International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Performance Evaluation of Wireless Router Testbed using Raspberry Pi

F S Samidi, M H M Haqeem, N A M Radzi, F Abdullah, A Ismail, M Z Jamaludin, Y E Jalil, W S H M W Ahmad

Abstract: This paper proposed a programmable wireless router testbed designed using a microcontroller, Raspberry Pi. The proposed router provides a solution to the limitation of off-the-shelf wireless router on its ability to be configured and embedded with new algorithms. The proposed router testbed is designed to have one Header Pi and three Forwarding Pis which act to inject label and forward packet respectively. The evaluation of results is validated with off-the-shelf router. From the setup, it is proven that Raspberry Pi is suitable to be used as a testbed for academic experiment as it has the ability to be reprogrammed. In addition, it has a compact design which eases for laboratories use. Based on the performance evaluation results, both proposed Raspberry Pi router testbed and off-the-shelf router are in line with each other despite the expected differences in terms of throughput, delay and jitter performance. This proves that the designed programmable Raspberry Pi router testbed can be utilized for small-scale hardware experiment purposes rather than software simulation.

Keywords: Algorithm, Raspberry Pi and Wireless router

Manuscript published on November 30, 2019.

* Correspondence Author

F S Samidi, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

M H M Haqeem, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

N A M Radzi, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

F Abdullah, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

A Ismail, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

M Z Jamaludin, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

Y E Jalil, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

W S H M W Ahmad, Institute of Informatics and Computing in Energy (IICE), Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43000 Kajang, Selangor, Malaysia.

I. INTRODUCTION

Having a wireless router is now a need in order to have access to the network without labyrinth of wires connected around the workplaces and houses. Depending on the manufacturer and model, it is able to function in a wired Local Area Network (LAN), in a wireless-only LAN, or in a mixed wired and wireless network. Other benefits of using a wireless router are that they allow the network to have a dynamic formation, cost-efficient and easy installation to the user [1].

On the other hand, in the researcher's point of view, off-the-shelf wireless router has a limitation on its ability to configure the settings and algorithms. These wireless routers only allow minor changes such as changing channels, setting the remote access point Media Access Control (MAC) address, and selecting router mode to be done. Hence, it does not allow the researches to apply their work or algorithms for testing. In order to reconfigure the router's algorithms, manufacture permission is needed. Unfortunately, most commercial router by merchants such as Cisco Frameworks, Inc. and Juniper Systems work in a closed improvement design [2].

Therefore, in order to solve this problem, we are taking the advantages of Raspberry Pi, a programmable computer board embedded with Linux powered Central Processing Unit (CPU) to develop a small scale reprogrammable wireless router testbed for academic usage. Raspberry Pi is chosen as it is powerful and consumed small power as stated in [3] and has been implemented in numerous projects and research such [4]–[9]. The programmable routers testbed will be able to support static routing and connect to the other router and client wirelessly via the access point or any Wi-Fi network. The performance of the testbed will be compared and benchmarked with an off-the-shelf wireless router. The first objective of this paper is to implement Raspberry Pi as the main lab-scale programmable router element which can be utilized for academic and research use. Secondly, to evaluate the performance of the testbed in sending data from server to client in terms of throughput, delay, and jitter.

II. RELATED WORKS

This section will describe the study on capabilities of Raspberry Pi to be implemented in various aspects of networking research and various past approaches to a reconfigurable router from other researchers. The conclusion of the related works is tabulated at the end of this section.

Nikhade et al. [4] proposed a Wireless Sensor Network (WSN) system by combining Raspberry Pi and Zig Bee as a monitoring application for the environment. They have studied the ability of Raspberry Pi as base station for all the sensors connections using Zig Bee communication protocol. The Raspberry Pi is also configured as the server and data gateway for WSN, database server, and web server. Other than that, Nikhade et al. also designed sensor nodes to be connected to the Raspberry Pi which utilizes ATMEGA324PA AVR Core Microcontroller. This design allows them to have multiple sensors on the sensor node board. Their approach has proven to simplify the complexity and reduce the cost of the development. However, the design implementation is not suitable for sending large file because bandwidth will be limited to ZigBee specification.

There is also different research by Zhao et al. [5], the Raspberry Pi is utilized in Internet of Things (IoT) application. The aim is to reduce the usage of cables by changing the low priority data cable to IoT implementation. In their research, the authors set up the Raspberry Pi as a server that has the ability to store and fetch the required data and connected it to several clients. Three different client-server communication environments are created, using Wi-Fi, ZigBee and local host file sharing for experimental purposes. The use of ZigBee as the communication platform is very useful in data security as it uses 128-bit symmetric encryption keys to ensure secure transmission. Despite that, ZigBee still needs support from other end devices to operate and as it doesn't have its own platform to operate as end devices.

Other than that, there is also research that implements open source software to define their network topology such as Gupta et al. [10]. They proposed a switch which is designed using Raspberry Pi integrated with OpenFlow, a Software-Defined Networking (SDN) software. In this research, they created a small, portable SDN Raspberry Pi based switch for testing their SDN application. The Raspberry Pi was installed with Ubuntu MATE Linux that support OpenFlow Specification 1.0-1.4. The setup required one client, a POX/RYU SDN controller and two servers connected to the Raspberry Pi, therefore they had to attach a Universal Serial Bus (USB) LAN card as the extension for the single Ethernet port that is available on the Raspberry Pi. Nonetheless, the disadvantage of using SDN is proven in [11], as a major network failure will happen if the central controller is damaged.

There is also a research in designing a new framework such as Xu et al. [2] where the paper introduced an open, adaptable, and modularized reconfigurable router. Their programmable router is the combination of a component development environment, included with code editor, a multiple platform compiler, and automated testing tool. The reconfigurable platform for the router is created with 5 layers of routing platform: Hardware, Virtual Operating System (VOS), Component Control, Component/Meta-component Library, and Routing service layer and most of the libraries are an open source component which gives advantages to routing capabilities improvement [2]. However, the routers implemented are being on a commercial scale which still does not solve the objective of having a lab-scale programmable router.

Another approach from Porch et al. [12] whereby the Banana Pi Router is used to perform Intelligent Centric Networks (ICN) which enables intelligence to the network providing effective communication in an inherent network. In order to simulate a complete ICN-focused testbed, they chose the circumstance of the pull-based flexible video transport. They monitored the execution of various client-based adaption at the application level and data sending strategies in the network. The testbed comprises of two devoted systems which interface the Banana Pis in a star-like topology. The supposed Management Network (MN), demonstrated is utilized for configuration and observation. Porch et al. also created a physically isolated management, which is meant as the Emulation Network (EN). The EN is acknowledged as a virtual system overlay that is made utilizing the normal systems administration devices, for example, iptables and traffic control. The reason of choosing Banana Pi Router is that it is one of the single-board PCs that gives more than one Ethernet interface and also it keeps up a SATA 2.0 interface, which permits to include a Solid State Drive (SSD) for further upgrade. Unfortunately, the paper also stated that the usage of Banana Pi causes serious problems when handling a large amount of network traffics and have large delay even for a simple topology.

Based on those previous works, Raspberry Pi is chosen as for the programmable wireless router testbed. Compared to computer board router such Banana Pi which also have a compact design which is suitable for lab scale usage, the Raspberry Pi has more features that allow it to be scalable and suitable for experimental purposes. Written by Vujovic et al. [5] Raspberry Pi is a small computer-based board that is low cost, powerful, and have the ability to be programmable which is suitable for the our programmable router testbed compared to Banana Pi router. This board is embedded with processor, graphical chips, memory and can be adapted to various external interfaces or devices in order to maximize the user as user's need. Raspberry Pi's functions are the same as a standard PC which required a console for command, an LCD display unit, and a power supply. SD memory card typically utilized in such an approach to resemble a hard drive to Raspberry Pi's processor. The Pi is powered by using a small scale USB connector and Web availability can be provided via Ethernet/LAN link or through Wi-Fi connectivity through a USB dongle. The important thing to note that, Raspberry Pi utilizes a working LINUX framework called Raspbian but not limited to that, there are likewise a couple of non-Linux OS alternatives that can also be embedded to the Raspberry Pi based on user needs [13].

However, the Raspberry Pi has several limitations as stated by Vujovic et al. [5]. Firstly, it does not have an ongoing clock (RTC) when powered with an inbuilt battery. The missing clock can be a problem when utilizing a system as a server when the system needs a data time stamp for recording purposes. Secondly, the Raspberry Pi dependably boots from a Secure Digital (SD) card. Hence, if an Operating System (OS) is installed in a Universal Serial Bus (USB) stick or an outside hard drive, it will fail to boot [5].

Not only that, the performance of the router testbed also may drop when handling heavy traffic in prolonged time [14]. Other than that, the ability to code in either Python, C or C++ adds another complexity in designing reprogrammable router using Raspberry Pi compared to other router such Banana Pi Router. Fortunately, the limitations of using Raspberry Pi in our case will not affect the testbed performance as the Raspberry Pi will not store any data and only forward data packets between devices in the network. The component integration also will be simple as Raspberry Pi has many preinstalled connection and expansion ports. The summary of overall related work is summarized in Table 1.

Table. 1 The summary on the previous application of Raspberry Pi and Wireless Router testbed.

Author (Year)	Related Work	Methodology	Advantages	Disadvantages
S.G. Nikhade et.al (2015)) [4]	Wireless Sensor Network System Using Raspberry Pi and ZigBee for Environmental Monitoring Application	- Combined gateway nodes of the wireless sensor network, database server and web server in the Raspberry Pi - Act as a base station for sensor nodes using communication protocol and client using external network	- Integration of gateway node, database and web server in Raspberry Pi - compact, scalable	- A small defined rate of 250 Kbits/s
C.W. Zhao et.al (2015)) [15]	Exploring IOT Application using Raspberry Pi	- Raspberry Pi is used as a server by connecting multiple clients in the same network - The study was made in 3 situations where the communication is between client-server using Wi-Fi, ZigBee, and local host	- Secured transmission using ZigBee – use 128-bit symmetric encryption keys	- Not suitable for sending a large file - Bandwidth limited to ZigBee specification
V.Gupta et.al (2018) [10]	Developed SDN switch using Raspberry Pi.	- Made using preloaded Raspbian OS - Ethernet port is increased using a USB LAN card.	- Raspberry Pi is cost-efficient and easy to use.	- Major network failure if the central controller damaged
X.Ke et.al (2014) [2]	Toward a Practical Reconfigurable Router: A Software Component Development Approach	- Created with 5 layers of routing platform: Hardware, VOS, and Component Control, Component/Meta C component Library, and Routing service layer. - Open source component for routing capabilities improvement	- Dynamically assembled, replaced, and updated in the form of components	- Only adopted in commercial environment - Need more component for a faster development cycle.
D.Posch et.al (2016) [12]	Emulating NDN-based Multimedia Delivery	- Using Banana Pi router to develop NDN - Have Management Network for monitoring and configuration -Embedded with Emulation Network for IP tables and traffic control	- Cost-effective and easy to use	- Banana Pi causes serious problems for a large amount of network traffics -Big delay for a simple topology

V.Vujociv et.al (2014) [5]	Raspberry Pi as a Wireless Sensor Node: Performance and Constraints	- Review about Raspberry Pi performance	- Runs on Linux OS and reprogrammable - Possible to expand Raspberry PI with Wi-Fi and Bluetooth adapters - Expandable system	- Does not have real-time clock (RTC) for server use - Does not support out of the box Wi-Fi modules and but need to be added by USB dongles.

III. METHODOLOGY

In this project, we propose a programmable router testbed by using Raspberry Pi. It has the ability to be reprogram and suitable for academic use. Its small size also making it easy to be utilized as a testbed in a constrained space in a lab. Therefore, Raspberry Pi is the most reasonable decision in this testbed.

Proposed Router Testbed Configuration and Setup

For the hardware setup, each router consists of four units of Raspberry Pi which are one Header Pi and three Forwarding Pi as Figure 1. Each Raspberry Pi will be equipped with LCD display and USB power cord. All four of the Raspberry Pi will be linked to each other and it is important to note that, the Raspberry Pi has only a single Ethernet port, therefore, the communication between Header Pi and all Forwarding Pi are improvised using Ethernet switch. More switches are attached to the hardware setup to allow multiple routers and Forwarding Pi to be interconnected with each other. In order to create a wireless environment for the router testbed, an access point is included in the hardware as an additional component to the router. The access point is configured to be in Bridge mode so that it can communicate directly with its pair.

When a connection is established between the router and the network elements for data transmission, the Header Pi will function to inject a string of label onto the data information before the labeled data is broadcasted to all the Forwarding Pi as shown in Figure 2. At that time, the Forwarding Pi will compare the received labeled data with its own predefined label and determine the next action as Figure 3. If the label is same, the Forwarding Pi will keep on forwarding the data to the desired destination. Otherwise, the data will be set to 0 and discarded.

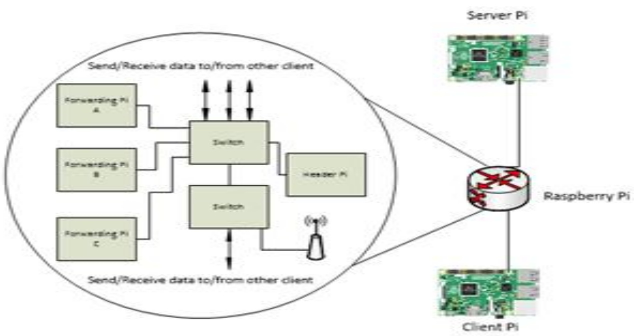

Fig. 1 The configuration of the Raspberry Pi Router Testbed

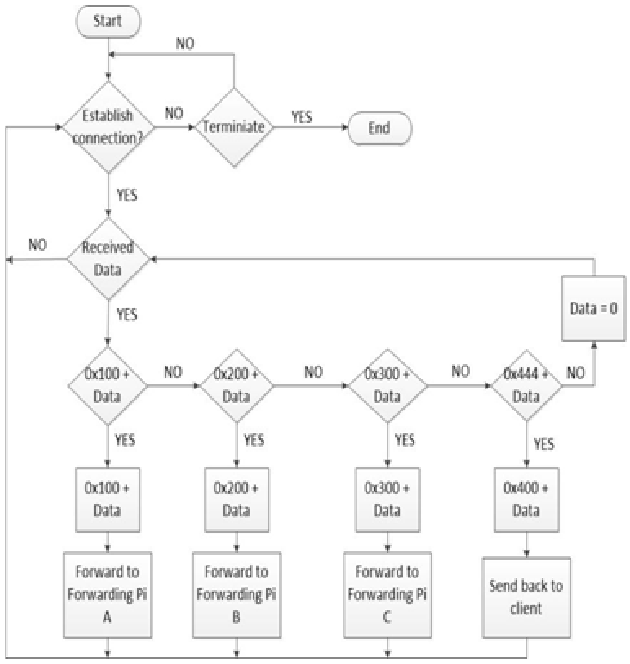

Fig. 2 Header Pi workflow to determine the packet label for data forwarding

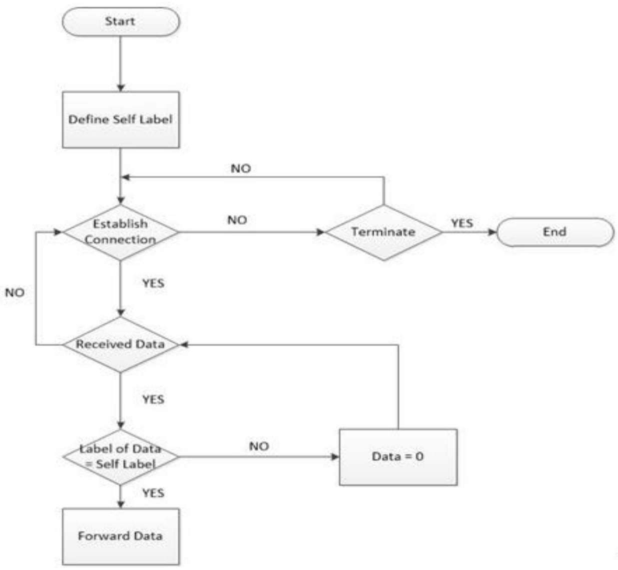

Fig. 3 Forwarding Pi working principle of determining either to eliminate or continue forward the data to the client

Testbed Setup

Our experimental testbed setup consists of a Raspberry Pi-based router and multiple wireless clients connected wirelessly.

The router is connected to Raspberry Pi client which will be used to generate and receive data. Furthermore, a server is also installed to the router testbed to allow all the nodes to communicate with one another. In order to emulate the wireless communication, Raspberry Pi Router B is connected to three wireless clients A, B, and C via Access Point. The server is installed with T-shark, a packet viewer software which helps to keep track of the forwarded and received packets so that the throughput, delay, and jitter of the router testbed can be tabulated for performance evaluation. The overall Raspberry Pi router setup is as shown in Figure 4. For comparison analysis, the experiment results of the Raspberry Pi-based router testbed will be benchmarked with an off-the-shelf wireless router performance. To serve that purposes, two computers will be set as a server and client are linked with one set of the wireless router as shown in Figure 5. The same performance parameters will be tabulated for both of the setups.

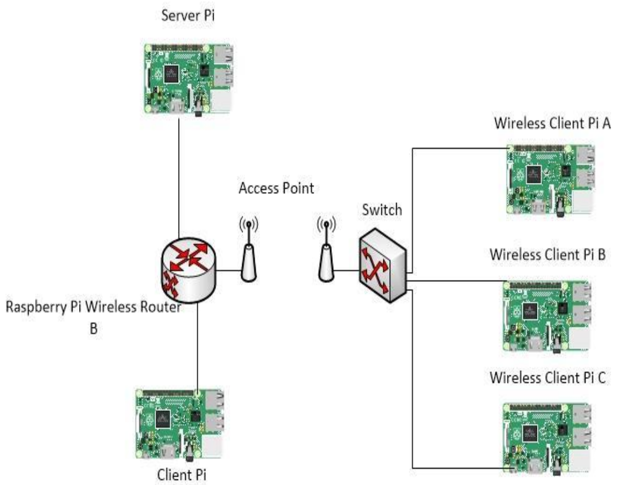

Fig. 4 Wireless Raspberry Pi router testbed experiment setup

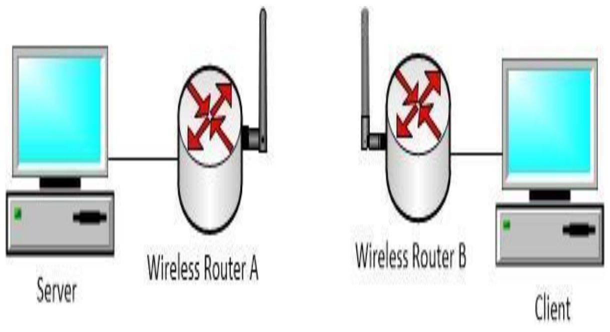

Fig. 5 Off-the-shelf wireless router for data benchmarking setup

IV. RESULTS AND DISCUSSIONS

The performance of Raspberry Pi router testbed in terms of throughput, delay and jitter parameter from a client to client is tabulated in this section. As stated in the previous section, the comparison is done between the Raspberry Pi router testbed and off-the-shelf wireless router for benchmarking purposes. The proposed Raspberry PI router testbed sends 1448 bytes of data packets at one time and due to that reason, the results are scaled to allow comparison to be done for both setups. The parameters are obtained by using packet analyzer, T-Shark and Wire shark for Raspberry Pi router and benchmark router respectively.

Parameters Result: Throughput

Figure 6 compares the average packet throughput for both the programmable Raspberry Pi router testbed and the commercialized router. The performance of data transfer based on offered can be specifically identified in the graphs. These results show that the Raspberry Pi router testbed has a similar throughput trend compared to the off-shelf-router. However, note that Figure 6 shows a slight decrease and not uniform throughput pattern occurs as the performance reduces as the throughput is observed to drop by 77,062 bps at maximum load due to prolonged usage of handing higher traffic network. Nonetheless, the Raspberry Pi router testbed provides the same throughput performance as the manufactured router.

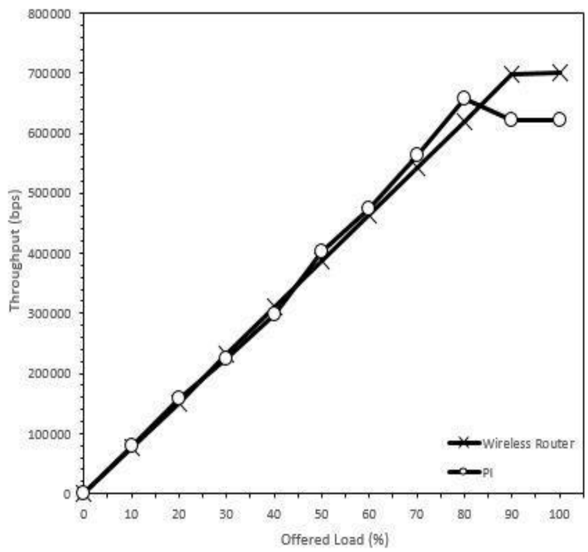

Fig. 6 Throughput comparison for Raspberry Pi router testbed and off-the-shelf wireless router

Parameters Result: Delay

Another performance comparison of the Raspberry Pi router test is the delay. Prior to getting into discussion, by using Raspberry Pi as the router element, high delay will be expected due to the processor process time. Therefore, it is reasonable to expect that the delay of data transmission in Raspberry Pi router testbed is higher than the off-the-shelf wireless router. The hypothesis is validated with the results of the experiment. The results in Figure 7 shows that the delay of data transmission of the Raspberry Pi router is recorded started higher than the wireless router by 0.7475s even at 10% of maximum offered load. This result occurred as the data is sent from the Server Pi to a wireless client Pi, the label data need be encapsulated to the packet and also be verified for forwarding by the Header Pi and the Forwarding Pi respectively. This process had caused the delay to become significantly larger compared to the off-shelf-wireless router. Fortunately, the trend of delay for data transmission of Raspberry Pi router testbed increasing with the same trend of the off-the-shelf wireless router.

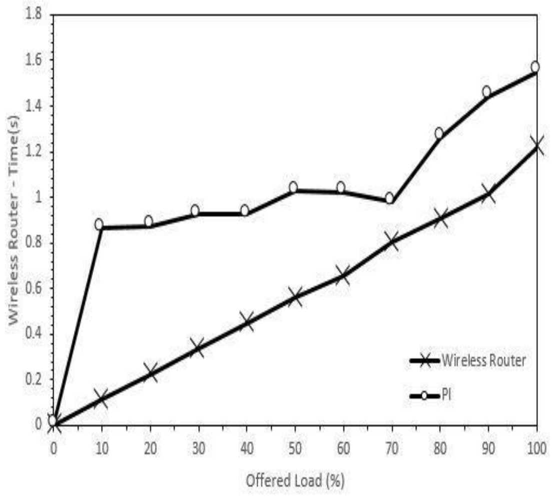

Fig. 7 Delay comparison for Raspberry Pi router testbed and off-the-shelf wireless router

Parameters Result: Jitter

The third performance evaluation of the programmable Raspberry Pi router testbed is jitter. Jitter is calculated to observe the rate of delay irregularities for sending the same offered load using the testbed. Figure 8 shows the comparison of the jitter for both the programmable Raspberry Pi router testbed and off-the-shelf router. The result shows that the Raspberry Pi router testbed has the highest jitter at 80% of the maximum offered load. It is also observed that the Raspberry Pi Router has an inconsistent trend due to the inconsistent time taken for the label encapsulation and data forwarding verification process. The jitter process also effected by the duration of the experiment as the heat from the computer board will affect the performance of the wireless router. Fortunately, the jitter for the data transfer of Raspberry Pi is observed to also have the same increasing trend with the off-the-shelf router.

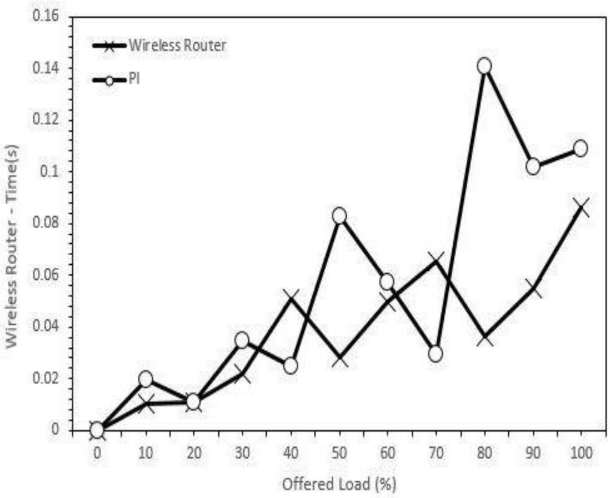

Fig. 8 Jitter comparison for Raspberry Pi router testbed and off-the-shelf wireless router

V. CONCLUSIONS

This paper presented a programmable wireless router test using Raspberry Pi, where each router consists of four units of Raspberry Pi which are, one Header Pi and three Forwarding Pis. The performance of Raspberry Pi router testbed for sending various file size in the network from the server to client are recorded in terms of throughput, delay and jitter client and the results are compared with the off-the-shelf router. Comparing the conventional wireless router with the Raspberry Pi router testbed, this system has proven to be reprogrammable and have a suitable scale to be used for academic and laboratories storage. Based on the results, the trend of graphs for both proposed Raspberry Pi router testbed and off-the-shelf router are in line with each other despite of the expected differences in terms of throughput, delay and jitter performance. Nonetheless, this programmable Raspberry Pi router testbed allows the researcher to explore and test new algorithms in hardware rather than software simulation.

ACKNOWLEDGMENT

The author gratefully acknowledges the funding of Yayasan Canselor UNITEN – UNIIG Seed Fund, J510050790.

REFERENCES

1. J. Lee, "Performance Evaluation of IEEE 802 . 15 . 4 for Low-Rate Wireless Personal Area Networks," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 742–749, 2006.
2. K. Xu, W. Chen, C. Lin, M. Xu, D. Ma, and Y. Qu, "Toward a Practical Reconfigurable Router: A Software Component Development Approach," *IEEE Netw.*, vol. 28, no. October, pp. 74–80, 2014.
3. R. Pi, M. Richardson, and S. Wallace, *Getting started with Raspberry Pi*. 2013.
4. S. G. Nikhade, "Wireless Sensor Network System using Raspberry Pi and Zigbee for Environmental Monitoring Applications," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, 2017, no. May 2015, pp. 376–381.
5. V. Vujovic and M. Maksimovic, "Raspberry Pi as a Wireless Sensor Node : Performances and Constraints," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014*, 2014, no. May, pp. 26–30.
6. V. Sandeep, K. L. Gopal, S. Naveen, A. Amudhan, and L. S. Kumar, "Globally accessible machine automation using Raspberry pi based on Internet of Things," *2015 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015*, pp. 1144–1147, 2015.
7. L. Caldas-Calle, J. Jara, M. Huerta, and P. Gallegos, "QoS evaluation of VPN in a Raspberry Pi devices over wireless network," *2017 Int. Caribb. Conf. Devices, Circuits Syst. ICCDCS 2017*, pp. 125–128, 2017.
8. K. Akash, M. P. Kumar, N. Venkatesan, and M. Venkatesan, "A single acting syringe pump based on Raspberry Pi - SOC," *2015 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2015*, pp. 1–3, 2016.
9. K. Kalaithasan, N. A. M. Radzi, and H. Z. Abidin, "Internet of things application in monitoring sick building syndrome," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 505–512, 2018.
10. V. Gupta, K. Kaur, and S. Kaur, "Developing Small Size Low-Cost Software-De fi ned Networking Switch Using Raspberry Pi," in *50th Annual Convention of Computer Society of India : Next-Generation Networks*, 2018, pp. 147–152.
11. G. P. Xavier and B. Kantarci, "Software Defined Networking challenges and future direction : A case study of implementing SDN features on OpenStack private cloud Software Defined Networking challenges and future direction : A case study of implementing SDN features on OpenStack private," 2016.
12. D. Posch, B. Rainer, S. Theuermann, A. Leibetseder, and H. Hellwagner, "Emulating NDN-based Multimedia Delivery," in *Proceedings of the 7th International Conference on Multimedia Systems, MMSys 2016*, 2016, pp. 270–273.

13. N. S. Yamanoor, "High Quality, Low Cost Education with the Raspberry Pi," in *GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings*, 2017, pp. 1–5.
14. J. Geerling, "Raspberry Pi 3 B+ Review and Performance Comparison| Jeff Geerling." [Online]. Available: https://www.jeffgeerling.com/blog/2018/raspberry-pi-3-b-review-and-performance-comparison. [Accessed: 12-Apr-2019].
15. C. W. Zhao, "Exploring IOT Application Using Raspberry Pi," *Int. J. Comput. Networks Appl.*, vol. 2, no. 1, pp. 27–34, 2015.
16. Manikanthan, S.V., Padmapriya, T., "An efficient cluster head selection and routing in mobile WSN" International Journal of Interactive Mobile Technologies, 2019.