VOL 4 (2020) NO 3

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Virtualized Fog Network with Load Balancing for IoT based Fog-to-Cloud

Istabraq M. Al-Joboury [#], Emad H. Al-Hemiary [#]

[#] *Al-Nahrain University, College of Information Engineering, Iraq*
E-mail: estabriq_94@coie-nahrain.edu.iq, emad@coie-nahrain.edu.iq

***Abstract*— Fog Computing is a new concept made by Cisco to provide same functionalities of Cloud Computing but near to Things to enhance performance such as reduce delay and response time. Packet loss may occur on single Fog server over a huge number of messages from Things because of several factors like limited bandwidth and capacity of queues in server. In this paper, Internet of Things based Fog-to-Cloud architecture is proposed to solve the problem of packet loss on Fog server using Load Balancing and virtualization. The architecture consists of 5 layers, namely: Things, gateway, Fog, Cloud, and application. Fog layer is virtualized to specified number of Fog servers using Graphical Network Simulator-3 and VirtualBox on local physical server. Server Load Balancing router is configured to distribute the huge traffic in Weighted Round Robin technique using Message Queue Telemetry Transport protocol. Then, maximum message from Fog layer are selected and sent to Cloud layer and the rest of messages are deleted within 1 hour using our proposed Data-in-Motion technique for storage, processing, and monitoring of messages. Thus, improving the performance of the Fog layer for storage and processing of messages, as well as reducing the packet loss to half and increasing throughput to 4 times than using single Fog server.**

***Keywords*— internet of things; cloud computing; fog computing; load balancing; data in motion; MQTT; packet loss; SLB.**

I. INTRODUCTION

Internet of Things (IoT) is the future of the Internet and most technologies for wireless telecommunication [1]. Things in IoT integrate and communicate with each other to transfer the useful information for intended persons. Things can be anything like pen, car, wristwatch, chair with smart sensors on them [2]. The number of Things grows rapidly; therefore, they will need IPv6 to get connected to the Internet. All application domains can be enabled by IoT to provide ubiquitous services and to improve societies and governments. The traditional architecture of IoT includes five layers: Things, gateways, middleware, application and business layer [3, 4]. Messages from Things such as sensors and actuators need to be stored in real time. Cloud Computing (CC) offers storage and processing in three different types: Software as a Service (SaaS) and Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) [5]. CC causes high response time and low security because it could be located in different region of Things [6]; therefore, Cisco has been proposed a new concept to handle a huge number of messages with less response time and higher security called Fog Computing (FC). FC provides storage, networking and processing as the same as CC; however, FC physically close to Things and users [7]. FC contains routers, switches, firewalls, Access Points (APs) and servers. It is recommended for delay-sensitive applications like healthcare and transportation [8]. Virtualization technology is the appropriate solution to achieve networking, computing and processing on multiple Virtual Machines (VMs) with a single physical server to reduce cost, power, complexity, and space. There are several organizations like Cisco, TELCO, ALTO, TCS, Avaya Networks, NICIRA, HP, F5, Nuage, Oracle Solaris and Cumulus Networks provide virtualization. It makes a middle layer known as Hypervisor between the physical and software layer to run multiple Operating Systems (OSs) simultaneously on single server such as Kernel Virtual Machine (KVM) [9], VirtualBox [10], Xen [11], and VMware [12].

Servers may fail to handle a huge number of messages from sensors and actuators in very fast and efficient which might lead to packet loss on application servers' due to limited bandwidth and capacity of queues. Therefore, multiple server arrangements with Load Balancing (LB) could solve the problem and share processing speed. LB is applied in some sites like Amazon which used High Availability Proxy (HAProxy) [13] with Network Address Translation (NAT) to employ the Least Connection (LC) technique. Other techniques like Nginx Plus API [14] and Elastic Beam [15] used layer 7 LB instead of layer 4 of Open Standard Interconnection (OSI). However, both are not free to use.

There are few works have been proposed the integration between Fog and Cloud or simply (F2C) as in [16], the authors introduce F2C architecture and its advantages with main challenges and implement IoT based F2C and IoT based Cloud architectures using Tareador simulation tool to provide the performance comparison between them in terms of execution time and speedup. The results show IoT based F2C is better than IoT based Cloud because it is reduced the execution time. However, the authors have not considered IoT protocols in the architecture and not implemented the proposed architectures practically. But, the authors in [17] implement F2C computing on real application for chronic obstructive pulmonary disease (COPD) patients and the results show that the F2C improves the quality of life of patients. None of two previous papers have considered the delay on their proposed, while the authors in [18], propose distributed service allocation strategy for both resource offering and service requirements based on F2C computing in order to reduce delay of service allocation and decrease traffic load on Cloud. Then, the same authors in [19] minimize the delay of the services and provide the capacity requirement. In addition, the queuing theory are considered in F2C concept as in [20], the optimal workload allocation is proposed based F2C architecture to reduce power consumption and delay. The results show that F2C is better performance that Cloud. We notice that the previous researchers have not been tested and implemented F2C with IoT protocols. Some authors try to propose an opposite path from Cloud to Fog (C2F) as in [21], the authors propose C2F architecture for monitoring healthcare network and smart homes. The results show C2F provides the better service to Things. Finally, the authors in [22] propose IoT architecture based on Cloud to combine Message Queue Telemetry Transport (MQTT) and Hypertext Transfer Protocol (HTTP) protocols and to distribute traffic among virtual servers using HAProxy. The performance evaluation of protocols is presented in terms of number of clients and Central Processing Unit (CPU) cores. The results show the MQTT protocol has better performance than HTTP. These authors have considered protocols, however the architecture is based on Cloud and not based on F2C. The Server Load Balancing (SLB) router is an IOS image of Cisco router. Inside SLB router, a virtual server is defined to represent a list of real servers called server farm. SLB router redirects messages from clients to this virtual server, and then the virtual server redirects messages to one of the servers in the server farm.SLB router is used in this paper IoT based Fog to distribute messages among specified number virtual Fog servers using Type 2. The aim of this paper is to propose IoT based F2C architecture for virtualizing Fog network with LB to mitigate the problem of packet loss and to increase throughput. Then, performance analysis is provided over a high volume of traffic. The proposed architecture uses MQTT protocol for communication between machines and end users.

The main contribution of this paper is to propose architecture named IoT F2C. The proposed architecture tries to reduce the packet loss on Fog layer using LB and virtualization and to reduce the high traffic on Cloud layer using the proposed Data-in-Motion (DM) technique with MQTT protocol. Up to our knowledge, this type of load balancer has not been used by researchers previously with IoT based F2C.

The rest of this paper is organized as follows: Section II and III provides an overview of MQTT protocol and SLB. Section IV, discusses the proposed IoT based F2C architecture and virtualized Fog network with LB. Finally, Section V and VI presents results and concludes this paper.

II. MQTT PROTOCOL

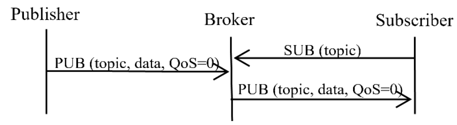
The MQTT protocol was created by Stanford-Clark and Nipper in 1999 [23]. Thereafter, MQTT was adopted by Advancing Open Standards for the Information Society (OASIS) in 2014 [24]. Then it standardized by ISO/IEC 20922 [25]. MQTT is customized for sensitive application like IoT and Machine to Machine (M2M). It works with Topic instead of using IP address and based on a Transmission Control Protocol (TCP). It used for applications and electronics with low power, bandwidth and cost [26]. There are two elements of MQTT: clients could be publisher, or subscriber to a certain topic, while the second type is a centralized server called broker. Subscribers do not need to know the IP addresses of publishers. Whilst, the broker must be configured with the IP and port of protocol [27]. The process of the protocol starts when client chooses to be either publisher or subscriber so that the broker forwards messages to subscribers [28].

The broker sets up a connection between client and broker with unlimited number of clients, then nominates the subscribed clients to specific topics to forward messages to them. The process works as the same hub/spoke model. Topics are organized as topic/subtopic/value like "homecare/heartbeat/100". Clients can subscribe to multiple topics at the same time. For protection purposes, the broker employs Secure Sockets Layer (SSL) / Transport Layer Security (TLS) encryption methods. These encryptions are enabled between broker and clients as in HTTP encryption methods. broker requests username and password for each client [29, 30]. MQTT has a synchronous communication, a lower overhead and three levels of quality of services (QoS) as shown in Figure 1, the QoS for a delivery assurance between clients and broker are [31]:

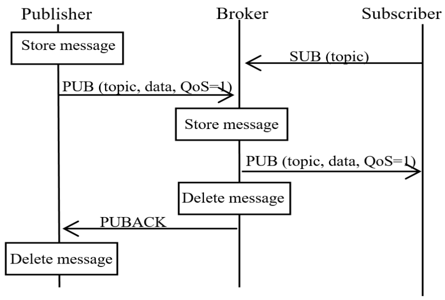
1) *QoS level 0*: This level does not deal with acknowledgement, in which the clients publish messages to clients subscribed to the same topic through a broker. Message is received at most once and does not know if the message is delivered or not. Therefore, loss may occur in this situation. Level 0 also known as fire and forget.

2) *QoS level 1:* This level sends acknowledgement every time clients publish messages. The message is received at least once. In this level, if data gets lost the broker retransmits data to the publisher.

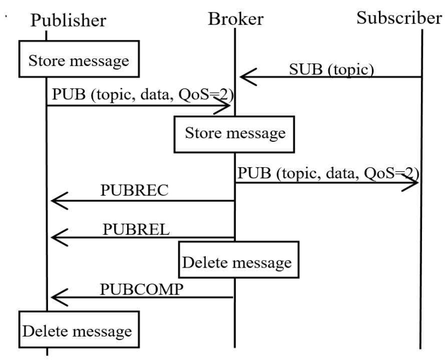
3) *QoS level 2:* This level requires four-way handshake to deliver data at exactly once and this may increase the overhead.



(a) QoS level 0



(b) QoS level 1



(c) QoS level 2

Fig. 1. The operation of MQTT: (a) with QoS level 0 (b) with QoS level 1 (c) with QoS level 2

III. SLB

SLB is a Cisco router and acts as router and load balancer in the same IOS. It is used to distribute requests from clients to list of number of servers called server farm. SLB router has a specific number of interfaces that can be assigned with IP address. Also, it contains virtual load balancer inside the router that can be assigned with virtual IP address. Clients use this virtual IP address as the destination field. SLB has two types of techniques to distribute messages, namely: Weighted Round Robin (WRR) and Weighted Least Connection (WLC) to provide high availability. Weighted on these techniques is configured based on server capabilities in server farm [33]. SLB reduces the number of required hardware, space, cost and energy because holds the functions of LB and routing in the same device. SLB facilitates the maintenance of configuration when new servers added or old servers removed and this process can be done without affecting any problems in configuration [34]. The virtual load balancer can be divided into two types, namely: Directed means the virtual load balancer can be configured in any range of IP address. This type need to configure NAT protocol between the virtual load balancer and server farm in order to translate the IP address of that load balancer to server according to technique used. The second type is dispatched indicates the IP address of virtual load balancer is configure with the same range of server farm; however, this type is not applicable with multiple routers [35, 36].

IV. IoT Based F2C Architecture

The proposed architecture IoT based F2C with virtualized Fog and LB consists of five layers over two sites: site 'A' (at Al-Nahrain University, College of Information Engineering), and site 'B' at Ministry of Higher Education and Scientific Research, Department of Research and Development (MoHESR/ RRD) and is discussed as follows:

A. Thinks layer

This layer consists of Things located at site 'A' with features of low cost, power and bandwidth such as sensors, actuators and microcontrollers. In this paper, one real pulse rate sensor [37] and thousands of virtual sensors are used to sense from patient's body. These virtual sensors are generated using Tsung tool [38]. Tsung is installed on Personal Computer (PC) with characteristics: Ubuntu 14.04.5 LTS OS, Memory: 3.7 GB, processor: Intel(R) Core (TM) i3-380 CPU @ 2.53GHz *4, disk: 488.1 GB. NodeMCU [39] with WiFi built-in is used to collect messages from Things and transmit it to the up layers. Sensors are programed using C/C++ and XML programing languages for real sensor by Arduino Integrated Development Environment (IDE) version 1.6.12 and virtual sensors respectively. Things are configured with MQTT version 3.1.1 protocol QoS level 0 and 1 to communicate with other layers.

B. Gateway layer

This layer consists of IEEE802.11n Mikrotik AP (RB2011UAS-ZHND-IN), Cisco switch (Catalyst 2900G Series) and Cisco router (Catalyst 2600G Series) located at site 'A'. AP is used to transmit messages from Things layer to Fog layer to be processed and stored. While, Cisco switch is used to connect different devices together. Opent Shortest Path (OSPF) [40] is configured in Cisco router to forward messages to the Internet.

C. Fog layer

Fog server receives messages from Things and stores it temporarily in MySQL database for specific time (for example 1 hour) and located at site 'A'. Fog server is HP ProLiant 380 G7 16 Core 32 based Ubuntu server 14.04 LTS with 32 GB of dynamic memory and 500 GB of permanent storage. Middleware script is adding to Fog layer to subscribe messages using Python Application Programming Interface (API) with help of PHP-Mosquitto broker. This layer has two scenarios:

1) Single Server: This scenario consists of single server based Linux where all messages from Things are processed by this server as shown in Figure 2; however, packet loss may occur in single server because of a large number of messages.

2) Multiple Servers: This scenario consists of specific number of servers. Fog servers is virtualized using VirtualBox that is connected to Graphical Network Simulator-3 (GNS3). SLB router (c3640-jk9s-mz.124-16.bin) is configured inside GNS3 and is responsible for performing

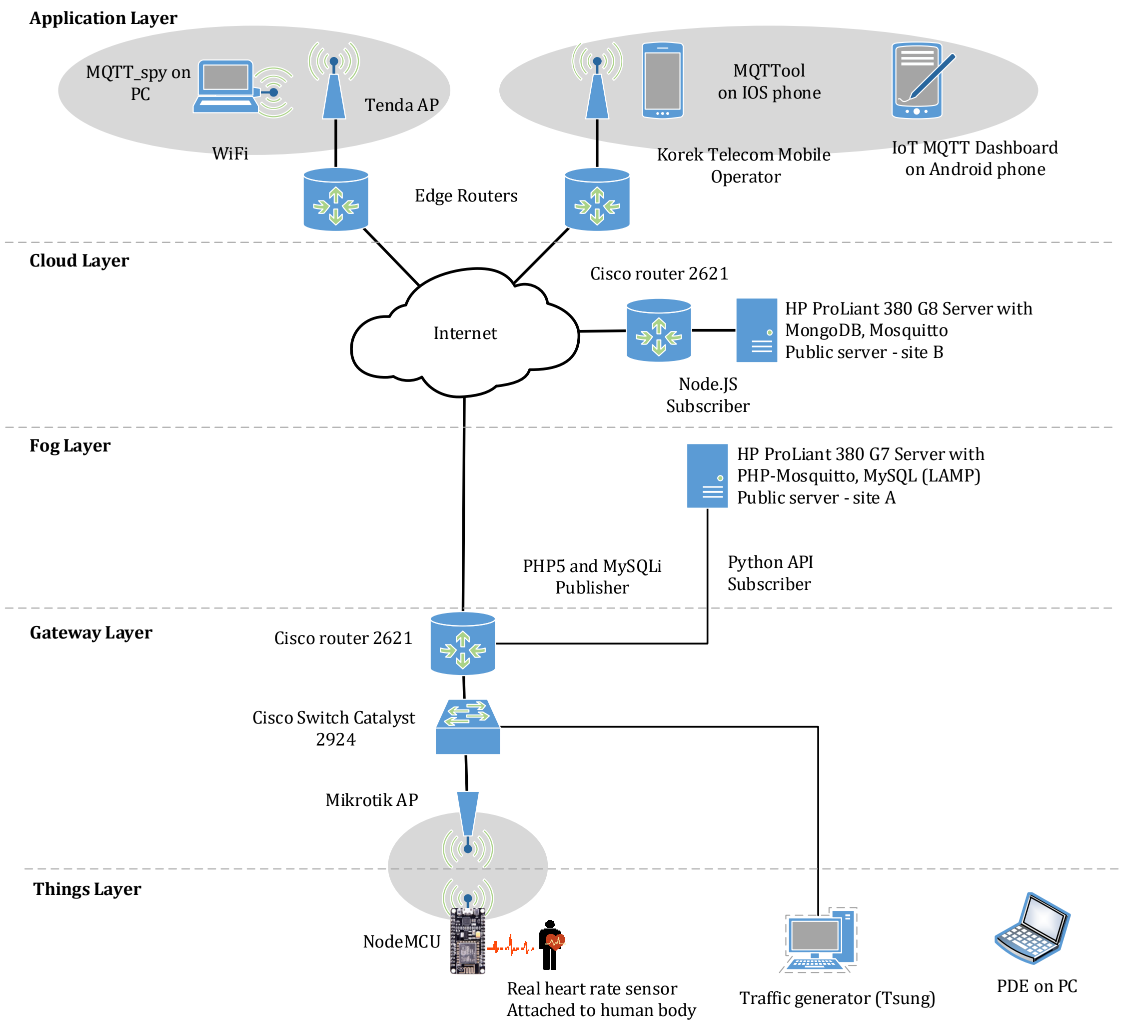


Fig. 2. IoT based F2C architecture with single server

LB WRR technique where messages are distributed on multiple virtual servers. Due to a large number of VMs, only one VM is installed with OS and configured with MySQL database, Mosquitto and Network configuration. Then, this one is cloned to 15 virtual Fog servers. SLB is configured with OSPF routing protocols to forward messages from Fog layer to the Internet. Cisco switch is presented in GNS3 to mediate and combine all virtualized Fog servers, Internet, and SLB and is connected to the real hardware of proposed IoT architecture as shown in Figure 3 and 4. It can notice that this scenario excludes the Cisco router (Catalyst 2600G Series) from gateway layer because SLB has the functionalities of router, thus it reduces number of required hardware.

After each scenario, maximum value from MySQL database in each server is selected and publish it every 1 hour to Cloud layer using PHP5 and MySQLi programming language. The proposed technique of combination of Python API with PHP5 and MySQLi scripts is named DM.

D. Cloud layer

Cloud server receives the selected messages from each scenario every 1 hour and stores it permanently in MongoDB using Node.js with the help of Mosquitto broker. Messages in Cloud are formatted in Java Script Object Notation (JSON). Cloud server is HP ProLiant 380 G8 16 Core based Ubuntu server 14.04 LTS with 32 GB of dynamic memory and 500 GB of permanent storage and layer located at site 'B'.

E. Application layer

Physicians and patients' family can monitor messages directly from Things using Processing Development Environments (PDE) version 3.2.1. Also, messages can be monitored from Fog and Cloud layer using MQTTool, MQTT Dashboard Tool, and Mqtt Spy by smart phones or PC.

V. RESULTS

Results are discussed with explanation in this section. The performance of throughput and packet loss are measured; both depends on bandwidth; therefore, link bandwidth are also measured. Internet Performance Working Group (iperf) tool is used to compute bandwidth of link. This tool is used with TCP/UDP and based on client/server model.

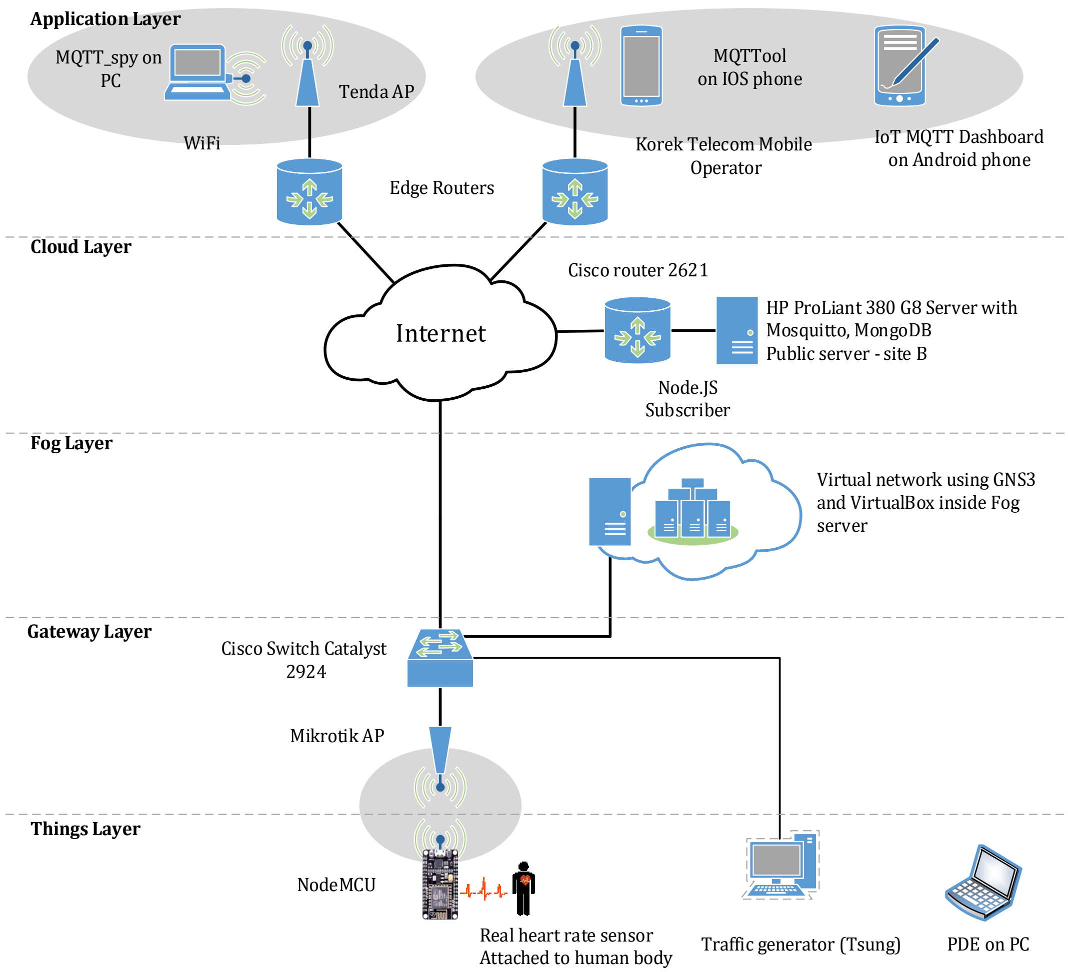


Fig. 3. IoT based F2C architecture with multiple server

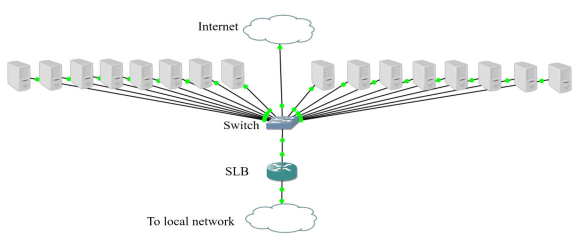


Fig. 4. Virtual network using GNS3 and VirtualBox inside Fog server

In this paper, IPerf is used with TCP because MQTT protocols based on TCP. The CLI used in the server side is:

iperf –s

where –s appends the host in server side. While, the CLI used in the client side is:

iperf –c x.x.x.x

Where –c means the host in client side and the x.x.x.x is the IP address of server side. We measure bandwidth between the Things layer located at site 'A' and Cloud layer located at site 'B' during the running test of the Tsung tool. The average bandwidth can be shown in Figure 5.

The Encapsulation of MQTT message transmitted from a client to a server is formed normally by adding a specified number of bytes in each layer to create the header as shown in Figure 6. The frame length becomes 77 bytes from an original of 9 bytes which represents the message coming from application layer. The other 68 bytes represent the total overhead of encapsulation. These measurements are computed using Wireshark [41]. MQTT frame is important to throughput and packet loss measurements. Throughput can be defined as the number of sucesscfful packets per unit time. Figure 7 shows the average throughput of Fog layer in each scrnario and is computed using Tsung tool and with the following equation [42]:

$$Throughput = \frac{\text{Size of successful messages in bits}}{\text{Transmission time in seconds}} \quad (1)$$

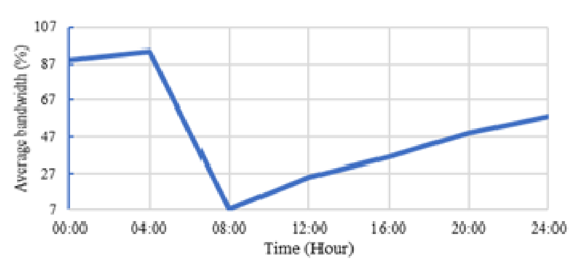


Fig. 5. Average bandwidth between Things and Cloud layer at site 'B'

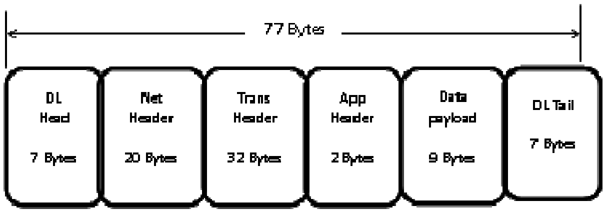


Fig. 6. MQTT Frame structure (DL: Datalink, Net: Network, Trans: Transport, App: Application)

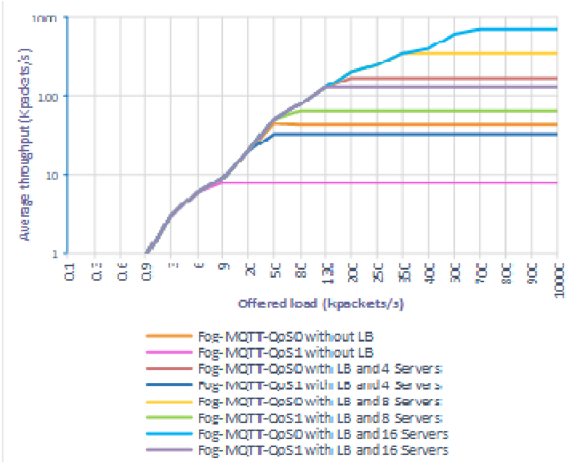


Fig. 7. Average throughput of proposed IoT based F2C architectures with LB

The results show that average throughput of MQTT-QoS 0 with LB and 4 servers is 4 times higher than MQTT-QoS 0 without LB, MQTT-QoS 0 with LB and 8 servers is 8 times higher than MQTT-QoS 0 without LB, and MQTT-QoS 0 with LB and 16 servers is 16 times higher than MQTT-QoS 0 without LB. Throughput depends on bandwidth and the queue of servers. It can be notice from results, there is a relationship between throughput and number of servers with LB. Throughput increase n times where n equal to number of servers.

Packet loss defined as number of packets of data fail to reach the final destination when they travel through network. Figure 8 shows the average packet loss of Fog layer in each scenario and is computed using Tsung tool and with the following equation [43]:

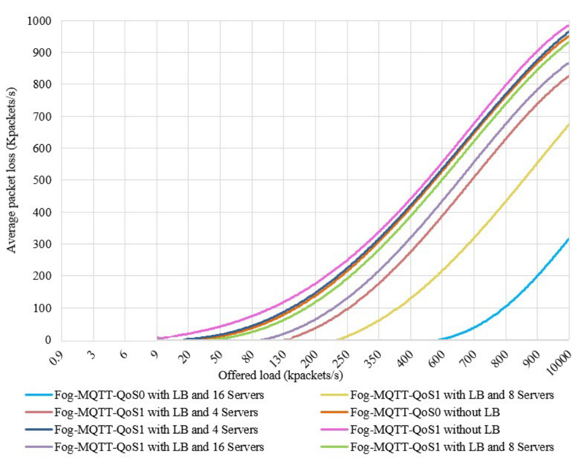


Fig. 8. Average packet loss of proposed IoT based F2C architectures with LB

As the figure show, the use of LB increases Fog network throughput and reduces packet loss to its minimum value (2 times and 1 time in QoS 0 and QoS 1 respectively). This result comes from the fact that using LB will distribute the traffic over four virtual Fog servers, thus all messages arrive safe and sound.

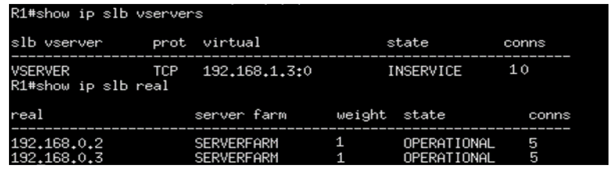


Fig. 9. Monitoring of SLBs

In Figure 9, the result from SLB router in GNS3 shows that the IP address of SLB and number of connections. In this test only two virtual Fog servers are running and one sensor. At the one moment, seven messages are received in SLB. Both servers have equal weight so that they can receive messages equally according to WRR fashion.

VI. Conclusions

IoT based F2C architecture is proposed to enhance the performance in terms of throughput and packet loss. The proposed architecture is implemented practically over two different sites in five layers: low cost and power Things, gateway, Fog, Cloud, and application. Fog layer is suggested in this architecture to improve performance such as reduce delay, however packet loss for high traffic may occur which impact on critical real-time applications like healthcare. Performance analysis of two scenarios is provided which the first scenario suggests using single Fog server, while the second scenario mitigates packet loss by employing LB and virtualization over 16 virtual servers. SLB router is used for distributing the huge volume of traffic from Things to according to WRR fashion. Virtualization technology is used to reduce cost and power using VirtualBox Type 2. GNS3 tool is used for creating a virtual network topology with SLB, switch and virtual servers that are created in VirtualBox. The emulated network is connected to the Internet to create a real IoT network. The results show that the second scenario reduces packet loss to half and increases throughput than the first scenario because the arrived messages are distributed among a specific number of servers. Finally, the connection between Fog and Cloud are provided using proposed DM technique. The latter proves our reason for choosing LB on the Fog layer instead of the Cloud because the Fog lies in the middle of Things and Cloud, thus any loss in the Fog will also result in loss in the Cloud

References

[1] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” Future Generation Computer Systems, vol. 78, pp. 641–658, 2018. DOI: 10.1016/j.future.2017.02.014.

[2] K. H. Rahouma, R. H. M. Aly, and H. F. Hamed, “Challenges and Solutions of Using the Social Internet of Things in Healthcare and Medical Solutions—A Survey,” Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications

Studies in Computational Intelligence, pp. 13–30, 2019. DOI:10.1007/978-3-030-24513-9_2.
[3] N. Zou, S. Liang, and D. He, “Issues and challenges of user and data interaction in healthcare-related IoT,” Library Hi Tech, vol. ahead-of-print, no. ahead-of-print, 2020. DOI: 10.1108/lht-09-2019-0177.
[4] I. M. Al-Joboury and E. H. Al-Hemiary, “IoT Protocols Based Fog/Cloud over High Traffic," The ISC Int'l Journal of Information Security, vol. 11, no. 3, pp. 173–180, 2019. DOI: 10.22042/ISECURE.2019.11.3.23.
[5] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, “HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments,” Future Generation Computer Systems, vol. 104, pp. 187–200, 2020. DOI: 10.1016/j.future.2019.10.043.
[6] T. S. Gunawan, M. H. H. Gani, F. D. A. Rahman, and M. Kartiwi, “Development of Face Recognition on Raspberry Pi for Security Enhancement of Smart Home System,” Indonesian Journal of Electrical Engineering and Informatics (IJEEI), vol. 5, no. 4, Jan. 2017. DOI: 10.11591/ijeei.v5i4.361.
[7] E. M. Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G. Ren, J. Zhu, and J. Farre, "What is a Fog Node A Tutorial on Current Concepts towards a Common Definition". arXiv preprint arXiv:1611.09193, 2016.
[8] G. Javadzadeh and A. M. Rahmani, “Fog Computing Applications in Smart Cities: A Systematic Survey,” Wireless Networks, vol. 26, no. 2, pp. 1433–1457, Dec. 2019. DOI: 10.1007/s11276-019-02208-y.
[9] Kernel-Based Virtual Machine (KVM), https://www.linux-kvm.org, [Accessed: 15-Nov-2019].
[10] Oracle VM VirtualBox, https://www.virtualbox.org, [Accessed: 15-Nov-2019].
[11] The Xen Project (Xen), https://www.xenproject.org, [Accessed: 15-Nov-2019].
[12] Virtual Machine Software (VMWare), http://www.vmware.com, [Accessed: 15-Nov-2019].
[13] “HAProxy,” powered by HAPROXY. [Online]. Available: https://www.haproxy.org/. [Accessed: 15-Nov-2019].
[14] “MQTT Load Balancing and Session Persistence with NGINX Plus,” NGINX. [Online]. Available: https://www.nginx.com/blog/nginx-plus-iot-load-balancing-mqtt/. [Accessed: 15-July-2019].
[15] “Scalable and Secure MQTT Load Balancing with Elastic Beam and HiveMQ,” HiveMQ, 12-Sep-2016. [Online]. Available: http://www.hivemq.com/blog/scalable-and-secure-mqtt-load-balancing-with-elastic-beam-and-hivemq/. [Accessed: 15-Nov-2019].
[16] A. Al-Qerem, M. Alauthman, A. Almomani, and B. B. Gupta, “IoT transaction processing through cooperative concurrency control on fog–cloud computing environment,” Soft Computing, vol. 24, no. 8, pp. 5695–5711, 2019. DOI: 10.1007/s00500-019-04220-y.
[17] X. Masip-Bruin, E. Marin-Tordera, A. Alonso, and J. Garcia, “Fog-to-cloud Computing (F2C): The key technology enabler for dependable e-health services deployment,” 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2016. DOI: 10.1109/MedHocNet.2016.7528425.
[18] V. B. Souza, X. Masip-Bruin, E. Marin-Tordera, W. Ramirez, and S. Sanchez, “Towards Distributed Service Allocation in Fog-to-Cloud (F2C) Scenarios,” 2016 IEEE Global Communications Conference (GLOBECOM), 2016. DOI: 10.1109/GLOCOM.2016.7842341.
[19] V. B. C. Souza, W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, G. Ren, and G. Tashakor, “Handling service allocation in combined Fog-cloud scenarios,” 2016 IEEE International Conference on Communications (ICC), 2016. DOI: 10.1109/ICC.2016.7511465.
[20] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, “Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption,” IEEE Internet of Things Journal, pp. 1–1, 2016. DOI: 10.1109/JIOT.2016.2565516.
[21] C. S. Nandyala and H.-K. Kim, “From Cloud to Fog and IoT-Based Real-Time U-Healthcare Monitoring for Smart Homes and Hospitals,” International Journal of Smart Home, vol. 10, no. 2, pp. 187–196, 2016. DOI:10.14257/ijsh.2016.10.2.18.
[22] R. Cao, Z. Tang, C. Liu, and B. Veeravalli, “A Scalable Multi-cloud Storage Architecture for Cloud-Supported Medical Internet of Things,” IEEE Internet of Things Journal, pp. 1–1, 2019. DOI: 10.1109/JIOT.2019.2946296.
[23] A. Malik and H. Om, “Cloud Computing and Internet of Things Integration: Architecture, Applications, Issues, and Challenges,” Sustainable Cloud and Energy Services, pp. 1–24, 2017. DOI: 10.1007/978-3-319-62238-5_1.
[24] "MQTT," 2014. [Online]. Available: http://mqtt.org/. [Accessed: 15-Nov-2019].
[25] A. Banks and R. Gupta. "MQTT Version 3.1. 1." OASIS standard, 2014.
[26] “ISO - International Organization for Standardization,” ISO/IEC 20922:2016 - Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1, 08-Jun-2016. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=69466. [Accessed: 15-Nov-2019].
[27] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, “Which IoT Protocol? Comparing Standardized Approaches over a Common M2M Application,” 2016 IEEE Global Communications Conference (GLOBECOM), 2016. DOI:10.1109/glocom.2016.7842383.
[28] M. A. Triawan, H. Hindersah, D. Yolanda, and F. Hadiatna, “Internet of things using publish and subscribe method cloud-based application to NFT-based hydroponic system,” 2016 6th International Conference on System Engineering and Technology (ICSET), 2016. DOI: 10.1109/ICSEngT.2016.7849631.
[29] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, “Performance evaluation of MQTT and CoAP via a common middleware,” 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014. DOI: 10.1109/ISSNIP.2014.6827678.
[30] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” Journal of Electrical and Computer Engineering, vol. 2017, pp. 1–25, 2017. DOI: 10.1155/2017/9324035
[31] K. Grgic, I. Speh, and I. Hedi, “A web-based IoT solution for monitoring data using MQTT protocol,” 2016 International Conference on Smart Systems and Technologies (SST), 2016. DOI: 10.1109/SST.2016.7765668.
[32] I. Al-Joboury and E. Al-Hemiary, “IoT-F2CDM-LB: IoT Based Fog-to-Cloud and Data-in-Motion Architectures with Load Balancing,” EAI Endorsed Transactions on Internet of Things, vol. 4, no. 13, p. 155332, Nov. 2018. DOI:10.4108/eai.6-4-2018.155332.
[33] J. Zinke and B. Schnor, "The impact of weights on the performance of Server Load Balancing systems," 2013 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2013.
[34] R. R. Adiputra, S. Hadiyoso, and Y. S. Hariyani, “Internet of Things: Low Cost and Wearable SpO2 Device for Health Monitoring,” International Journal of Electrical and Computer Engineering (IJECE), vol. 8, no. 2, p. 939, Jan. 2018. DOI:10.11591/ijece.v8i2.pp939-945.
[35] J. Tiso, Designing Cisco network service architectures (ARCH): Foundation learning guide. Indianapolis, IN: Cisco Press, 2016.
[36] L. Hou, S. Zhao, X. Xiong, K. Zheng, P. Chatzimisios, M. S. Hossain, and W. Xiang, “Internet of Things Cloud: Architecture and Implementation,” IEEE Communications Magazine, vol. 54, no. 12, pp. 32–39, 2016. DOI: 10.1109/MCOM.2016.1600398CM.
[37] Pulse sensor, https://pulsesensor.com/. [Accessed: 15-Nov-2019].
[38] Tsung, http://tsung.erlang-projects.org/. [Accessed: 15-Nov-2019].
[39] NodeMCU, http://nodemcu.com/index_en.html/. [Accessed: 15-Nov-2019].
[40] G. K. Dey, M. M. Ahmed, and K. T. Ahmmed, “Performance analysis and redistribution among RIPv2, EIGRP & OSPF Routing Protocol,” 2015 International Conference on Computer and Information Engineering (ICCIE), 2015. DOI: 10.1109/ccie.2015.7399308.
[41] Wireshark. https://www.wireshark.org/. [Accessed: 15-Nov-2019].
[42] M. Mamunur and P. Datta, “Performance Analysis of Vehicular Ad Hoc Network (VANET) Considering Different Scenarios of a City,” International Journal of Computer Applications, vol. 162, no. 10, pp. 1–7, 2017. DOI: 10.5120/ijca2017913329.
[43] K. Suresh and R. J. Kannan, “Review of Advancements in Multi-tenant Framework in Cloud Computing,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 11, no. 3, p. 1102, Jan. 2018. DOI: 10.11591/ijeecs.v11.i3.pp1102-1108