing

Point of comparison	Server on-premise (LB)	Server on-premise (Without LB)
Traffic Received at 30m of flow (Pkt/sec)	1550	550
Throughput (bit/sec)	300000	29000
Delay (sec)	0.00045	0.0006

VI. Conclusion

In integrated networks, load balancing routing is one of the best routing techniques to improve network performance. With this routing technique, the local congestion at some connections and intermediate nodes is minimized because the traffic is distributed evenly for all connections in the network. However, in the case of the integrated networks with a large area and high node and user density, the load balancing routing and server on cloud deployment enhances the QoT, and the end-to-end delay since the data transmission routes can pass through multiple hops. Load balancing techniques or an integrated network of WLAN and IoT were proposed in this paper. The proposed algorithm is based on load balancing and cloud networking.

The performance of the proposed modified algorithm is demonstrated by the simulation method using Riverbed. The simulation results have shown that the proposed algorithm can improve network performance in terms of SNR, packet delivery ratio (PDR), and throughput compared with previous algorithms that focused on integration solely or the topic of load balancing alone.

When applying the same data with the same timers and parameters to the different network topologies presented in this paper, it was noted that network throughput is at its best when using a server on the cloud with load balancing. Network delay varies depending on the quality of the WAN connection used and the number of source nodes. Traffic received is at its best using a server on the cloud with load balancing. In all cases, using load balancing enhances network performance considerably. Server on-premise with load balancing has excellent results, as shown in the results section with graphs and numbers, but it needs high maintenance and a good IT team always present on site. The server on the cloud has the best performance regarding the results section; it shows excellent enhancements concerning packets received, delay, throughput, and SNR, but it includes the cost of the cloud services and WAN connections needed to ensure a real network enhancement. Several parameters can affect the network performance, especially the distance between the nodes and the integration of different technologies at once.

VII. References

[1] L. H. Binh and T. -V. T. Duong, “Load balancing routing under constraints of quality of transmission in mesh wireless network based on software-defined networking,” in *Journal of Communications and Networks*, vol. 23, no. 1, pp. 12-22, Feb. 2021

[2] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, “Internet of Things forensics: Recent advances, taxonomy, requirements, and open challenges,” in Future Gener. Comput.Syst. vol. 92, pp. 265_275, Mar. 2019.

[3]K. Gill, S.-H. Yang, F. Yao, and X. Lu, “A ZigBee-based home automation system,” in IEEE Trans. Consum.Electron, vol. 55, no. 2, pp. 422_430, May. 2009.

[4]K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas and V. Katos, "Which IoT Protocol? Comparing Standardized Approaches over a Common M2M Application," *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-7, 2016

[5] “IEEE 802.11ax high-efficiency WLAN (HEW)," [Online]. Available: http://www.ieee802.org/11/Reports/tgax_update.htm, Accessed: Nov.20, 2019.

[6] *A. B. Prasetijo, E. D. Widianto and E. T. Hidayatullah, "Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat,"* 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 393-396, 2016.*

[7] Yu, Ye & Li, Xin & Qian, Chen.” SDLB: A Scalable and Dynamic Software Load Balancer for Fog and Mobile Edge Computing.” 55-60, 2017.

[8] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan and G. -J. Ren, "Foggy clouds and cloudy fogs," in IEEE Wireless Communications, vol. 23, no. 5, pp. 120-128, October 2016

[9] S. M. Babu, A. J. Lakshmi, and B. T. Rao, “A study on cloud-based Internet of Things: Cloudio T,” in Proc. Global Conf. Commun. Technol. (GCCT), pp. 60_65, Apr. 2015.

[10] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study ofLPWAN technologies for large-scale IoT deployment,” ICT Exp., vol. 5, no. 1, pp. 1_7, Mar. 2019.

[11] A. Ikpehai et al., "Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2225-2240, April 2019

[12] Luong, P., Nguyen, T.M. & Le, “L.B. Throughput analysis for coexisting IEEE 802.15.4 and 802.11 networks under unsaturated traffic”. In Journal of Wireless Com Network, p 127, 2016.

[13] Phuong Luong, Tri Minh Nguyen, and Long Bao Le, "Throughput analysis for coexisting IEEE 802.15.4 and 802.11 networks under unsaturated traffic", Luong et al. EURASIP Journal on Wireless Communications and Networking, 2016

[14] Sharad S. Wagh, Avinash More, Prashant R. Kharote,“Performance Evaluation of IEEE 802.15.4 Protocol Under Coexistence of Wi-Fi 802.11b”, Procedia Computer Science, Pages 745-751, Volume 57, 2015.

[15] S. Kaur and M. Kumar, “Review on load balancing in mobile ad-hoc networks,”Int. J. Advanced Research Comput. Sci. Software Eng., vol. 5, no. 4, p. 5, 2015.

[16] M. I. Gumel, N. Faruk, and A. A. Ayeni, “Routing with load balancing in wireless mesh networks,” Int. J. Current Research, vol. 3, no. 7, pp. 87–92, 2011.

[17] L. Ho and H. Gacanin, “Design principles for ultra-dense Wi-Fi deployments,” in Proc. IEEE WCNC, 2018.

[18] N. T. Thomopoulos,” Fundamentals of queuing systems statistical methods for analyzing queuing models”. Science Business Media, New York, 2012.

[19] A.-S. K. Pathan, M. M. Monowar, and S. Khan, “Simulation technologies in networking and communications - selecting the best tool for the test”. CRC Press, Taylor & Francis Group, LLC, 2015.

[20] S. Khan, A.-S. K. Pathan, and N. A. Alrajeh, “Wireless Sensor Networks - Current Status and Future Trends”. CRC Press, 2012.

[21] D. Bertsekas and R. Gallager, “Data Networks”, Second Edition. Prentice-Hall, 1992.

[22] Jondral, Friedrich.. “White Gaussian Noise – Models for Engineers. Frequenz”. 72. 10.1515/freq-2017