A Study of QoS in an Integrated Architecture of WLAN and Hetnet Based LTE-A

Slvar Abdulazeez Arif[1] & Tara Ali Yahiya[2]

[1]Physics Education Department, Education Faculty, Tishk International University, Erbil, Iraq
[2]University of Paris-Saclay, Paris, France
Correspondence: Slvar Abdulazeez Arif, Tishk International University, Erbil, Iraq.
Email: slvar.abdulazeez@tiu.edu.iq

Doi: 10.23918/eajse.v7i1p254

Abstract: In this paper, an integration architecture based on loose coupling concept between WiFi and LTE heterogeneous networks (HetNet) was proposed. The ultimate objective is to investigate the feasibility and the practicality of the design through investigating the performance measures for different applications in the network. As well, a functionality of the load balancing was deployed in the architecture in order to prove that the architecture is flexible and can be opened to any further functionality. The architecture was built and simulated using riverbed network simulator. It was proved that the architecture is working appropriately, and the connectivity of the different technologies were demonstrated by applications communication with each other's in the architecture. Along with, different existing load balancer algorithms were tested on the architecture and numerical results were obtained demonstrating that algorithm is better.

Keywords: HetNet, Loose Coupling, LTE-A, WLAN, QoS, Load Balancing

1. Introduction

Currently, the demand for communicating anytime and anywhere is becoming one of the higher priorities of the mobile users. It is expected that the future networks will consist of a group of heterogeneous systems that the users will be able to access the networks simultaneously through having the choice of accessing the network through different available technologies (Autoridade Nacional de Comunicações, 2021).

The HetNet in the wireless is becoming one of the important trends in the area. To give the ability to a user for connecting anytime and anywhere, there is no wireless radio access technology can do this alone, or it will be the case of having a various radio access technology, in each geographical area creating a wireless infrastructure (Choque et al., 2014). Loose coupling is one of the main solutions for solving this problem. The loose coupling concept works in a way which the Long-term Evaluation (LTE) and WiFi networks are independent physically. When both of the networks will connect to the internet, the user devices will be free of choosing which one of the technologies to connect to. This increases the chance of connecting anywhere and anytime.

Received: April 19, 2021
Accepted: June 27, 2021
Arif, S.A., & Yahiya, T.A. (2021). A Study of QoS in an Integrated Architecture of WLAN and Hetnet Based LTE-A. *Eurasian Journal of Science & Engineering*, 7(1), 254-273.

As the user devices want to connect to WiFi, they will first scan the network for available access points in the WiFi, but before starting to send or receive data, they should get authentication by the selected access points (“supportcenter”, 2021). And the other technology candidate is LTE which was defined in 3GPP with the releases of 8 to 10. However, the LTE does not meet all the requirements of the 4G network. But it is advanced version which is LTE-A does, so as the LTE-A able to meet the 4G system’s requirements by forwarding the idea of small cells (Macrocell, Picocell. Femtocell). But the Macrocell network architecture limits the LTE-A’s performance, so Macrocell’s complementary is Femto and Picocells are supported by LTE. These new introduced cells are small cells with cost efficient equipments and typical for positioning them in Hotspots. If the device is placed indoors, the cell is called Femtocell. When the placement is in outdoor then the cell is called Picocell (Bjerke, 2011). In the paper, to prove that the loose coupling works well and to assure the guarantee of the network’s Quality of service (QoS) the load balancing as a network functionality is introduced. By load balancing, a jungle of various types of traffics, and their loads can be scheduled dynamically through the cells to avoid blocking of the traffics in high loaded areas, while in the low loaded areas the resources are underutilized or wasted. With the recent evolutions, the load balancing is extended to work in heterogeneous networks with different network technologies (Alisa, 2013), which this paper is focused on.

2. Background of Telecommunication Technologies

2.1 Long Term Evolution (LTE) and LTE-U

The 3GPP improved LTE through different releases toward the path to the 5G, for example release 8 which is called LTE-Advanced, or LTE-A was actively investigated with a research and standardization perspective. It supports Carrier Aggregation (CA) possibility to aggregate multiple Rel-8 Component Carriers (CC) on MAC level to increase system capacity or user throughput with the scheduling flexibility (initially either Frequency Division Duplexing (FDD) or Time Division Duplexing (TDD), later with combined FDD and TDD). Also, enhanced MIMO with up to (8x8) MIMO antennas at eNodeB to improve spectral efficiency.And Heterogeneous Networks (HetNet) – network with different types of nodes to allow home usage of LTE and increasing capacity in hotspots. This includes such concepts as HeNB and enhanced Inter-Cell Interference Coordination (ICIC). Introducing automation to network operation in the means of Self-Configuration, Self-Optimization and Self-Healing. Multicast Broadcast Multimedia Services – with the Single Frequency Network (SFN) to allow broadcasting the same service content within different cells using LTE radio, coordinated Multi-Point Transmission/Reception (CoMP) (Temple, 2017).

Cellular/WiFi interworking (Ismail et al., 2013) is only possible when subscribers can adaptively use either licensed LTE or unlicensed WiFi networks for provisioning multimedia services. It requires communication management through asynchronous radio access technologies (RATs), and modifications of the protocol stacks and interface functionalities. This is even though Celluar/WiFi interworking offers a capacity surge for operators. The requirement complicates resource allocation and user service hard to guarantee. Taking these issues into account, LTEU technology was introduced as a part of LTE Release 13, as this enables users to access the licensed and unlicensed spectrum under a unified LTE network infrastructure (Chen et al., 2016).

2.1.1 Hetnet in LTE-A

LTE-A introduced HetNet with different types of deployment represented by the following technologies: Macrocells are designed to provide large coverage. They are not made for crowded urban and indoor environments, as they cannot provide high throughput. Instead, several techniques were proposed which gives universal wireless coverage: Carrier aggression increases the bandwidth allocated to the UEs, which can be done by enabling concurrent use of different frequency results. Multiple-antenna solutions give substantial diversity and multiplexing gains. Multiple cells employing CoMP can coordinate their timings to serve UEs with unfavorable link conditions, which makes this technique good to mitigate outage at the cell edges. These solutions, designed for homogenous cellular networks, are however reaching their theoretical limits. More recently, studies in the deployment of Femtocells have shown the potential for incredible system gain achievements. Femtocell base stations (BSs) are lowpower, miniature wireless access points, deployed at a home and then connected to backhaul networks via residential wireline broadband access links, e.g., digital subscriber lines (DSL), cable broadband connections, or optical fibers. Femtocells serves a dozen of active users with a range of less than 50 m (Lopez-Perez et al., 2014).

Macro-eNodeBs are expensive to install so installing new base stations to improve signal quality is not always possible. This is the reason of small cells emerge. These cells are made of cheaper equipment with small coverage areas, which makes them optimal for hotspots. In addition to femtocells, which are intended for indoor placement, the other small cell type is picocell which are intended for outdoor use and have a higher transmit power. Femto- and picocells can use different access policies. Open access policies are usually used by picocells as they can offload users from the surrounding macrocells. Femtocells on the other hand uses close access policy as the users usually want all of the cell capacity for themselves (Bjerke, 2011).

Femtocells are made by the telecommunication industry to give a high performance and high-quality service for home users. Therefore, they are installed indoors and connected to the broadband service modem similarly to a WiFi access point. They have the same functionality as macrocells, however they are restricted by a 10-30 m range and can only be used by a few users. Nonetheless they are low cost. The installation of femtocells is like a plug and play device which requires a little planning, as it has a built-in self-configuration which minimizes the impact on the macrocells through self-provisioning (Hagos, 2015).

A relay is like a repeater that is already commonly used in 2G and 3G technologies, basically, relay nodes act as low power base stations that are used for increasing the coverage in the regions where the coverage is low. To implement a relay station, the cost of purchase will be less than having eNodeB. The connection between the eNodeB and relay is wireless which decreases the implementation of infrastructure cost. A repeater's job, for the existing base stations is expanding the coverage to areas where cannot be reached by the base station, or in the edge areas where there is a high Signal to Interference and Noise Ratio (SINR), and also in the areas where shadowing occurs or for providing coverage in indoors. The relay of Layer 1 expands and sends uplink and downlink signals between the base station and UE. The main disadvantage of the relay in Layer 1 is while expanding and sending the signal, the undesirable interference like inter call interference and noise will be forwarded with it. The relay of Layer 2 works in a way of extracting or demodulating and decoding the arriving signals. And then before the amplified version of the signal is transmitted, the signal will be re-modulated and encoded. This approach will solve the problems occuring in Layer 1 relay which there will be no inter call interference or noise re-transmitted or amplified with it. The relay of Layer 3 is working in the same way of relay in Layer 2. With having additions like having a unique Physical Cell ID for ensuring that UE will know that it is connected to a base station which is a relay node. And the relay of Layer 3 was standardized by 3GPP for releasing with LTE-A (Panwar & Mallick, 2015).

2.1.2 LTE-A Hetnet Integration with WLAN

The WiFi network met with a huge traffic growth and will continue to meet more in the coming years because of the rising number of more powerful UE (user equipment), and with more attractive user applications. Actually, according to the predictions, the mobile network operators should improve the capacity of their network by a factor of 100x, to be able to meet the customer demands by 2020. Thus, the inter-working between the WLAN and LTE gained attention. LTE can take advantage from the licensed companies to fulfill the Quality of Service (QoS) and be the modality of controlling ad-hoc WiFi broadcasts. The WiFi can give the ability to the operators to cost-effectively provide their networks and reach to the large bandwidth in the unlicensed network spectrum. If both technologies are integrated efficiently, it will be a good opportunity for the future wireless systems to improve their spectral efficiency and realize effective traffic offloading/ aggregation between both technologies.

For providing a united LTE-A/WiFi access, the coupling concept was used which has three main levels depending on how the integration was done (loosely, tightly, and very tightly coupled) between both technologies. The coupling types are explained in the following (“supportcenter”, 2021):

1. Loose coupling: The WiFi and LTE networks are independent physically and their common point is their connection to the same IP network for the internet. Both of the networks have a separated access network, and all the functionalities are related to scanning, association and connection will be done within the coverage area of that technology. This means that, both standards can co-exist together without changing any layer of the network.
2. Tight coupling: The access points in WiFi are directly connected to the EPC (Evolved Packet Core) of the cellular network, but still the UE needs to use the time-consuming WiFi security mechanism, and this approach is standardized as a solution by the 3GPP in the release 10. The interconnection architecture of tightly coupling between wireless access networks will participate in one of the collaborating radio or core network of the wireless access network. The most popular example is the interconnection at the RNC, GGSN, or SGSN level of the 3G network. Tight coupling is more complex than loose coupling, and for connecting to wireless access networks it needs gateways to be installed.
3. Very tight coupling: The access points of the WiFi are connected by an LTE eNodeB and covered by that eNodeB. When the control functions like mobility and security are held in the LTE network, the data traffic is offloaded to WiFi network.

3. Problem Statement

The rapid growth of the telecommunication technologies will lead to the increase in number of devices and consequently will lead to the load on the network operator that has a limited resource. With the introduction of Long-Term Evaluation Advance (LTE-A), the increase in capacity of the network was witnessed since it offers high bandwidth and high data rate. However, considering the homogenous network that copes with the increase of the capacity of the network is not a practical solution. This is due to the high cost of the base station (BS), the difficulty of deploying BS in some harsh areas, and frequency planning issue. This is why LTEA introduced small cells beside the traditional macrocell.

These small cells are represented by the picocell and femtocell with low power BS and less cost. All these types will form what so called heterogeneous network of HetNet and all these technologies need license operation to work.

Wireless local Area Network (WLAN) achieved big success in terms of deployment, and it does not need any license for operation. The co-existence between LTE-A with (wireless fidelity) WiFi has been one of the main research problems that the 3GPP is interested in LTE-U. The aim is to offload the LTE-A network by introducing WLAN to the network. However, the interworking of both technologies is a challenging issue, and it can be solved by proposing an architecture that combines both of them and then building on any network functionality.

In the paper, an integration architecture based on loose coupling concept between WiFi and LTE heterogeneous networks (HetNet) was proposed. The architecture was built and simulated using riverbed network simulator, it was proved that the architecture is working appropriately. As well, different load balancer algorithms were tested on the architecture and numerical results were obtained with demonstrating which algorithm is better.

With the rapid increase of the data traffic in the wireless network and increasing demands of the users for connecting anytime anywhere to the radio access network with various types of applications, the current wireless network cannot afford the capacity increase in network load with its actual type of homogenous network. This will lead to the violation of the Quality of Service (QoS) of the different applications regardless of their types, real-time and non-realtime. For solving this problem, the Loose coupling architecture in heterogeneous network (Hetnet) was applied in the paper, including different technologies like Wireless LAN (WLAN) based WiFi and LTE-A (Macrocell, Femtocell and Picocell) and with experimenting different real time and non-real time applications like Hypertext Transfer Protocol (HTTP), Voice and Video between the users of different technologies. And finally, for showing that the created scenarios are working well the load balancing concept as a network functionality is used for assuring the efficiency and showing the realistic of the created network outcomes.

4- LTE-A HetNet and WLAN Loose Coupling Architecture: Simulation Methodology and Numerical Results

Loose coupling architecture is considered in the paper for the integration between LTE-A HetNet and WLAN. This is why this paper contains five main parts, namely, simulation methodology and scenarios, node configurations and attributes of loose coupling scenario, description of standard applications, running the simulation scenarios, and finally simulation results of loose coupling architecture. In the paper, Riverbed Modeller 18.8 is used for creating the simulations, which is the latest version of this modeler by modelling different network types and technologies (including VoIP, TCP, OSPFv3, MPLS, IPv6, WLAN, LTE,etc.) (Saranya & Maheswari, 2018).

4.1 Simulation Environment Design

The network topology showed in figure 1, is designed based on Loose Coupling architecture which consist of two technologies: WiFi based WLAN, and LTE-A Hetnet. The composition of HetNet is Picocell, Femtocell and LTE Macrocell. In this section all the nodes and links of every technology that is used in the architecture will be described in detail.

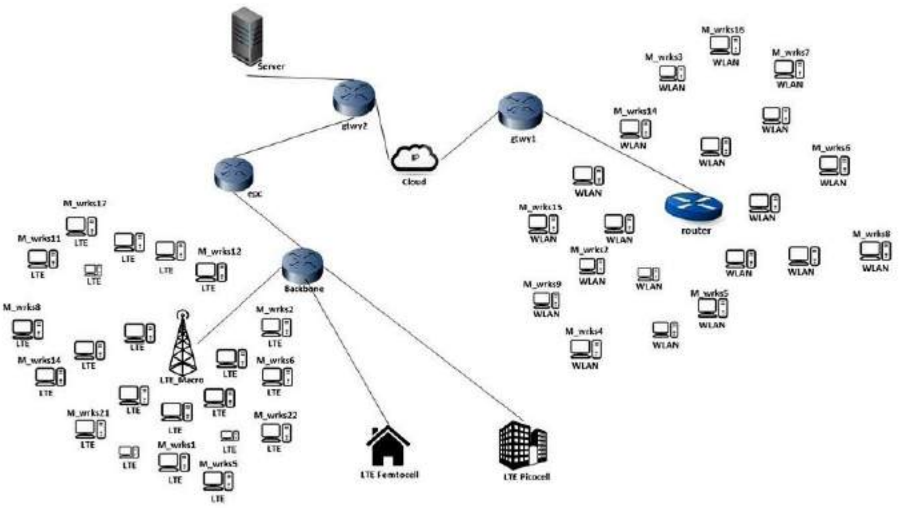

Figure 1: Loose Coupling network topology of WLAN and LTE-A

In the architecture of figure 1, there are two gateways, one is between the server and cloud, and the other is between cloud and WLAN (Wireless LAN) router. Their types are ethernet4_slip8_gtwy_adv, this node model represents an IP-based gateway supporting four Ethernet hub interfaces, and eight serial line interfaces. IP packets arriving on any interface are routed to the appropriate output interface based on their destination IP address. The Routing Information Protocol (RIP) or the Open Shortest Path First (OSPF) protocol may be used to dynamically and automatically create the gateway's routing tables and select routes in an adaptive manner (Saranya & Maheswari, 2018).

Basically, in the architecture, cloud is used to connect the UE to the server through the gateways. Its model is ip32_cloud, the node model represents an IP cloud supporting up to 32 serial line interfaces at a selectable data rate through which an IP traffic can be modelled. This cloud requires a fixed amount of time to route each packet, as determined by the "Packet Latency" attribute of the node. Packets are routed on a first-come-first-serve basis and may encounter queuing depending on the transmission rates of the corresponding output interfaces (Saranya & Maheswari, 2018).

Figure 2: Loosely Coupled LTE Femtocell

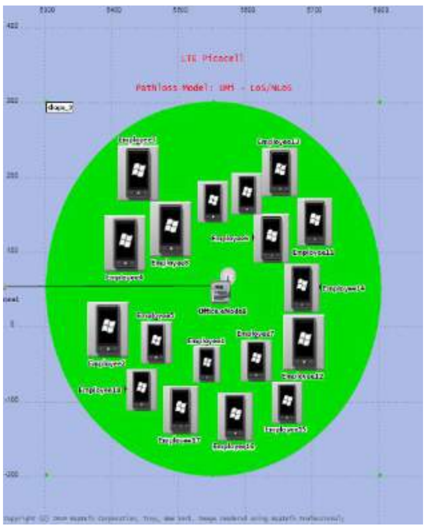

Figure 3: Loosely Coupled LTE Picocell

In the Riverbed modeler, generally there are two types of UEs, fixed and mobile nodes, and there are different model types according to the technology used in the topology. In the network topology of figure 1, there are two different types of devices according to the technology, which are WLAN (Wireless LAN) devices. LTE UE totally in the topology and there are 75 user devices that are wirelessly connected to the eNodeB using LTE interface and WLAN router through one underlying WLAN connection at 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The devices are of different types, they are 20 WLAN computers or workstations, 15 Android devices in Femto cell, 18 Windows phones in Pico cell, and 22 computers or workstations in Macrocell as they are shown in the figures 1, 2 and 3.

We have a WLAN network using routers as access points (AP) for users and connecting the server through a cloud. Because the aim of the paper is to have a loose coupling scenario between different technologies, another technology which is LTE-A (Long-Term Evaluation Advanced) is used, which contains different types of cells that are Macrocell, Femtocell cell and Picocells, which create the topology of HetNet. In the network topology of figure 1, the Macrocell through the backbone is connected to the EPC and then to the server. It has the biggest number of UEs connected to it compared to the other technologies which have 22 users, because it has more ability than other technologies.

As Riverbed Modeller contains a collection of standard applications including HTTP, FTP, Email, Database, peer-to-peer File Sharing, Voice, Video Conferencing, etc. In the network topology of this part, three applications are used which are HTTP with Light Browsing, Voice with IP Telephony, and Video conferencing with Low Resolution Video.

In this research, the behavior of voice application is studied by applying it in different technologies and between these technologies. Here, the most important QoS parameters for voice are selected, like jitter which indicates the inter-delay between the voice packets, the delay of transmitted packet and the Mean Opinion Score (MOS) value that measures the voice quality.

Video Conferencing is a real-time application that can be either client-server or peer-to-peer application depending on the context. And it may use User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) depending on the type of architecture. Video has many QoS parameters. The end-to-end delay is the parameter that is worked in this study.

4.2 Simulation Results

In order to investigate the feasibility and the validity of the proposed loose coupling architecture, simulation is carried out to collect the parameters of QoS of different applications. All the simulations run for 10, 20, 30, 40 and 50 minutes and each time the average of the values of these parameters are collected. Table 1 depicts all the values for each selected parameter. Page response time is selected for each technology to be measured and the tables show similar results for all of them. This shows that all the devices in the topology are connected to the server of the HTTP, and they can access it properly within a short time without violating the QoS parameter. As for the voice, the jitter is 0 which is a good value as it depicts the inter packets delay, which is an ideal value since a perfect environment is supposed without so much load in the network. As for the MOS value, which is almost 4 in all technologies demonstrates that the quality of experience of the voice by the users is very good. The value of the voice shows that the proposed architecture guarantees the QoS and Quality of Experience (QoE) in the same time. Finally, for the video values, again same values were noticed, except for the WLAN, this is due to the big size of the frame for WLAN compared to the LTE-A. However, the same quantity of packets is sent and received. As well, the values of the traffic sent and received are different. And this is a normal behaviour due to the nature of traffic of video as there is no uniform shape of packet inter-arrival. Also, there is no limited data rate for it, it can range between a minimum and a maximum rate.

All these results illustrate that the architecture is working efficiently and correctly as the communication is established well between WLAN devices and LTE UEs through voice and video applications.

Table 1: Loose coupling simulation results

WiFi	HTTP	Page Response Time (sec)	0.108382
	Voice	Jitter (sec)	0
		MOS value	4.0161
		Packet End to End Delay (sec)	0.10619
	Video	Traffic Sent (bytes/sec)	141091
		Traffic Received (bytes/sec)	141062
		Packet End-to-End Delay (sec)	0.08741
Femtocell	HTTP	Page Response Time (sec)	0.212427333
	Voice	Jitter (sec)	0
		MOS value	4.0077
		Packet End to End Delay (sec)	0.10617
	Video	Traffic Sent (bytes/sec)	141091
		Traffic Received (bytes/sec)	139421
		Packet End-to-End Delay (sec)	0.020401
Picocell	HTTP	Page Response Time (sec)	0.217175
	Voice	Jitter (sec)	0
		MOS value	4.0131
		Packet End to End Delay (sec)	0.10948

	Video	Traffic Sent (bytes/sec)	141091
		Traffic Received (bytes/sec)	141062
		Packet End-to-End Delay (sec)	0.088
Macrocell	HTTP	Page Response Time (sec)	0.247908636
	Voice	Jitter (sec)	0
		MOS value	4.0144
		Packet End to End Delay (sec)	0.108
	Video	Traffic Sent (bytes/sec)	157,978
		Traffic Received (bytes/sec)	157,978
		Packet End-to-End Delay (sec)	0.0900028

5- Load Balancing in LTE-A HetNet and WLAN Loose Coupling Architecture

A loose coupling architecture between LTE-A HetNet and WLAN is designed and verified that the architecture is working well through deploying several applications with testing their connectivity and their QoS. This part introduces the function of load balancing to the same architecture to prove that it is working properly, and any network functionalities can be introduced in it, such as scheduling, QoS, mobility, load balancing, etc. This part introduces the simulation methodology and scenarios, node configurations and attributes of load balancing scenario, description of standard applications, and finally simulation results of load balancing architecture.

The network topology as shown in Figure 4, is a load Balancing scenario that contains two technologies, similar to the scenario of Loose Coupling in part 4: namely WLAN LTE-A based Femtocell, Picocell and Macrocell. In this section, only the necessary and newly nodes and links are added to the architecture of loose coupling.

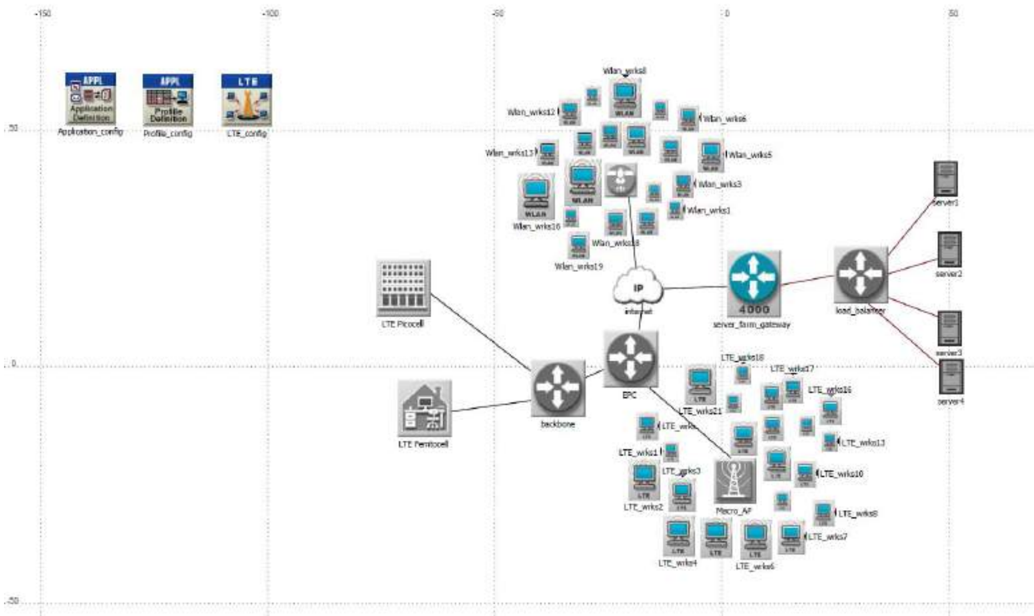

Figure 4: Load Balancing integrated in loose coupling architecture of WLAN and LTE-A HetNet

In the paper the following load balancing algorithms were studied:

- Round Robin

  This algorithm is one of the static load balancing algorithms which does not consider previous states. It is a simple algorithm, for task allocation it uses the Round Robin method. It works with randomly selecting the first node, and then in a Round Robin manner allocates the task equally to all the other remaining nodes. The main advantage of this algorithm is that any interposes communication is not needed. Some tasks will be heavily loaded, because there is no previous processor's running time information. For solving this problem, WRR (Weighted Round Robin) algorithm is suggested. In the WRR algorithm, for each node, a specific weight is specified which will get the requests based on their own weights (Al-Naib et al., 2013) (Serra & Rodrigues, 2013).

- Random

  The random load-balancing algorithm is one of the static algorithms and it works in a very simple way. It randomly divides the load over the servers and generates a random number for selecting a server with sending the current connection to it. The system has a set of load balanced servers, and a random number generator is used for determining which server gets the next connection (Silva et al., 2016).

- Server load

  The load balancer distributes the load in terms of request per second to each server in the pool, the request here can be initiated by any non-real time application to the load balancer. Generally, the load balancer calculates the number of requests and distributes evenly to the servers connected to it.

- No Load Balancing

  This case considers that there is no load balancer, as the router has only one interface connected to one server, this server treats all the requests coming from the load balancer. Note that the load functionality of the load balancer is disabled on the router. Later it only forwards requests without any processing.

5.1 Simulation Results and Analysis

In the network topology of this paper, one application is used which HTTP with Light is browsing. Note that any kind of non-real-time application can be chosen to study its performance, since load balancer in riverbed is used only to balance traffic of non-real-time application. The studied parameters that are studied in this part for load balancer scenario are:

- CPU Utilization (%)

  This parameter is about the utilization reports of a 'Simple CPU' in percentage (%). For modeling the IP packet forwarding delays and application processing delays, the 'Simple CPU' is used. This parameter also considers the background CPU utilization specified in the CPU Background Utilization attribute (Saranya & Maheswari, 2018).

- Processing Delay (sec)

  This is the experienced Delay by an IP datagram though the IP layer (i.e., it is a delay duration between arriving the packet at the IP layer and to the time it will be forwarded) (Saranya & Maheswari, 2018).

The delay includes:

1. Queuing delay (to get to the head of the queue to start processing) (Saranya & Maheswari, 2018)
2. Processing delay (based on the processing speed/forwarding rate) (Saranya & Maheswari, 2018)

- Load (requests/sec)

  This is the ratio of arriving http requests to the server. Also, those requests may belong to various sessions maintained at the server. And the requests for the same session are queued until the first request is completed (Saranya & Maheswari, 2018).

- Task Processing Time (sec)

  This is the time that the server required for processing an http request. And it is measured as the time between a request arrives at the server to the time it is processed completely by the server (Saranya & Maheswari, 2018).

5.2 Simulation Results of Load Balancing

In this section, the simulation results of all four scenarios (Random, Round Robin, Server load and No-Load Balancing) that are created for load balancing in this paper with their used applications and selected statistics with important parameters in different technologies (WLAN, LTE-Macrocell, Femtocell, and Picocell) are shown in detail. The results are shown as "Top Results" as mentioned in pervious part and in different durations of running time (10, 20, 30, 40, 50) for each scenario. Each parameter is explained in a separate section with graphs and explanations. For showing the similarities and differences between the load balancing algorithms in each server and also in the load balancer node itself, only one parameter is shown in each graph with containing all four parameters of load balancing. This section contains two main parts: first CPU Utilization and Processing Delay in load balancer node, second Task Processing Time and CPU Utilization in all four servers with all used algorithms.

5.2.1 Comparison of Different Load Balancing Algorithms in Load Balancer Node

- CPU Utilization Parameter

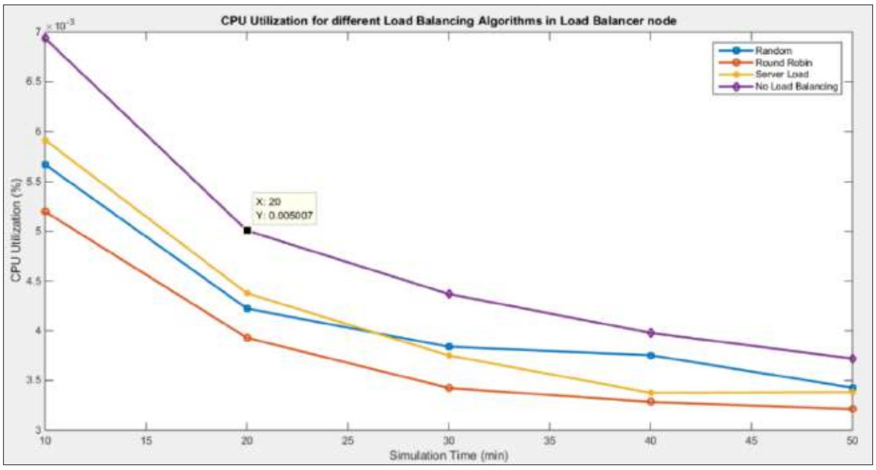

Figure 5: CPU Utilization (%) for different Load Balancing algorithms in Load Balancer node

Figure 5 shows the average percentage of the CPU Utilization parameter in the Load balancer and in the four different load balancing algorithms during different simulations times. As shown in the figure, in the scenario the simulation time was 10-minutes. The No-Load Balancing algorithm has the highest value of the CPU utilization percentage comparing to the other algorithms and the Round Robin algorithm has the lowest percentage value. This is because, in no load balancing scenario the load that goes out from the load balancer will be all directed to the one server which is Server1, and this leads to that in this case the traffic drop will be zero for load balancer along with having an increased value of processing delay. When the simulation time is increased in the scenarios, the CPU usage of all the load balancing algorithms continue decreasing. But in 30-minutes scenario, the Random algorithm starts to increase its percentage value and it is being more than the value of Server Load. This increment continues until the 50-minutes scenario, then it decreases again, and its percentage value becomes almost same or equal to the value of Server Load algorithm. All of the algorithms' percentage values in 50-minutes scenario decrease, but still in the percentage value of No-Load Balancing algorithm is the highest and the Round Robin is the least.

- Processing Delay Parameter

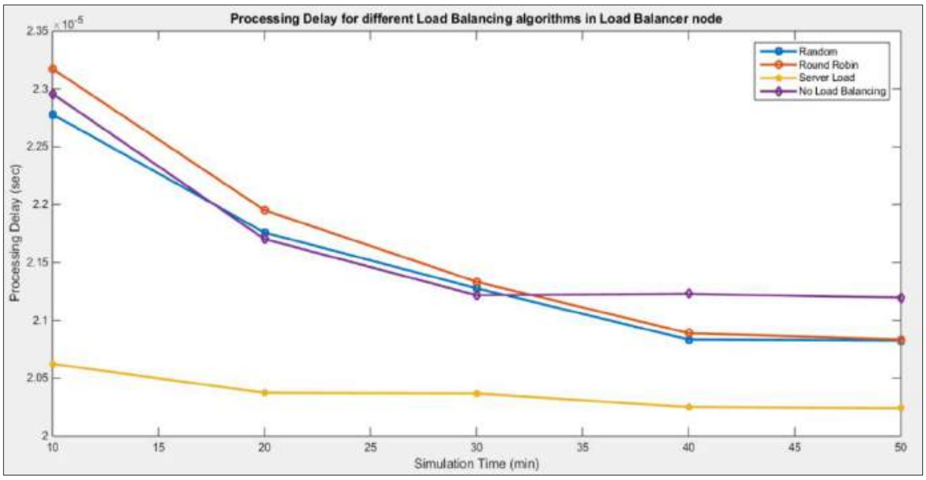

Figure 6: Processing Delay for different Load Balancing algorithms in Load Balancer node

Figure 6 shows the average values in seconds of the Processing Delay parameter in the Load balancer. And in the four different load balancing algorithms during different simulations times. The delay can be defined as a delay of the duration time between the arriving packets to the destination and to the time of sending them again from the load balancer. Also, delay includes queuing delay and processing delay. The queuing delay is the delay of a packet when it attempts to reach the queue's top to be processed, and the processing delay is the delay which depends on the processing speed per the forwarding rate. As shown in the figure, the load balancer has the lowest processing delay in the server load algorithm scenario. If it is compared to the other three load balancing algorithms, which is in the server load case, the load balancer will choose the lowest load server among all the servers and will distribute the load evenly on all servers, Thus, the load on each server will be lower, even if compared with no load balancing case, so this makes the processing delay values become less than all the used algorithms. At the 10minutes scenario, the Round Robin algorithm case has the highest processing delay between all other algorithms.

Then, it starts to decrease along with decreasing other algorithms' values until the 40-minutes scenario. The no load balancing algorithm starts to increase and progress on the other algorithms along with Random and Round Robin algorithms. Their values are near to each other until in 50-minutes scenario. Then, their value become almost same and at that time the value of no-load balancing algorithm becomes the highest among all the load balancing algorithms in this research.

5.2.2 Comparison of Different Load Balancing Algorithms in the Servers

- CPU Utilization Parameter

1. In Server 1

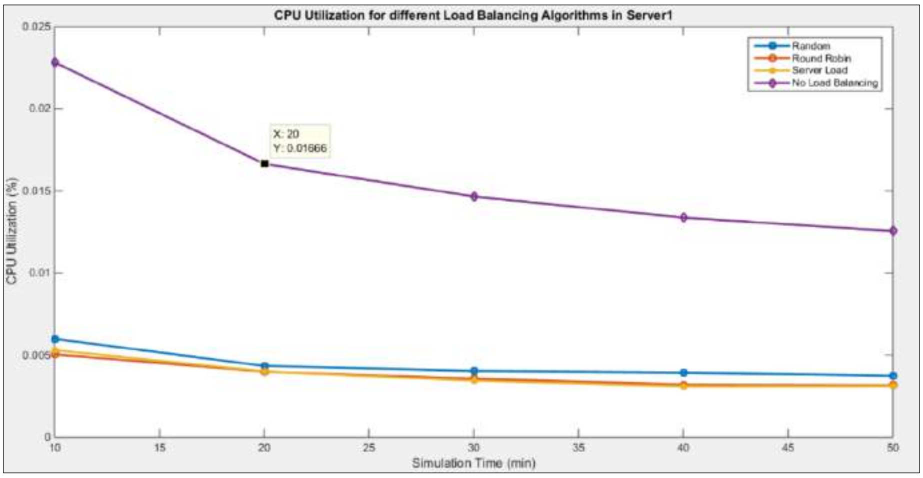

Figure 7: CPU Utilization (%) for different Load Balancing Algorithms in Server1

Figure 7 shows the average percentage of the CPU Utilization parameter in the server1 and in the four different load balancing algorithms during different simulation times. As shown in the figure, the server1 has the lowest average value of CPU Utilization parameter in the case of No-load balancing algorithm. Because the load balancer directly sends all the load to the server1, so the load balancer has zero traffic drop and more processing delay in the case of no-load balancing algorithm. And in the server1 there is a high load with high CPU Utilization and there is nothing for the other three servers. Also in the figure, the second lowest CPU Utilization percentage is in the case of Random algorithm. It continues to be the second lowest in all the running simulation times, because the Random algorithm works in a random manner, and it arranges the servers along with loads randomly. For the Round Robin and Server Load algorithms, as it is shown in the figure, their percentage values of the CPU Utilization among all the running simulation times are very close, almost same and they have the highest values. The Round Robin algorithm works in a way that the load balancer distributes the loads uniformly, chooses the servers in turn, and Server load. Also, distributes the load equally but it chooses the server that has the lowest load.

2. In Server 2

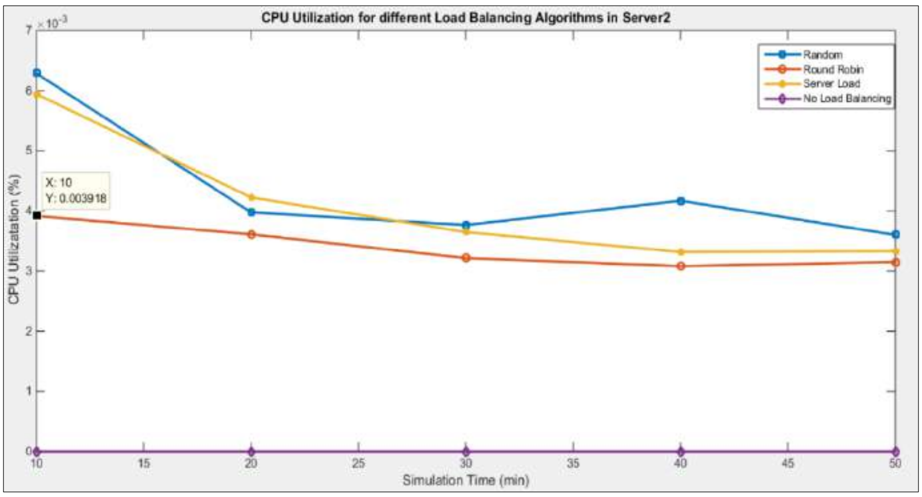

Figure 8: CPU Utilization (%) for different Load Balancing Algorithms in Server2

Figure 8 shows the average percentage of the CPU Utilization parameter in the server 2 with four different load balancing algorithms during different simulations times. As shown in the figure, the no load balancing algorithm has zero percentage value of CPU Utilization in all the running simulation times. Because there is no load on server2 as all the load has been forwarded to server1. As shown in the figure, the Random algorithm has the highest percentage value of CPU Utilization in server2 in all the simulation times, except in 20-minutes simulation time in which Server Load algorithm has a higher percentage value. And also, it has the second highest percentage value. And the Round Robin algorithm has the lowest percentage value comparing to other algorithms.

3. In Server 3

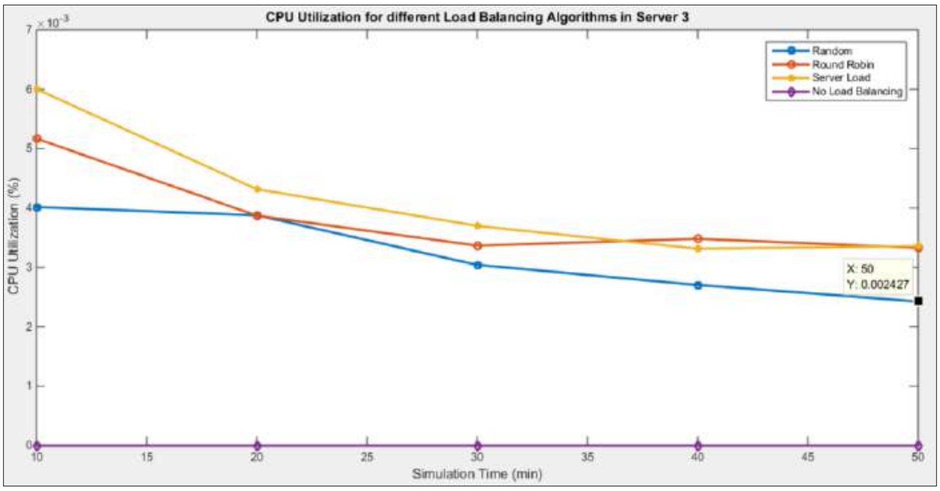

Figure 9: CPU Utilization (%) for different Load Balancing Algorithms in Server3

Figure 9 shows the average percentage of the CPU Utilization parameter in the server 3 with the four different load balancing algorithms during different simulations times. As shown in the figure, same as in previous figure of server 2, the no load balancing algorithm has zero percentage value of CPU Utilization in all the running simulation times. Because there is no load on server 3 as all the load is forwarded to server1. For the other three algorithms, the Server Load algorithm has the highest percentage value of CPU Utilization in server3, except in 40minutes simulation time, in which the Round Robin algorithm has a higher percentage value. And also, it has the second highest percentage value, and these two algorithms from the beginning till the end of their values are decreasing together until the end. Within 50-minutes simulation time, their values are near to each other or almost same. Lastly, the Random algorithm has the lowest percentage value in all the simulation times comparing to other algorithms, except in 20-minutes simulation time, it is value is near to the Round Robin’s value.

4. In Server 4

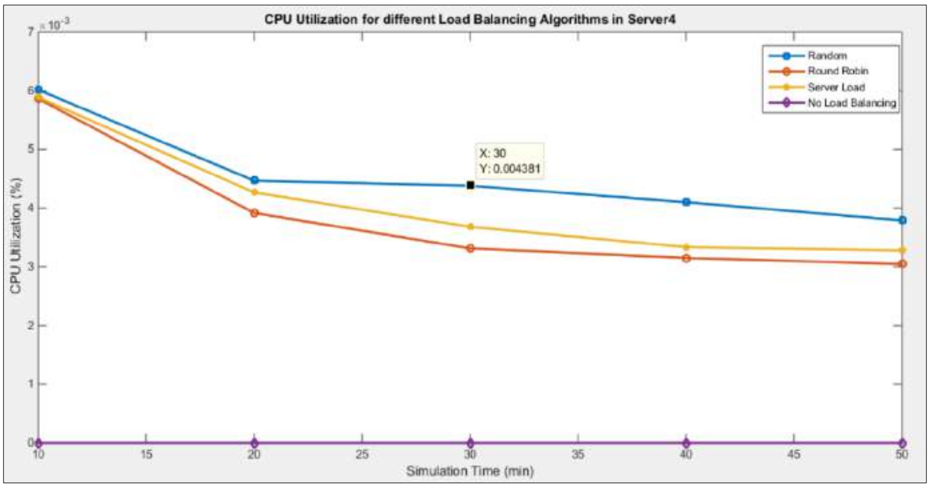

Figure 10: CPU Utilization (%) for different Load Balancing Algorithms in Server4

Figure 10 shows the average percentage of the CPU Utilization parameter in the server 4 with four different load balancing algorithms during different simulations times. As shown in the figure, same as in previous figures of server 2 and server 3, the no load balancing algorithm has zero percentage value of CPU Utilization in all running simulation times. Because there is no load on server 4, as all the load is forwarded to server1. As shown in the figure, in the 10minutes scenario, the server in other three algorithms have nearly same percentage value of the CPU Utilization. But along with increasing the running simulation time, the values for the three algorithms in the server are changing and continue in decreasing. The Random algorithm has the highest percentage values among all the algorithms in all the simulation times. The second highest value is for the Server Load algorithm and the lowest value is for the Round Robin algorithm.

- Task Processing Time

1. In Server 1

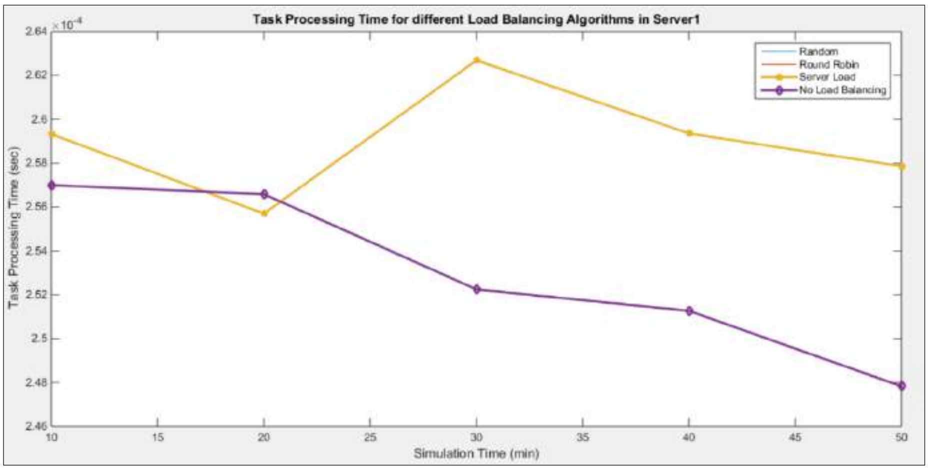

Figure 11: Task Processing Time for different Load Balancing Algorithms in Server1

Figure 11 shows the average values in seconds of the Task Processing Time parameter in the Server 1 with four different load balancing algorithms during different simulations times. The Task Processing

Time can be defined as the time required by a server for processing an application request, so it is the duration time between arriving a request to the server and to the time of completely processing that request by the server. As shown in the figure 11, only two lines are appearing, one is for the Server Load algorithm and the other is for No Load Balancing algorithm with four different types of load balancing algorithms used in the scenarios. This is because the values of the Random and Server Load are close to each other. That is why, the chart line of Random algorithm cannot be seen in the figure, and same in Round Robin and No-Load Balancing algorithms. So as shown in the figure, the values of Server Load and Random algorithms are the highest among all the simulation times, except in the 20-minutes scenario, their values are less than the values of No-Load Balancing and Round Robin algorithms. But then their values start to increase until in the 50-minutes scenario, they are higher than the values of No-Load Balancing and Round Robin algorithms.

2. In Server 2

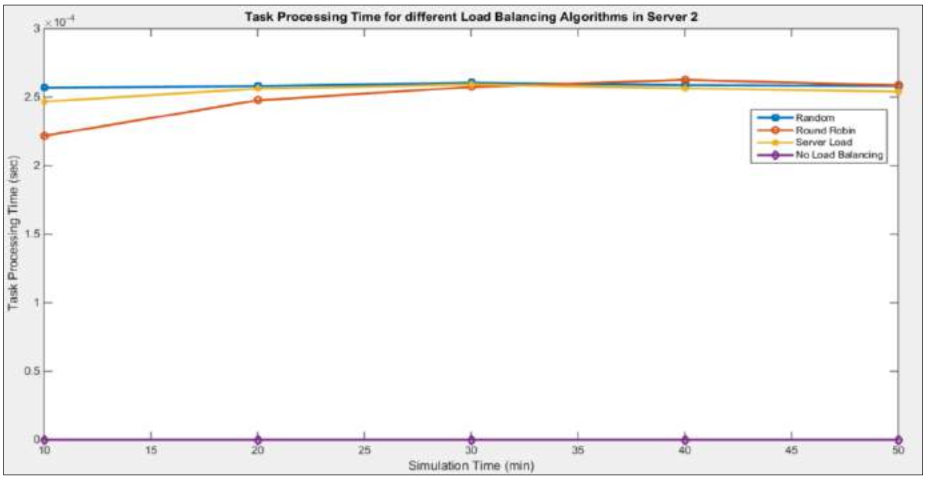

Figure 12: Task Processing Time for different Load Balancing Algorithms in Server2

Figure 12 shows the average values in seconds of the Task Processing Time parameter in the Server 2 with four different load balancing algorithms during different simulations times. As shown in the figure, the no load balancing algorithm has zero values of Task Processing Time in all the running simulation times. Because there is no load on server2 as all the load is forwarded to server1. As shown in the figure, at the beginning in the 10-minutes scenario, the other three algorithms of load balancing have different values as the Random algorithm has the highest value and the Round Robin algorithm has the lowest. However, the values of Random and Server Load algorithms are near to each other. But generally, from 20-minutes scenario until the end which is 50-minutes scenario, the values of the three algorithms in the Server 2 continue to be stable and close to each other or almost having same values.

3. In Server 3

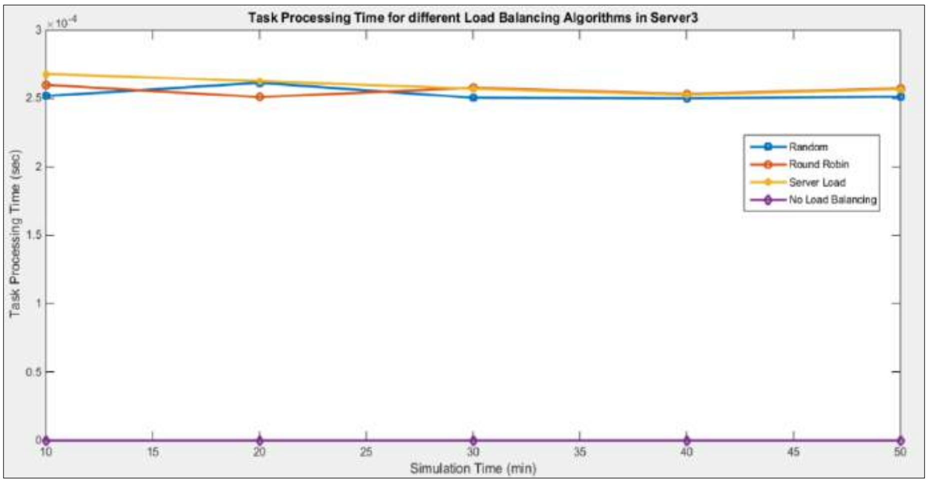

Figure 13: Task Processing Time for different Load Balancing Algorithms in Server3

Figure 13 shows the average values in seconds of the Task Processing Time parameter in the Server 3 with four different load balancing algorithms during different simulations times. As shown in the figure, same as in previous figure of server 2, the no load balancing algorithm has zero values of Task Processing Time in all the running simulation times. Because there is no load on server 3 as all the load is forwarded to server1. At the beginning of the chart or at the 10-minutes scenario, the Server Load algorithm has the highest value, Random algorithm has the lowest and Round Robin is in between. But those values continue to change among different simulation times, while the changes are not that much as their values are near to each other and in the end at 50-minutes scenario, the value of Random algorithm remains to be the least among the three algorithms besides the no load balancing algorithm.

4. In Server 4

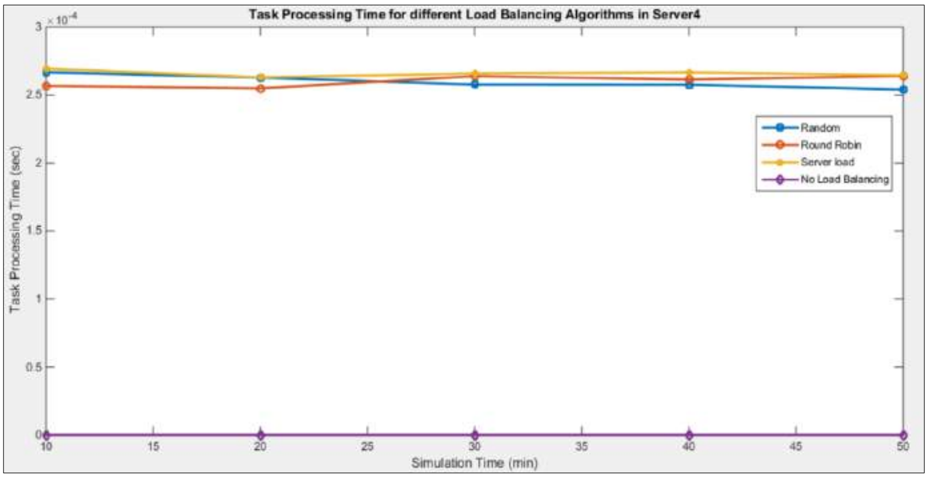

Figure 14: Task Processing Time for different Load Balancing Algorithms in Server4

Figure 14 shows the average values in seconds of the Task Processing Time parameter in the Server 3 with four different load balancing algorithms during different simulations times. As shown in the figure, same as in previous figures of server 2 and server 3, the no load balancing algorithm has zero values of Task Processing Time in all the running simulation times. Because there is no load on server 4 as all the load is forwarded to server1. As shown in the figure, the values of the three other algorithms are close to each other. In the 10-minutes scenario, the Server Load algorithm has the highest value in all the simulation times, while the value of Random algorithm is very near to Server Load algorithm which is a little bit less than it, as the Round Robin algorithm has the least value among the three algorithms. In the 30-minutes scenario, the Round Robin algorithm advancing the Random algorithm and in the 50-minutes scenario the Random is the least among the three algorithms of load balancing.

6. Conclusion

Loose Coupling architecture combines different technologies to work together and get benefit from the interworking technologies to make the network's QoS better and increasing the capability of meeting the user's demands. In the paper, the loose coupling is worked on heterogeneous network which consists of WLAN based WiFi and LTE-A technologies. As the LTE-A introduced the idea of having small cells in the network so they can solve the problem of coverage and capacity along with increasing network's QoS.

In this research, through the Riverbed Modeller, a loose coupling architecture is proposed and simulated for the integration of both technologies with having the small cells (Macrocell, Picocell and Macrocell) in LTE-A. And by testing the performance of the network with it is feasibility in terms of connectivity of different types of applications like non-real time (HTTP) and real-time (VOIP – Voice over IP and Video conferencing) with having a big number of user devices, so with having huge number of mobile users in the network, the traffic loads are increased. Simulation results showed that, the proposed loose coupling architecture guarantees the QoS for all the technologies with reasonable values for the parameters of QoS for each application.

As loose coupling is a generic architecture, any network functionality can be integrated into it while ensuring the connectivity among all the technologies. This is why, in the paper, a load balancing functionality is introduced in order to show that the architecture is a valid and flexible to any kind of functionalities. Hence, different types of load balancing algorithms are studied, and their performance are evaluated in terms of CPU utilization, load, processing delay, etc.

The main contribution can be summarized as follows: Proposing a loose coupling architecture of integration between WLAN and LTE-A for offloading traffic from WLAN to LTE or vice versa. Simulating the loose coupling architecture through Riverbed with testing its feasibility and performance in terms of connectivity of different types of applications. Adding a functionality on the top of loose coupling architecture which is represented by the load balancing testing different types of load balancing algorithms on the loose coupling architecture and studying their performance through a comparison work.

Since in this research, the loose coupling architecture between two technologies (WLAN and LTE-A) is worked, for improving this research the next step in the future will be implementing the tight coupling on the same scenario of the paper. In the tight coupling, the access points of the WiFi can be directly connected to LTE-A's EPC (Evolved Packet Core). And further step may be the implementation of the very tight coupling which is represented by LTE-U in which the access points of the WiFi are connected to the LTE-A's eNodeB. But at the same time, they are covered by the eNodeB, and while the control functions are saved in the LTE, the traffics will be offloaded to the WiFi network.

References

Alisa, Z. (2013). Evaluating the Performance of Wireless Network using OPNET Modeler. *International Journal of Computer Applications*, *62*(13), 22-28. https://doi.org/10.5120/10140-4946

Al-Naib, I., Singh, R., Shalaby, M., Ozaki, T., & Morandotti, R. (2013). Enhanced Q-factor in Optimally Coupled Macrocell THz Metamaterials: Effect of Spatial Arrangement. *IEEE Journal of Selected Topics in Quantum Electronics*, *19*(1), 8400807-8400807. https://doi.org/10.1109/jstqe.2012.2202639

Autoridade Nacional de Comunicações, A. (2021). *Autoridade Nacional de Comunicações*. Autoridade Nacional de Comunicações. Retrieved on 20 May 2019, from https://www.anacom.pt/render.jsp?contentId=1100003.

Bjerke, B. (2011). LTE-advanced and the evolution of LTE deployments. *IEEE Wireless Communications*, *18*(5), 4-5. https://doi.org/10.1109/mwc.2011.6056684

Chen, J., Valsecchi, M., & Gegenfurtner, K. (2016). A tight coupling between finger and oculomotor commands. *Journal of Vision*, *16*(12), 375. https://doi.org/10.1167/16.12.375

Choque, J., Díez, L., García, A., Agüero, R., & Muñoz, L. (2014). Optimum multi-service access selection over heterogeneous wireless networks. *Transactions on Emerging Telecommunications Technologies*, 27(3), 328-338.

Hagos, D. (2015). The performance of network-controlled mobile data offloading from LTE to WiFi networks. *Telecommunication Systems*, *61*(4), 675-694. https://doi.org/10.1007/s11235-015-0061-2

Ismail, M., Abdrabou, A., & Zhuang, W. (2013). Cooperative decentralized resource allocation in heterogeneous wireless access medium. *IEEE Transactions on Wireless Communications*, *12*(2), 714-724. https://doi.org/10.1109/twc.2012.121112.120148

Lopez-Perez, D., Xiaoli Chu, Vasilakos, A., & Claussen, H. (2014). Power minimization based resource allocation for interference mitigation in OFDMA Femtocell networks. *IEEE Journal on Selected Areas in Communications*, *32*(2), 333-344. https://doi.org/10.1109/jsac.2014.141213

Panwar, R., & Mallick, B. (2015). A Comparative Study of Load Balancing Algorithms in Cloud Computing. *International Journal of Computer Applications*, *117*(24), 33-37. https://doi.org/10.5120/20890-3669

Saranya, D., & Maheswari, L. (2018). Implementation of Load Balancing Algorithms in Cloud Computing Environment using Cloud Analyst Simulator. *International Journal of Recent Trends in Engineering and Research*, *4*(7), 22-27. https://doi.org/10.23883/ijrter.2018.4351.o8zop

Serra, P., & Rodrigues, A. (2013). The Cost-Efficient Location Management in the LTE Picocell/Macrocell Network. *IEEE Communications Letters*, *17*(5), 904-907. https://doi.org/10.1109/lcomm.2013.040213.130048

Silva, K., Araújo, J., & Francês, C. (2016). A Performance Evaluation of WLAN-Femtocell-LTE beyond the Capacity Crunch. Does Femtocell have to overcome WLAN or can they coexist in HetNets? *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, *15*(4), 402-417. https://doi.org/10.1590/2179-10742016v15i4747

Supportcenter (2021). Retrieved on 18 May 2019, from https://community.riverbed.com/s/.

Temple, A. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting.

Yuan, Y. (2013). *LTE-Advanced Relay Technology and Standardization* (1st ed.). Springer.