Computing and Informatics, Vol. 41, 2022, 330–356, doi: 10.31577/cai_2022_1_330

BOOSTED PERFORMANCE, QUICK RESPONSE, AND BETTER QOS USING IOT PLUS

Omar Mohammad

Faculty of Arts and Sciences
Lebanese International University (LIU) – BIU
Beirut-Bekaa, Lebanon
e-mail: omar.mohammad@liu.edu.lb

Zahra Yousef

Faculty of Engineering
Islamic University of Lebanon IUL
Wardaniye, Lebanon
e-mail: Zahrayousef995@gmail.com

Mirna Atieh

Faculty of Economics Science and Administration
Lebanese University UL
Beirut, Lebanon
e-mail: matieh@ul.edu.lb

Kassem Ahmad

Faculty of Engineering
Islamic University of Lebanon IUL
Wardaniye, Lebanon
e-mail: Kassem.ahmad@liu.edu.lb

Abstract. The Internet of Things (IoT), as a concept, was not officially named until 1999 where it was still used by big computer and communication companies. It is the connection between objects anywhere, anytime, using internet communication. IoT is one of the network concepts which are growing rapidly in the last few years. The connected devices reach billions which leads to a huge increase in data transfer through the network. This rapid increase of transferred data is overloading network servers which result in more processing and routing time. Fog computing and cloud computing paradigms extend the edge of the network, thus enabling a new variety of applications and services. In this research, we focus on the processing and routing time, moreover, we present a new model in the application layer of the IoT system to classify IoT applications according to their valued data. Also, we work on modeling the fog computing architecture and use the cell operator as the main fog center to store data and compare its performance with the traditional model. We present a comparative study with the traditional IoT architecture based on classifying applications and define a priority for each application. We aim to give solutions to lower data transmission time, reduce routing processes, decrease internet usages, increase response speed, deliver important and sensitive data first, improve the quality of services, enhance the overall performance of IoT systems by depending on fog network as the main layer for processing and storing data, then by giving each application a priority value to be served according to it where the application with the highest priority is served first on the network. Our method which is based on static priority shows better performance and management against the RWS and DRAG method which are based on many parameters to take a decision.

Keywords: Fog computing revolution, processing time, speed, reliability, bandwidth drop, traffic congestion, priority-based scheduling packets, IoT application priorities

1 INTRODUCTION

IoT is being used in almost all fields such as homes, industries, healthcare, and many others. Since IoT uses a lot of technologies and protocols to serve different devices connected. Every day new and different technologies arise, and here comes the power of IoT where it covers a variety of technologies to take the benefits of each technology. Fog computing is a decentralized computing architecture where data is processed and stored between the source of origin and a cloud infrastructure This results in decreasing data transmission overloading, hence, improves the performance of computing in Cloud platforms by reducing the requirement to process and store large volumes of superfluous data. But this development will introduce many new challenges.

A scalable and reliable technique should be implemented to pass the challenges with the fast development of IoT and the rapid growth of connected devices as development work which is an extension of work originally presented in [1]. Internet

is a network of interconnected networks that allows communication between people no matter how far they are. Nowadays you can communicate with your friends and family no matter where they are located with the existence of the internet [2]. IoT technology includes a large collection of networked objects, frameworks, and sensors, which takes the benefits of development in computing power, electronics miniaturization, and network interconnections to provide new capabilities that are unrealistic. [3]. This innovation ensures to be beneficial for people with inabilities, allowing improved levels of independence and quality of life at a sensible cost [4] It was broadly utilized in smart homes, smart wearable, smart city, smart environment, and smart enterprises [5]. On the opposite side, communication is the main part of IoT; Device-to-Device, Device-to-Cloud, and Device-to-Gateway [6]. Nevertheless, the expanding number of connected devices will reach 50 billion, as Cisco claims, by 2020 [7]. And a large number of the newly connected devices will be at the edge of the network and will require support for mobility, low latency, real time, and location-aware services. These are the challenges for the traditional cloud architecture which launch a new computational paradigm named fog computing. It should be clear that fog computing is not a substitution for cloud computing, these paradigms should be taken as a complement for each other to support real time, low latency applications that happen at the edge of the network. The rest of the paper is organized as follows: Section 2 contains a contribution to the state of the art, Section 3 describes the architecture of IoT. Section 4 identifies the challenges. Solutions will be listed and described in details in Section 5, Section 6 consists of a comparison between the methods, Section 7 shows a theoretical study of the work, Section 8 explains the experimental result and finally, Section 9 concludes the work.

2 STATE OF THE ART

Technology nowadays is developing in a fast way to improve human life and make it easier [5]. One of the technologies that have made a huge improvement in our daily lives is the internet. The growth of IoT depends on the devices around you that need to be interconnected, so as much as these devices increase the network of IoT increases [2]. Cloud computing, in the last years, has added a new dimension to the traditional means of computations and data storage. Researchers in [8] have suggested interference-aware scheduling for IoT sensors-based health care systems, it uses data size and sampling rate as a parameter for scheduling. It efficiently decreases the interference between sensors and eliminates the loss of data. In [9] two schemes for enhancing the IoT communication are proposed: the preconfigured access and joint spatial and code domain, they are an extension of the multi-user shared access (MUSA) scheme to the spatial domain. The authors in [10] have proposed an Efficient Task Scheduling in the Internet of Things (ETSI) algorithm. It schedules different tasks to the suitable nodes. The ETSI algorithm was said to be effective for task execution when compared to related algorithms. Al-Kashoash in [11] developes packet discarding based node clustering (PDC) for congestion control where all the

nodes presented in a specific domain of interest are clustered into many groups, and for each group, a cluster head is selected and a PDNC is implemented at each node to reduce the number of packets affected during congestion. In [12] they implemented a dynamic congestion control for a hierarchical information-centric network model for IoT sensor networks. with many research and work efforts to deploy queuing models to solve congestion problems. Kumar in [13] proposes a model that assumes that the region is divided into square grids. The number of grids is dependent on the resolution of the grid. Analysts in [14] design a technology to help reduce congestion and develop a control method to modulate data transmission rate whenever a fluctuation is presented in bandwidth and delays. Researchers have proposed a new way to improve the performance of TCP networks. The proposed technique implements TCP cubic to ensure steady-state to reduce packet drop, their experimental results showed proper enhancement in the case of throughput and interprotocol fairness for the proposed approach. In [15] a proposed improvement over TCP Westwood (TCPW), which is an adaptive sliding window algorithm used for narrow band-IoT (NB-IoT), used to enhance the status report policy in the RLC protocol stack of the radio link control layer of the NB-IoT to regulate data transmission and to achieve automatic repeat-request retransmission. The polling-TCPW intensifies throughput and decreases transmission delay of RLC with a guaranteed system stability.

3 ARCHITECTURE

There is still no common architecture for IoT systems but various architecture proposed by different researchers. The most fundamental architecture (Figure 1) has three layers: Perception, Network, and Application [16].

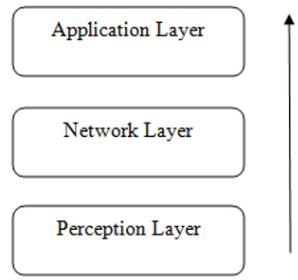

Figure 1. Three-layer basic architecture

Perception Layer. The perception layer additionally can be named as a sensing layer [17]. The primary job of this layer is to sense and accumulate physical data variation from the environment and then transform the sensed data into digital information. Sensors, cameras, and Radio Frequency Identification Devices (RFID) are examples of objects that are present in this layer [18]. Sensors are

the real item that gathers information and other object using various short-range communication protocols like WiFi, Bluetooth, and other technology [19].

Network Layers. As it is considered the main part of the system, the network layer support secure data transmission between the perception layer and the application layer [20]. Also, it provides services that enable seamless connectivity between devices and services such as addressing, routing, resource optimization, security, Quality of Service (QoS), and mobility support [21].

Application Layer. The top layer in IoT architecture is the application layer. This layer gives customized services based on user-relevant needs [22]. This layer's main responsibility is to link the major gap between users and applications. It combines the industry to achieve the high-level intelligent applications type solutions such as disaster monitoring, health monitoring, transposition, and fortune, medical and ecological environment, and handled global management compatible to all intelligent type applications [23].

4 CHALLENGES

IoT is a network of networks in which a huge number of objects, sensors, and devices are connected through a communications infrastructure to provide valued services [24]. By 2021, 94 % of workloads and compute instances will be processed by cloud data centers while 6 % only will be processed by traditional data centers [25], while mobile devices are counted for most methods for service applications [26]. Therefore, there will be a gigantic focus on the cloud leading to a heavy load on it [27], hence the performance will be affected and traffic congestion occurs. Traffic congestion could cause a lot of problems such as delays, packet loss, and timeouts [28]. By merging cloud and fog networks we can overcome the issue of overloading. According to literature research, the most important challenges are:

Processing Time. Processing delay is the time it takes the routers to process the packet header for error checking or determining next the destination while fog network for local user and cloud network for roaming user.

Routing Traffic. Routing is the process of selecting a path for traffic in a network, or between multiple networks. Distributing data on fog and cloud will help in reducing the routing table and routing process by eliminating the upper edge (Cloud) when we are on the local network.

Speed. Fog network is closer for the user than cloud network which means that using fog is faster than the cloud. Moreover, fog is the link between user and cloud and retransmitting data from fog to cloud will take more time in sending and receiving.

Bandwidth. In a network, bandwidth is the amount of data that can be transmitted in a fixed amount of time. The lack of bandwidth will interrupt the IoT services.

Performance. By achieving the first four challenges the performance will be much better. Reduced routing processes help deliver data in less time by reducing the time needed in routing processes, Also the increased speed of data transmission gives fast response time. Moreover, the more bandwidth you have, the more data you can load at once.

Quality of Service. QoS is a set of technologies that work on a network to guarantee its ability to dependably run high-priority applications and traffic under limited network capacity. QoS technologies accomplish this by providing differentiated handling and capacity allocation to specific flows in network traffic. This enables the network rules to assign the order in which packets are handled, and the amount of bandwidth afforded to that application or traffic flow.

5 SOLUTION

A scalable and reliable model should be implemented to pass the challenges with the fast development of IoT and the rapid growth of connected devices. We work with solutions in different IoT layers the network and the application layer. This section covers the present systems, the cloud-centric, fog offloading, priority scheduling, and our new method that merges the two systems in one to take their advantages.

Cloud-Centric. Cloud computing supports infrastructures, platforms, software, and storage as a service for the IoT system and users. Aneka is a cloud-centric system that offers a wide variety of services with multiple programming models for all kinds of clients. also, Aneka uses the resources and computing power of public and private cloud to give better performance and scalable storage [29]. The cooperation of public and private cloud requires an extra handling process in the background to guarantee QoS and privacy because of information sharing among private and public. Cloud-centric still relies on a middleware layer to deliver user data to cloud storage which implies that data needs more transmission time and introduces more routing delays.

Fog Offloading. Offloading is the process of distributing the load on many fog nodes to reduce IoT service delays. It can help mobile devices with overcoming resource limitations by offloading computationally intensive tasks to the remote cloud servers [30]. Naha in [31] presents a new framework that minimizes the processing delay when offloading requests, but still uses a cloud network for storage and transparent tasks to choose when to offload. The problem is when deciding to offload to another busy fog, we check for another fog which derives an increment in waiting time, moreover, offloading presents additional queuing time.

Dynamic Routing Algorithm for priority Guarantee, DRAG. DRAG is a method based on a priority scheduling schema. The main idea of this method

is to deliver as many as possible packets taking into consideration the level of priorities of these packets. So, it is focusing on delivering as many as possible packets of higher priorities concerning location, cost, remaining energy, and other parameters of these packets. This method guarantees the priority of packets using a queue management policy [32].

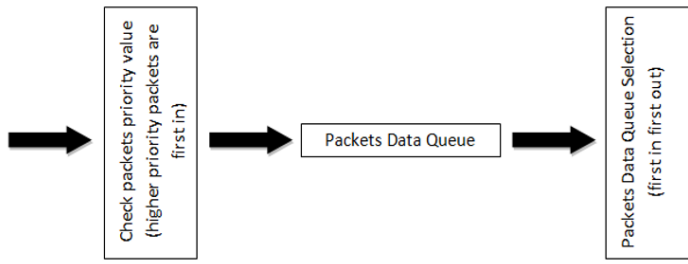

Figure 2. DRAG Scheduling (Source: Sun, Xu: Dynamic Routing Algorithm for Priority Guarantee in Low Duty-Cycled Wireless Sensor Networks [32])

Random Weighted Scheduling, RWS. RWS is a method that first selected the packets and assigned them to a priority queue then processed the packets based on a priority scheduling method. The packets are assigned to a queue based on a priority value (2 bit) find in the packet. Then a random priority number is generated and assigned for each packet in the queue and it is used to select a packet from a queue. Then NB (Number of Back-offs), BE (Back-off Exponent), and CW (Contention Window) parameters are generated after selecting the packet to allow fast channel access for data with lower values of CW and BE. CW is the number of back-off periods that the channel activity needs to be clear before starting packet transmission, BE is the number of back-off periods that a device before attempting to access the channel should wait and NB is the number of back-offs when the medium is busy [33].

While these methods and other methods use dynamic priority calculations to schedule and process data packets and assign each packet a level of priority. But dynamic priority calculations need more processing, time, energy, and cost to schedule the packets. So, we come up with a static priority calculation to assign packets a level of priority using a static table [34]. This static method will save more processing time, energy, and cost.

6 IOT PLUS

Our new method (Figure 4) depends on merging cloud networks and fog networks but with an independence of each other. The process goes through phases: classify applications and sensors, remove encryption from unconfidential data.

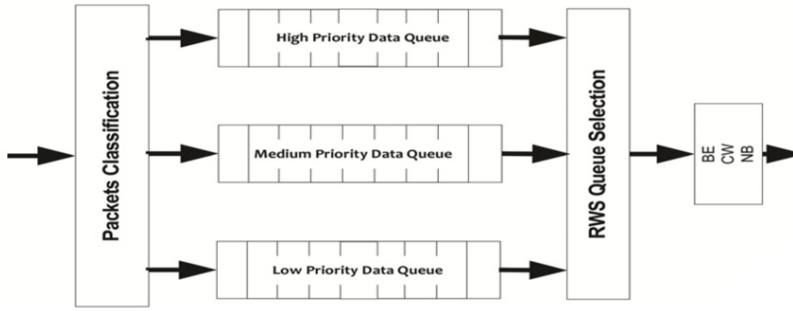

Figure 3. RWS Scheduling (Source: Sheikh, Wolhuter, Engelbrecht: A Random Priority-Based Scheduling Strategy for Wireless Sensor Networks Using Contiki [33])

Classifying Application and Sensors. In this process application and sensors, a node is arranged according to its function, priority, and level of served objects. Priority is varied from one application to another, for example an application that delivers the body temperature of a patient has higher priority on the application that sends the temperature of his home, while a similar individual uses the same application, same detected information, but in a different situation.

This step plays a major role during maintenance, monitoring, and busy hours. Although sometimes the bandwidth drops down due to various reasons like noise, rain, and others. During this drop, data must be managed and served according to its value. This method is a static priority scheduling security method which schedules, secures, and processes packets based on their priority values and the system traffic status. These priority values are selected from a static table containing the priorities for each IoT application type. IoT applications will be classified and arranged from higher to less priority based on the application type to create a static priority table. After classifying applications, the data coming from applications of higher priority will be processed first, and then the data coming from applications of lower priority. Thus, urgent and sensitive data will have a higher priority for better performance. Each packet will be processed with a different level of priority according to the traffic status of the system.

In this method, we have referred to a survey and relevant reliable statistics about IoT application priorities to create the application priorities table. The survey was made to sort IoT applications based on their priority and the statistics were graphs that show the most used applications, the most applications countries spent money on, and the applications that need a high security to analyze and sort applications to create the static table of application priorities. The survey has been distributed to 300 individuals who work as Internet service providers, network engineers, and instructors related to the field of computer and communication networks. The form

was created using Google Forms and contains an explanation of each application and how to set priority for each one. The form was distributed through email addresses for the domain experts to give relevant value in application priority to get a clear view sorting application from the most used application (higher priority) to the lowest one (lower priority). After analyzing the survey results and the graphs we come up with an IoT application priorities table (see Table 1).

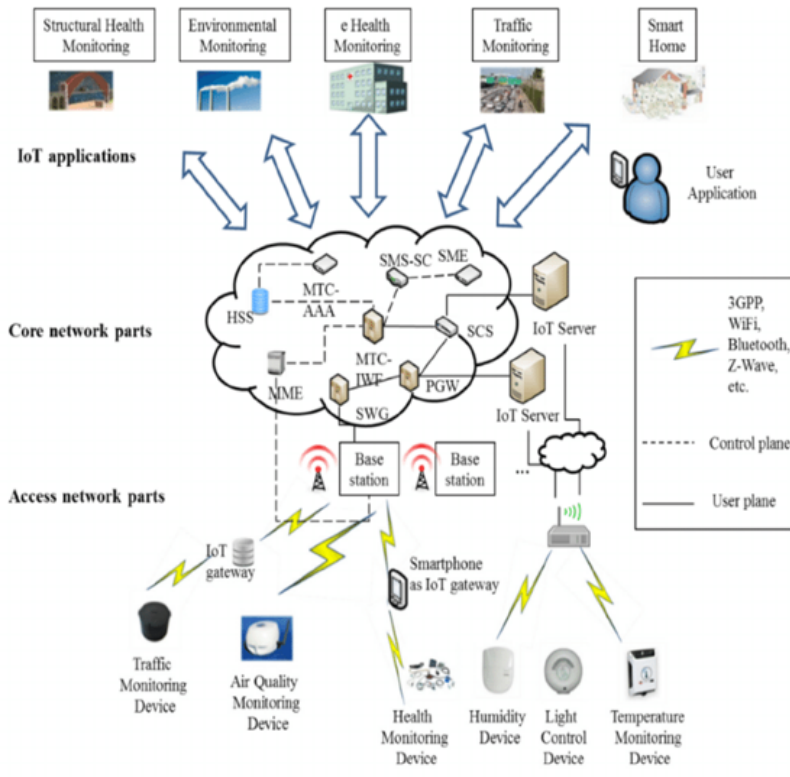

Figure 4. Hosting IoT on fog and cloud centers

After creating the IoT application priorities, we will schedule and manage packets using the DSCP (Differentiated Services Code Point) field. DSCP is a field found in the packet header which is used to classify traffic of packets and deliver packets with high priorities with the best effort. DSCP field is composed of 6 bits. To classify the packets, DSCP maintains a set of values for different levels of service which are found in the DSCP field (Table 3).

After checking the standard DSCP values in Table 3 we selected the highest DSCP decimal value which is 56 and reserved the values from 57 to 63 for the IoT applications and the results were as in Table 3. We chose the DSCP decimal value 60

Application	**Priority**
Smart City	1
Industrial Internet	2
Connected Health	3
Smart Utilities	4
Smart Home	5
Smart Supply Chain	6
Smart Farming	7
Smart Retail	8

Table 1. IoT applications priorities

as the threshold value because the 4 applications at the top are of high priorities while the last 4 applications are of lower priorities. Table 2 shows the DSCP value assigned for each IoT application where the last two IoT applications were assigned the same value.

Application	**Priority**	**Decimal Value**	**DSCP Value**
Smart City	1	57	111001
Industrial Internet	2	58	111010
Connected Health	3	59	111011
Smart Utilities	4	60	111100
Smart Home	5	61	111101
Smart Supply Chain	6	62	111110
Smart Farming	7	63	111111
Smart Retail	8	63	111111

Table 2. IoT applications DSCP values

Data encryption modulating (Confidentiality level). Confidentiality refers to protecting information from being accessed by unauthorized parties. In other words, only the people who are authorized to do so can gain access to sensitive data. The data sent through the network must be encrypted first. But encryption of data at the source then decryption on the destination will cause more processing time and delay especially during congestion time. But some applications send data that can be viewed to any user without affecting its information. Moreover, the price of encryption is more expensive than the data itself. For example, we will work on the same sensed data but from a different application. The sensed data is temperature, suppose that the same network transmits data contain the temperature of a patient using a health care application and another temperature but for a smart city. While the temperature of the city can be obtained using any mobile or using GPS (Global Positioning System) and anyone can see it without any risk on the data, while the temperature of the patient should be viewed only by the hospital or

DSCP Value	Decimal Value	Meaning	Probability Probability	Equivalent IP Precedence Value	
101 110	46	High	Forwarding (EF)	N/A	101 – Critical
000 000	0	Best Effort	N/A	000 – Routine	
001 010	10		AF11	Low	001 – Priority
001 100	12	AF12	Medium	001 – Priority	
001 110	14	AF13	High	001 – Priority	
010 010	18	AF21	Low	010 – Immediate	
010 100	20	AF22	Medium	010 – Immediate	
010 110	22	AF23	High	010 – Immediate	
011 010	26	AF31	Low	011 – Flash	
011 100	28	AF32	Medium	011 – Flash	
011 110	30	AF33	High	011 – Flash	
100 010	34	AF41	Low	100 – Flash Override	
100 100	36	AF42	Medium	100 – Flash Override	
100 110	38	AF43	High	100 – Flash Override	
001 000	8	CS1		1	
010 000	167	CS2		2	
011 000	24	CS3		3	
100 000	32	CS4		4	
101 000	40	CS5		5	
110 000	48	CS6		6	
111 000	56	CS7		7	
000 000	0	Default			
101 110	46	EF			

Table 3. DSCP standard values (Source: Cisco Nexus 1000V Quality of Service Configuration Guide, Release 4.0(4)SV1(3) [35])

patient-doctor. Then we can reduce the processing time by sending unimportant data without encryption while the sensitive data must be encrypted. This step is activated only when we have traffic congestion, during normal operation, all data is encrypted even the data that comes from low priority applications.

7 COMPARISON

The three models provide many solutions at different levels to support IoT services with better performance and quality. The comparison is made to compare between the three solutions to determine which one is better to apply and the most enhancement in case of the data processing, response speed, available bandwidth, best performance, and QoS. The description below is treated at random while the results show the difference when dealing with data and the advantages of the model against

DSCP Value	Decimal Value	Meaning
000000 0	Best effort	
000000	0	Default
001010	10	AF11
001100	12	AF12
001110	14	AF13
010000	16	CS2
010010	18	AF21
010100	20	AF22
010110	22	AF23
011000	24	CS3
011010	26	AF31
011100	28	AF32
011110	30	AF33
100000	32	CS4
100010	34	AF41
100110	36	AF42
101000	40	CS5
101110	46	High priority expedited forwarding
101110	46	EF
110000	48	CS6
111000	56	CS7
111001	57	High priority application
111010	58	High priority application
111011	59	High priority application
111100	60	High priority application
111101	61	Low priority application
111110	62	Low priority application
111111	63	Low priority application
111111	63	Low priority application

Table 4. shows the final DSCP table where we added our constant DSCP values to the standard DSCP values

others.

Data Processing. The size of data processed in the cloud center is bigger than the data in the fog center which means that the cloud needs more processing time. Also, in fog computing, the load is distributed which helps in reducing processing time [36]. But there is a condition for deciding which is better in the case of fog-centric (fog offloading or direct host in cell operator with the priority of application). The first is better when the fog centers are available by distributing the load on several units. But if the centers are overloaded, offloading becomes slowest due to additional queuing delay and the need for more routing updates.

Response Speed. Response time is better in our method and fog offloading than cloud-centric due to the presence of a user at a close distance to the fog edges (Figure 5).

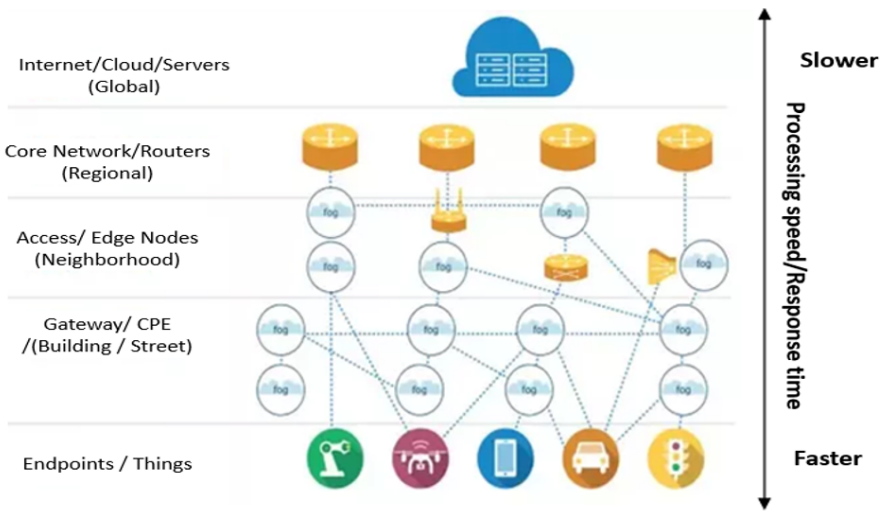

Figure 5. Response speed between users and edges level

Bandwidth. In the cloud-centric system, the bandwidth is shared across the global users. While in fog offloading, users can benefit from sharing multiple bandwidths if some centers are abused. In both cases, there is still a need for more bandwidth during busy hours, and when heavy load data is processed. But in our method, there is no starvation for bandwidth, since the user uses the WAN infrastructure to share data. The bandwidth, in this case, is the whole capacity of the transmission channel. The lack of bandwidth or any error on the internet will cause an interruption in the system and a drop in QoS or a stop of the service for cloud computing and fog offloading. But it can run normally in our system because it is independent of real internet services (Figure 6).

Performance. Cloud-centric model provides scalable storage with a variety of platform but still depend totally on the middleware layer to serve users. Fog offloading improves the first one by offloading the data process and minimize processing time, but still depends on the cloud and introduces more queue delay. Our method takes the benefit of fog in a fast response and a short distance to serve users and the scalability of the cloud network to ensure reliable services. Furthermore, IoT Plus eliminates the additional tasks used in the first two methods and gives static parameters to take decisions. In all challenges, fog computing is better than cloud computing where our method has an extra point over fog offloading by serving application according to its valued data and by operating

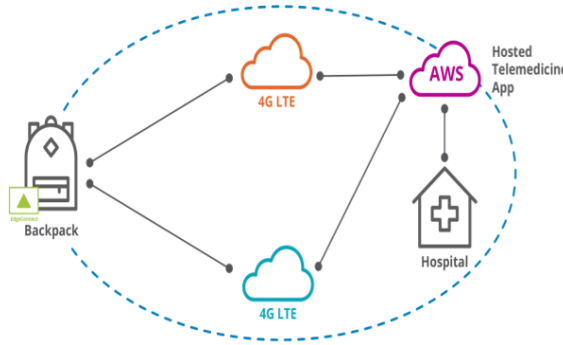

Figure 6. WAN hosted IoT application

without real internet services which make it better during bandwidth drop and busy hours.

QoS. This step is composed of two main procedures: The first procedure is scheduling IoT packets based on the static priority table. Packets will be scheduled such that packets with higher priorities would be selected and processed before packets with lower priorities. For example, if we have packets of smart city type and packets of smart farming type then the packets of smart city type would be processed first.

8 THEORETICAL STUDY ON THE APPLICATION LAYER

In this section, we will compare DRAG and RWS with our IoT Plus model to calculate and compare delays and the number of packets processed to find the best performance during busy hours. While the packet is being sent from the source to the destination, four delays occur which are: transmission delay, propagation delay, queuing delay, and processing delay [37]. In this example, we will suppose that each packet step will take 1 ms processing time and each additional queue delay will add 1 ms.

Scenario of DRAG. In the DRAG method, the priority packet values are checked to insert the packet in the appropriate place in the queue. Then packets are selected from the queue as first in first out.

We have 4 steps where each step adds 1 ms to the processing delay. An additional 1 ms is added after each sends process data, a background process is needed to reselect from the queue the packets to be sent again. The processing of each packet took 5 ms without adding the queue delay during the insertion to the queue.

Scenario of RWS. In the RWS method, the priority packet values are checked to insert each packet in the appropriate queue. For example, packets with high

Figure 7. DRAG scenario

priority values will be inserted into the high priority queue, medium priority values will be inserted into the medium priority queue and small priority values will be inserted into the small priority queue. Then for each queue, a random priority number is generated to be used in the selection of the packets from the queue. After selection, three dynamic parameters which are CW, BE, and NB are generated which will be used to allow the access channel for the data with high priority faster than data with low priority.

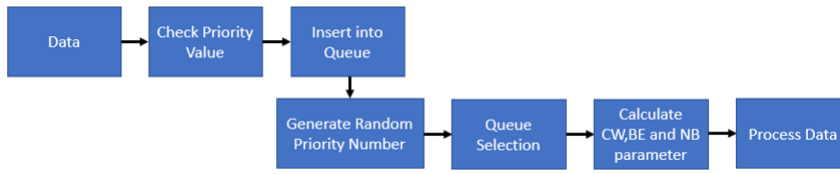

Figure 8. RWS scenario

In RWS, we have 6 steps where each step adds 1 ms to the processing delay. The processing of each packet took 6 ms without adding the queue delay during the insertion to the queue.

Scenario of IoT Plus. In scenario 1 that considers processing high-priority packets, we have 3 steps where each step adds 1 ms to the processing delay. The processing of each packet took 3 ms. Moreover, there is no additional queue delay needed in the process.

In scenario 2 that considers processing low priority packets, we have 2 steps where each step adds 1 ms to the processing delay. The processing of each packet took 2 ms. Moreover, there is no additional queue delay needed in the process.

To compare our model with these methods, we will calculate the overall delay for 300 packets. Table 7 compares the difference in the delay for a packet between the DRAG method, RWS method, and our method IoT Plus. From this table, we can observe that our method is faster by a rate of 1.6 (1.5/0.9) than DRAG and by a rate of 2 (1.8/0.9) than RWS.

Another calculation is the number of packets that will be transferred through a busy network during 1 hour. 1 hour $= (3\,600 * 1\,000)$ ms. Table 8 compares the difference in the number of packets processed in one hour of busy traffic between the

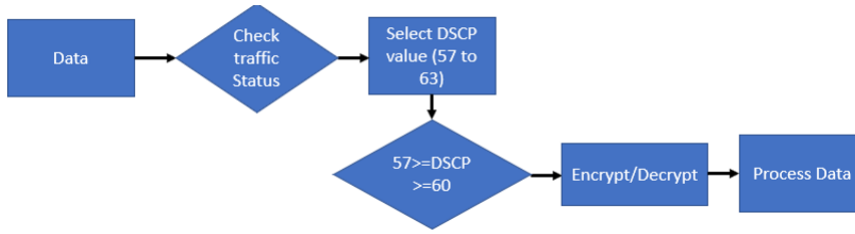

Figure 9. IoT Plus scenario 1

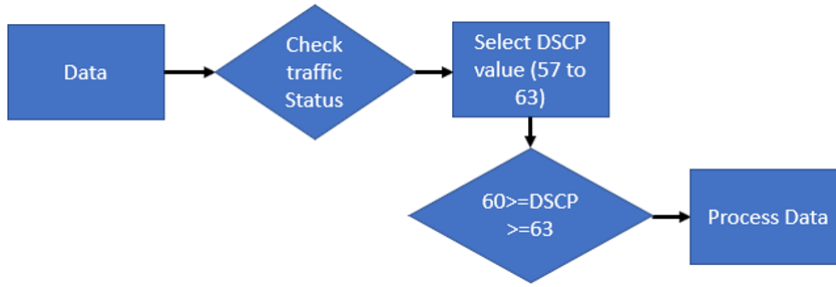

Figure 10. IoT Plus scenario 2

DRAG method, RWS method, and our method IoT Plus. From this table, we can observe that our method delivers more packets than other methods. Our method delivers 480 000 packets (1 200 000 − 720 000) more than DRAG and 600 000 packets (1 200 000 − 600 000) more than RWS.

Figure 11 shows the number of packets processed at different times in seconds using the DRAG method, RWS method, and IoT Plus method. If we add the queuing delay to the DRAG and RWS method, IoT-S+ will score better results since there is no queuing delay in our method. Using these results, we can observe that our method IoT Plus can deliver about double the size of packets more than other methods. Table 9 compares the existing methods and our new method. From this table, we can observe that all methods provide enhancements, but our method can score better performance in the case of traffic congestion.

Method	**Delay for 300 Packets**
DRAG	$300 * 5\,\text{ms} = 1.5\,\text{s}$
RWS	$300 * 6\,\text{ms} = 1.8\,\text{s}$
IoT Plus (considering case of processing packet with high priority value)	$300 * 3\,\text{ms} = 0.9\,\text{s}$

Table 5. Comparing delay for 300 packets using DRAG, RWS, and IoT Plus

Method	Number of packets through one hour
DRAG	$3\,600 * 1\,000\,\text{ms}/5\,\text{ms} = 720\,000$ packets
RWS	$3\,600 * 1\,000\,\text{ms}/6\,\text{ms} = 600\,000$ packets
IoTPlus (considering case of processing packet with high priority value)	$3\,600 * 1\,000\,\text{ms}/3\,\text{ms} = 1\,200\,000$ packets

Table 6. Comparing packet numbers using DRAG, RWS, and IoT Plus

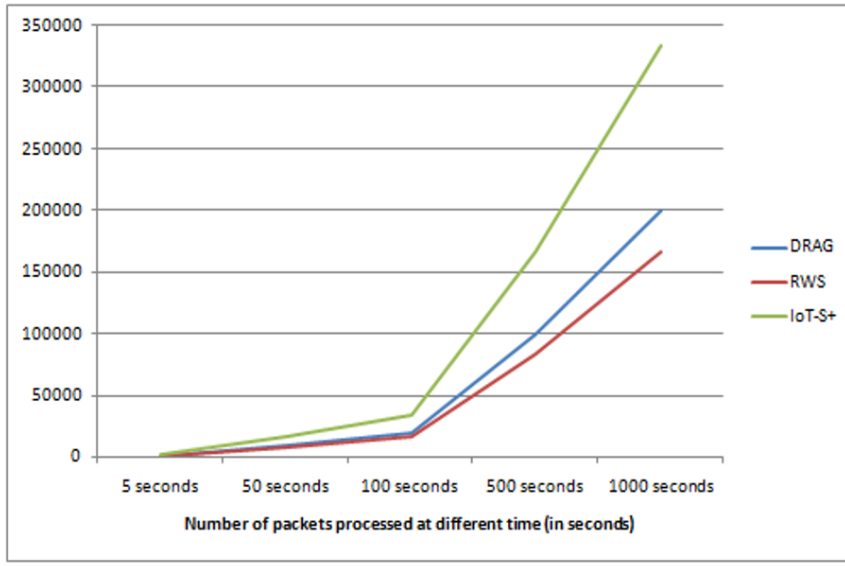

Figure 11. Number of packets processed at a different time (seconds)

9 EXPERIMENTAL RESULTS

This part describes the experiments done and shows the importance of priority in some cases to enhance performance. The experiments are done on multiple devices using different applications and real data stimulation. When comparing the results, we found that some application interrupts the services of IoT and affects users. Using the priority, we can help solve some problems. To test our method, we have used hardware and software programs for multiple experiments. The hardware used in this experiment is a standard laptop with middle specifications that are available for most users, Router Board RB931 – 2nD and Router Board RBSXTsq5HPnD. The software programs used in this experiment are Microsoft Windows 10 Pro, Router OS 6.42.1, and Winbox 3.18.

Experiment. The first experiment shows the difference in the average number of processing data with and without security, the experiment aims to show real simulation on the core router and how the range of data changes us-

Method	**Decission Parameter**	**Parameter Type**	**Enhacement**	**Disadvanteges**
DRAG	dynamic	packet priority, back-off duration	QoS, increasing in delivery ratio, decreasing per-hop delay	background process to calculate parameters (back-off duration), additional queue delay and increasing processing time to sort queues
RWS	dynamic	packet priority, CW and BE	QoS, reduction in packet loss	creating new queue delay and increassing processing time to sort queues
IoT-S+	static	application priority	QoS, reduction in processing time, increasing in response time and eliminating queue delays	add new rows to DSCP field

Table 7. Comparison between DRAG, RWS, and IoT Plus

ing the same hardware. Figure 12 shows that the average processed data is 36.7 Mbps/18.0 Mbps transmission while we overload the processor and shows that with encryption the routers can process less data where we observe the average is 29.6 Mbps/14.3 Mbps in transmission reception. The second experiment shows the difference in response time during processing overload. We used the ping tool in the Router Board to evaluate the response with and without encryption.

Figure 13 shows the minimum, maximum, and average time of packets response. Moreover, it shows the loss percentage of packets in the first stage we overload the processors and start pinging without encryption. As we see that we have 1 % of data loss, a minimum of 2 ms, a maximum of 84 ms, and an average of 8 ms response time.

In Figure 14, we overloaded the processors and start pinging with encryption. It shows that we have 3 % of data loss, a minimum of 3 ms, a maximum of 96 ms, and an average of 17 ms response time. Table 10 summarizes the comparison of parameters for the data processed with and without security.

Security	**Average Processed Data in Mb/s (t/r)**	**Response Time in (ms)**	**Data Loss Percentage**
None	36.7/18.0	8	1 %
Encryption	29.5/14.3	12	3 %

Table 8. Comparison of parameters for data processed with/without security

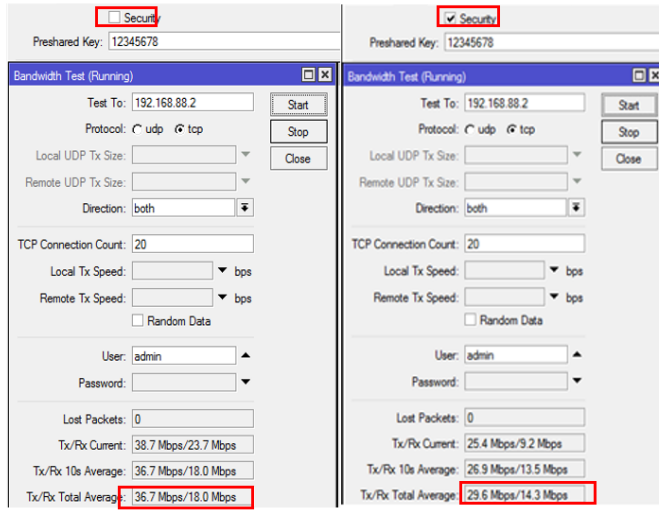

Figure 12. Maximum processed data without security

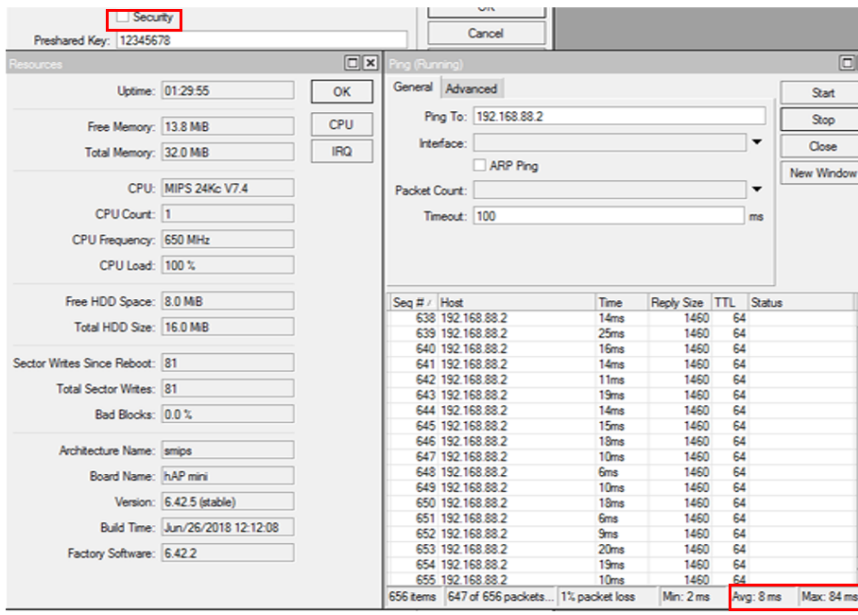

Figure 13. Response time without security

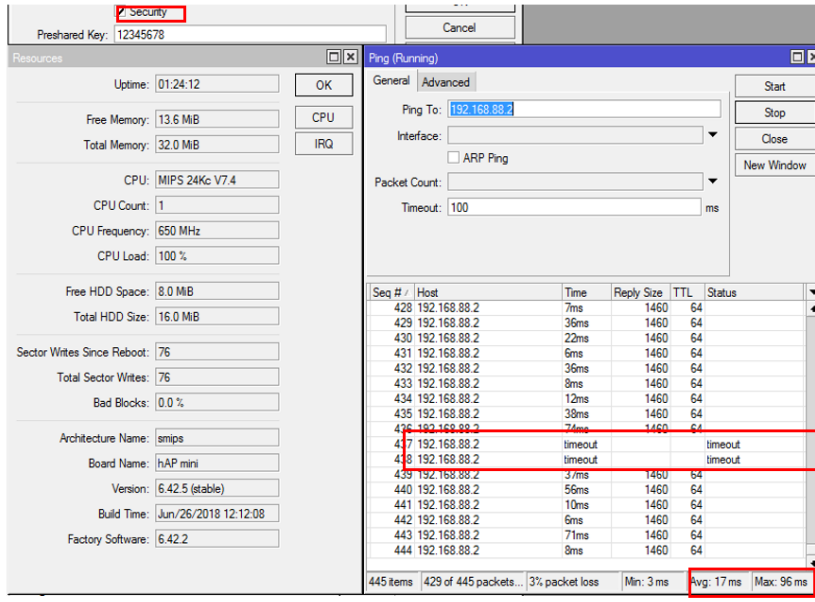

Figure 14. Response time with encryption security

Table 9 shows many tests made on a different number of packets to test packet loss with and without encryption. As we can observe, response time without encryption is better than the response time with encryption.

Packet Loss in %	**With Encryption**	**Without Encryption**
Test with 300 packets	3 %	1 %
Test with 1 060 packets	2 %	1.2 %
Test with 1 501 packets	2.6 %	1.2 %
Test 7 with 2 280 packets	4 %	1.5 %
Test with 2 900 packets	2.5 %	2 %
Test with 3 250 packets	1.7 %	1.3 %
Test with 4 060 packets	3 %	0.8 %
Test with 4 880 packets	2 %	0.8 %
Test with 5 022 packets	1.4 %	1.8 %
Test with 5 200 packets	3.2 %	0.6 %
Test with 5 336 packets	4.1 %	1.2 %

Table 9. Multiple tests for packet loss of different packets processed with and without security

Processing Time in ms	With Encryption	Without Encryption
Test with 300 packets	12 ms	8 ms
Test with 1 060 packets	12 ms	6 ms
Test with 1 501 packets	11 ms	8 ms
Test with 2 802 packets	9 ms	6 ms
Test with 2 900 packets	12 ms	11 ms
Test with 3 250 packets	14 ms	10 ms
Test with 4 060 packets	18 ms	7 ms
Test with 4 880 packets	12 ms	10 ms
Test with 5 022 packets	12 ms	6 ms
Test with 5 200 packets	21 ms	16 ms
Test with 5 336 packets	24 ms	15 ms

Table 10. Multiple tests for processing time of different packets processed with and without security

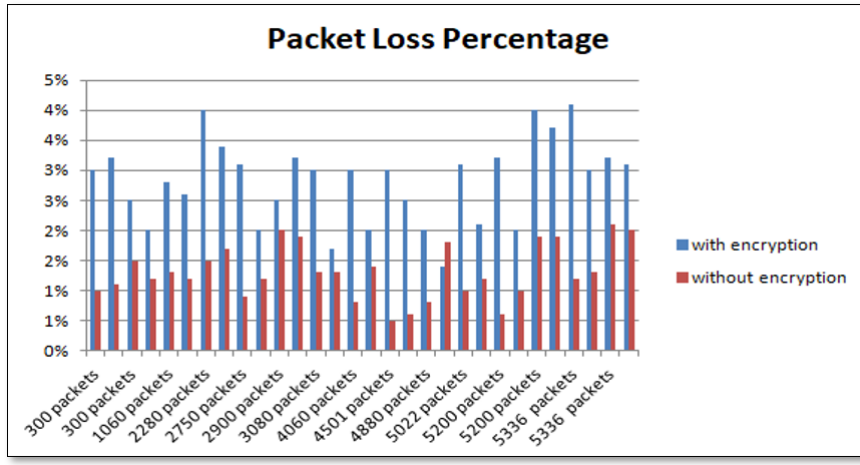

Figure 15. Comparing packet loss with and without security

Figure 15 compares the packet loss for a different number of packets based on the experiments we did before. As we can see, the packet loss without encryption is lower than the packet loss with encryption.

Figure 16 compares the response time in milliseconds for the different number of packets based on the experiments we did before. As we can see, response time without encryption is better than the response time with encryption.

After these testings, we recognize the following: in case of normal traffic, the whole system can run in full security because the system can handle it, while as the result shows when we face traffic congestion the performance is better without encrypting low priority data in case of average processed data, delays, response time,

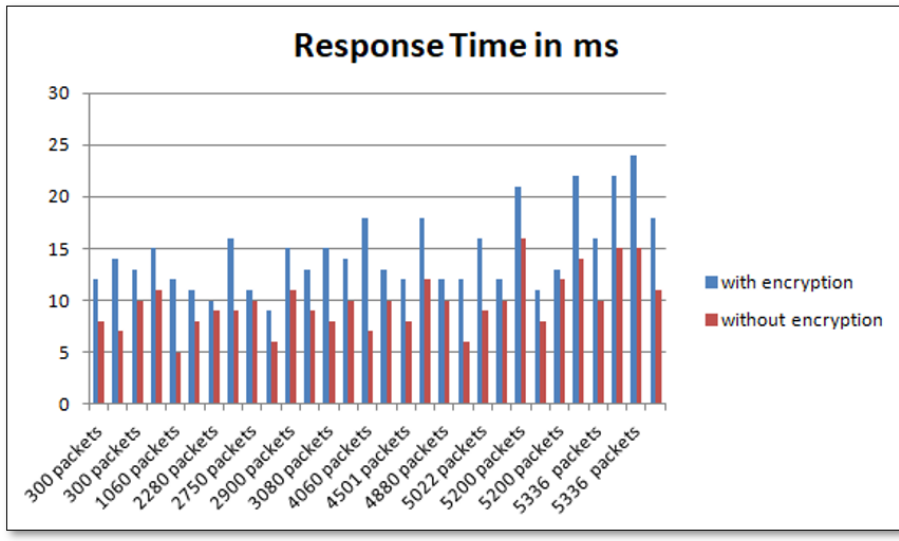

Figure 16. Comparing response time in milliseconds with and without security

and packet loss. The result shows a heavy load during congestion and an increasing number in packet loss, slower response time, and a decrease in the size of processed data due to resource overloading. We can see that the number of packets processed during congestion is less than the normal use. Moreover, the elimination of encryption helps in processing more data and decreases the response time, then it helps to overcome these problems when we have an overload on the network the system switch to IoT Plus to save the available resources to the application with the most priority, then it starts to eliminate encryption from low priority application to save fog resources for all processes during the congestion.

10 CONCLUSION

We have outlined the key characteristics of fog computing, a platform to deliver a rich portfolio of new services and applications at the edge of the network. The three methods enhanced the performance of the IoT system with different proportions, while the strong point of our method is the static parameter that is used to classify packets in the network that is based on detecting the priority value in contrary the DRAG, and RWS do the same work to classify packets but with dynamic parameter where every classifying process is based on the previous one and every time, a hidden calculation must be done to make the right decision which takes more time and resources to do the job. The issue resulted from RWS and DRAG is the unwanted delay due to extra processes in the background which is absent in the case of IoT Plus. At the end of this research, our method has gained the expected

results. We have achieved our goal of decreasing the processing delay in case of traffic congestion where packets with low priority would be processed without security while high priority packets would be processed first with security to perform better performance and keep the sensitive and important data secure. Thus, the system will stay secure.

REFERENCES

[1] Ahmad, K.—Mohammad, O.—Atieh, M.—Ramadan, H.: IoT: Architecture, Challenges, and Solutions Using Fog Network and Application Classification. 2018 International Arab Conference on Information Technology (ACIT), 2018, pp. 1–7, doi: 10.1109/ACIT.2018.8672696.

[2] Vyas, D. A.—Bhatt, D. N.—Jha, D.: IoT: Trends, Challenges and Future Scope. International Journal of Computer Science and Communication (IJCSC), Vol. 7, 2015, No. 1, pp. 186–197.

[3] Koushik, A. N.—Rashmi, B. S.: 4th Generation SCADA Implementation for Automation. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, 2016, No. 3, pp. 629–631.

[4] Rose, K.—Eldridge, S.—Chapin, L.: The Internet of Things: An Overview – Understanding the Issues and Challenges of a More Connected World. The Internet Society (ISOC), Vol. 80, 2015, pp. 1–50.

[5] Pundir, Y.—Sharma, N.—Singh, Y.: Internet of Things (IoT): Challenges and Future Directions. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, 2016, No. 3, pp. 960–964.

[6] Tschofenig, H.—Arkko, J.—Thaler, D.—McPherson, D.: Architectural Considerations in Smart Object Networking. RFC 7452, 2015, doi: 10.17487/RFC7452.

[7] Evans, D.: The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. White Paper, Cisco, 2011.

[8] Dhar, S. K.—Bhunia, S. S.—Mukherjee, N.: Interference Aware Scheduling of Sensors in IoT Enabled Health-Care Monitoring System. 2014 Fourth International Conference of Emerging Applications of Information Technology, IEEE, 2014, pp. 152–157, doi: 10.1109/EAIT.2014.50.

[9] Jiang, Z.—Han, B.—Chen, P.—Yang, F.—Bi, Q.: On Novel Access and Scheduling Schemes for IoT Communications. Mobile Information Systems, Vol. 2016, 2016, Art. No. 3973287, doi: 10.1155/2016/3973287.

[10] Baranidharan, B.—Saravanan, K.: ETSI: Efficient Task Scheduling in Internet of Things. International Journal of Pure and Applied Mathematics, Vol. 117, 2017, No. 22, pp. 229–233.

[11] Al-Kashoash, H. A.—Amer, H. M.—Mihaylova, L.—Kemp, A. H.: Optimization-Based Hybrid Congestion Alleviation for 6LoWPAN Networks. IEEE Internet of Things Journal, Vol. 4, 2017, No. 6, pp. 2070–2081, doi: 10.1109/JIOT.2017.2754918.

[12] Sukjaimuk, R.—Nguyen, Q. N.—Sato, T.: Dynamic Congestion Control in Information-Centric Networking Utilizing Sensors for the IoT. 2018 IEEE Region Ten Symposium (Tensymp), IEEE, 2018, pp. 63–68, doi: 10.1109/TENCONSpring.2018.8691983.

[13] Kumar, M.—Sabale, K.—Mini, S.—Panigrahi, T.: Priority Based Deployment of IoT Devices. 2018 International Conference on Information Networking (ICOIN), IEEE, 2018, pp. 760–764, doi: 10.1109/ICOIN.2018.8343220.

[14] Mishra, N.—Verma, L. P.—Srivastava, P. K.—Gupta, A.: An Analysis of IoT Congestion Control Policies. Procedia Computer Science, Vol. 132, 2018, pp. 444–450, doi: 10.1016/j.procs.2018.05.158.

[15] Zhou, C.—Zhao, J.—Liu, H.: Adaptive Status Report with Congestion Control in NB-IoT. 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), IEEE, 2019, pp. 1–5, doi: 10.1109/IOTSMS48152.2019.8939217.

[16] Abdmeziem, M. R.—Tandjaoui, D.—Romdhani, I.: Architecting the Internet of Things: State of the Art. In: Koubaa, A., Shakshuki, E. (Eds.): Robots and Sensor Clouds. Springer, Cham, Studies in Systems, Decision and Control, Vol. 36, 2016, pp. 55–75, doi: 10.1007/978-3-319-22168-7_3.

[17] Sikder, A. K.—Petracca, G.—Aksu, H.—Jaeger, T.—Uluagac, A. S.: A Survey on Sensor-Based Threats to Internet-of-Things (IoT) Devices and Applications. IEEE Communications Surveys and Tutorials, Vol. 23, 2021, No. 2, pp. 1125–1159, doi: 10.1109/COMST.2021.3064507.

[18] Bari, N.—Mani, G.—Berkovich, S.: Internet of Things as a Methodological Concept. 2013 Fourth International Conference on Computing for Geospatial Research and Application, IEEE, 2013, pp. 48–55, doi: 10.1109/COMGEO.2013.8.

[19] Lane, N. D.—Miluzzo, E.—Lu, H.—Peebles, D.—Choudhury, T.—Campbell, A. T.: A Survey of Mobile Phone Sensing. IEEE Communications Magazine, Vol. 48, 2010, No. 9, pp. 140–150, doi: 10.1109/MCOM.2010.5560598.

[20] Yu, Y.—Wang, J.—Zhou, G.: The Exploration in the Education of Professionals in Applied Internet of Things Engineering. 2010 4th International Conference on Distance Learning and Education, IEEE, 2010, pp. 74–77, doi: 10.1109/ICDLE.2010.5606038.

[21] Gavrilović, Z.—Maksimović, M.—Popović, B.: The Impact of the Internet of Things on the Digital Economy. Novi Ekonomist: Journal of Economic Theory and Practice, Vol. 10, 2016, No. 20, pp. 97–102 (in Serbian).

[22] Sethi, P.—Sarangi, S. R.: Internet of Things: Architectures, Protocols, and Applications. Journal of Electrical and Computer Engineering, Vol. 2017, 2017, Art. No. 9324035, doi: 10.1155/2017/9324035.

[23] Suo, H.—Wan, J.—Zou, C.—Liu, J.: Security in the Internet of Things: A Review. 2012 International Conference on Computer Science and Electronics Engineering, Vol. 3, 2012, pp. 648–651, doi: 10.1109/ICCSEE.2012.373.

[24] Silva, B. N.—Khan, M.—Han, K.: Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges. IETE Technical Review, Vol. 35, 2018, No. 2, pp. 205–220, doi: 10.1080/02564602.2016.1276416.

[25] Bello, O.—Zeadally, S.—Badra, M.: Network Layer Inter-Operation of Device-to-Device Communication Technologies in Internet of Things (IoT). Ad Hoc Networks, Vol. 57, 2017, pp. 52–62, doi: 10.1016/j.adhoc.2016.06.010.

[26] Shi, Y. R.—Hou, T.: Internet of Things Key Technologies and Architectures Research in Information Processing. Applied Mechanics and Materials, Vol. 347–350, 2013, pp. 2511–2515, doi: 10.4028/www.scientific.net/AMM.347-350.2511.

[27] Ahmad, K.—Ramadan, H.—El-Hajj, M.—Hamieh, J.: Performance Analysis and Comparison of Detecting DoS Attacks in IoT Using Machine Learning, Deep Learning and Data Mining: A Survey. Proceedings of 13th IEEE International Conference for Internet Technology and Secured Transactions (ICITST 2018), 2018, pp. 84–92.

[28] Wilson, M.: Network Congestion – 5 Causes and How to Alleviate Issues with Your Network Being Congested! 2021, https://www.pcwdld.com/network-congestion.

[29] Gubbi, J.—Buyya, R.—Marusic, S.—Palaniswami, M.: Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems, Vol. 29, 2013, No. 7, pp. 1645–1660, doi: 10.1016/j.future.2013.01.010.

[30] Liu, J.—Wang, S.—Zhou, A.—Kumar, S. A.—Yang, F.—Buyya, R.: Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability. IEEE Transactions on Cloud Computing, Vol. 6, 2018, No. 4, pp. 1191–1202, doi: 10.1109/TCC.2016.2567392.

[31] Naha, R. K.—Garg, S.—Georgakopoulos, D.—Jayaraman, P. P.—Gao, L.—Xiang, Y.—Ranjan, R.: Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. IEEE Access, Vol. 6, 2018, pp. 47980–48009, doi: 10.1109/ACCESS.2018.2866491.

[32] Sun, G.—Xu, B.: Dynamic Routing Algorithm for Priority Guarantee in Low Duty-Cycled Wireless Sensor Networks. In: Pandurangan, G., Anil Kumar, V. S., Ming, G., Liu, Y., Li, Y. (Eds.): Wireless Algorithms, Systems, and Applications (WASA 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6221, 2010, pp. 146–156, doi: 10.1007/978-3-642-14654-1_19.

[33] Sheikh, S. M.—Wolhuter, R.—Engelbrecht, H. A.: A Random Priority Based Scheduling Strategy for Wireless Sensor Networks Using Contiki. Proceedings of the 13th International Joint Conference on E-Business and Telecommunications (ICETE 2016), 2016, pp. 121–128, doi: 10.5220/0005949301210128.

[34] Ahmad, K.—Mohammad, O.—Atieh, M.—Ramadan, H.: Enhanced Performance and Faster Response Using New IoT Litetechnique. The International Arab Journal of Information Technology, Vol. 16, 2019, No. 3A, pp. 548–556.

[35] Cisco: Cisco Nexus 1000V Quality of Service Configuration Guide, Release 4.0(4)sv1(3). https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/4_0_4_s_v_1_3/qos/configuration/guide/n1000v_qos/n1000v_qos_6dscpval.html [accessed 1.2.2020].

[36] Palo Alto TechDocs: Enforce QoS Based on DSCP Classification. https://docs.paloaltonetworks.com/pan-os/7-1/pan-os-admin/quality-of-service/enforce-qos-based-on-dscp-classification [accessed 2.2.2020].

[37] GeeksforGeeks: Packet Switching and Delays in Computer Network. `https://www.geeksforgeeks.org/computer-network-packet-switching-delays/` [accessed 15.11.2019].

Omar Mohammad obtained his M.Sc. degree from the Lebanese International University in 2018, and his B.Sc. degree from LIU in 2016. His field of interest lies in the area of the Internet of Things, robotic control, wireless communication, wireless body sensor, network security, and application development. Currently, he works as a researcher and instructor in the School of Art and Science at the Lebanese International University. He works as a coordinator and researcher in the development department of RoboGeex Academy Lebanon. He was honored with many awards from different universities and institutes like IEEE international conferences (best paper award) and was ranked number 1 in Lebanese Telecommunication official exams 2013. He has published several papers in international conferences and journals.

Zahra Yousef obtained her Master's degree in telecommunication networks from the Islamic University (Beirut). She obtained her Bachelor in computer and communication engineering from Al Salam University College. She is currently Assistant Professor and researcher at Al-Salam University College. She worked as an engineer at the Iraqi Ministry of Electricity – Autonomous Control Department. She has good knowledge of programming languages like CSS, HTML, and JavaScript and skills in digital graphic modeling. She has published several papers in different international journals and conferences.

Mirna Atieh obtained her Ph.D. in informatics and artificial intelligence in February 2008 from the Institut National des Sciences Appliquées INSA de Rennes (France). She is currently Assistant Professor and Researcher at the Lebanese University in Lebanon – Faculty of Economic Sciences and Business Administration – Department of Business Computer. Her main research interests are in the areas of Artificial Intelligence (AI), Networking and Telecommunication, and the Internet of Things (IoT). She has multiple scientific collaborations with some universities in France and Canada. She has published several papers in international conferences and journals.

Kassem Ahmad obtained his Ph.D. degree in telecom and networks security in 2013 from the Ecole Polytechnique of the University of Nantes (France). Also, he has received his Master's degree in telecommunication networks from the University of Quebec, INRS-EMT (Canada). He obtained his Bachelor in computer and communication engineering from IUL University. He worked on different European projects in developing solutions and protocols for the next generation of mobile networks and IoT (Internet of Things). He currently works as Assistant Professor and researcher in the School of Arts and Sciences at Lebanese International University. He worked at CNJF (Committee National des Jeux de la Francophonie) as an IT manager. He was honored with several awards from different institutes like the Order of Engineers and Architects of Beirut and IEEE International Conferences. He has published several papers in different international journals and conferences.