International Journal of Engineering & Technology, 7 (2.15) (2018) 58-60

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Wireless Network Traffic Analysis and Troubleshooting using Raspberry Pi

Mohamad Nur Haziq Mohd Safri*, Wan Nor Shuhadah Wan Nik, Zarina Mohamad, Mumtazimah Mohamad

Faculty of Informatics and Computing, University Sultan Zainal Abidin, Besut Campus, 22200 Besut, Terengganu, Malaysia
**Email: 037387@putra.unisza.edu.my :*

Abstract

In the past five decades, computer network has kept up growing with the increases of its complexity. In such situation, the management, monitoring and maintenance of such computer network requires special attention to ensure optimal network access capability is achieved. Wireless network traffic analysis is a process of recording, studying and analyzing packets in wireless network for network performance analysis purposes. In some cases, the quality of network access performance can be very low without knowing the actual problem. Therefore, in this paper, the performance of wireless network traffic is proposed to be analyzed by using a Raspberry Pi which further able to send an alert to network admin to lessen the downtime. Raspberry Pi is a low cost, a small and portable size of a computer board that can be used to plug-in to monitor, keyboard, mouse, pen drive, etc. In this project, a MyTraceroute (MTR) program is installed on the Raspberry Pi to capture the IP of the Access Point (AP) and show packets loss percentage in the network. The results will be saved in the form of text file and sent to network admin by using email. The solution proposed in this paper is able to support solution to a problem on efficient monitoring, managing and maintaining wireless network traffics.

Keywords: *Network Traffic Analysis; Raspberry Pi; Network Troubleshooting.*

1. Introduction

Today, the Internet is one of the most effective and efficient ways to communicate with each other. Whether it is through Facebook, Google, Twitter, or another website, the internet gives us the opportunity to connect with all kinds of different people and read news and information from all over the world. Wi-Fi is a facility that allows smartphones, computers, and other devices to access the internet wirelessly. "The faster, the better" which refers to the speed of the internet that people can access. In the past five decades', computer network has kept up growing, the complexity of it and the number of users that use it also is being increased from time to times. Because of that, the amount of network traffic that flows over their nodes has increased drastically. Therefore, the management, maintenance, and monitoring of the network in real environment is very important to keep the network access smooth.

Wireless network traffic analysis is a process of recording, reviewing, and analyzing network packet in wireless network for performance analysis purposes. This project uses Raspberry Pi with Wi-Fi built-in to automatically identify problematic AP and send an alert to network admin which further speeds up the process of troubleshooting and lessen downtime.

Raspberry Pi is a low cost, small and portable size of a computer board that can be used to plug-in to monitor, keyboard, mouse, pen-drive, etc. [1]. There are two programming languages that have been used, i.e. Scratch and Python, which enable users to design animation or program and develop script respectively [1]. Many advantages of Raspberry Pi have been described in literatures. This includes a small independent computer that run independently, has very large working memory, possible to connect the Raspberry Pi with Wi-Fi and Bluetooth adapters. In addition, it is also cheaper than other wireless sensor nodes [2].

Wireless network traffic sometimes can be congested with network packet, e.g. in the university environments. In this environment, most students are accessing the Internet wirelessly which may introduce congestions in some events. In this case, a very low quality of network access may be experienced by users (in this case, the students and university staffs) and the process of troubleshooting the problem may take time. Therefore, the management, maintenance, and monitoring of the network in real environment is very important in order to maintain a good quality of wireless network access.

With the above mentioned problem, therefore, the objectives of this project are described as follows: 1) to study on how to use Raspberry Pi for detecting Wi-Fi coverage and send alert to network admin, 2) to implement Raspberry Pi as a Wi-Fi coverage detector, 3) to test and evaluate a Raspberry Pi as a Wi-Fi coverage detector in a real environment for personal use.

2. Related works

Firstly, from the perspective of using *Internet of Things* (IOT) devices, project in [3] employs a simple signal processing technique presented in ZiFi to detect signals from an Access Point (AP) on sensor nodes. In this article, the author proposed a sensor localization system using Wi-Fi AP's as anchors. Besides, Cheah Wai Zhao et. al. has conducted a project on "Exploring IOT Ap-

plication Using Raspberry Pi" [1]. In this paper, the XBee (ZigBee module) is used to demonstrate wireless communication data transmission. XBee was connected to Raspberry Pi to act as data transmitter and another XBee is connected to Window to act as a receiver.

Further, Tariq AL-Kadi et. al. has conducted a project on "Arduino Wi-Fi Network Analyzer" in [4]. In this paper, they focused on designing a simple Wi-Fi communication analyzer using Arduino microcontroller so that it interacts with Wi-Fi shield and display the results of Wi-Fi networks into an LCD monitor [4]. The authors highlight the advantage of their model which is claimed to be useful for distributing Wi-Fi network in a vast area.

Meanwhile, from the perspective of sending an alert to network administrator, project in [5] uses Python script to send email notifications every time a motion is detected. To allow for email notifications, SSMTP needs to be installed in OS [5].

Further, Patchava Vamsikrishna et. al. has conducted a project titled "Raspberry Pi Controlled SMS-UpdateNotification (SUN) System" [6]. In the project, the authors used *Short Message Service* (SMS) to send updates to users from authorized person (e.g. network admin) [6]. To complete the task, the authors use GSM module with Raspberry Pi, where the messages were read and displayed on the website captured by Raspberry Pi [6].

3. Design of network traffic analysis

In this section, our project framework and a project flowchart are presented, as shown in Fig. 1 and Fig. 2, respectively.

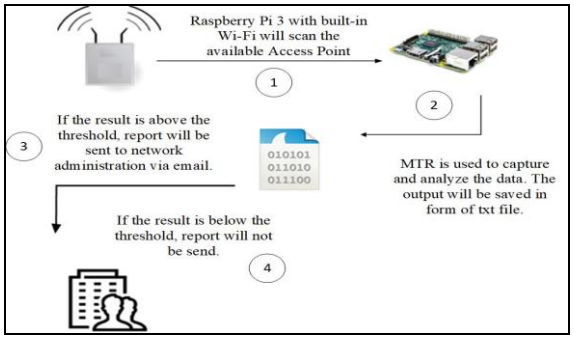

Fig. 1: Project Framework

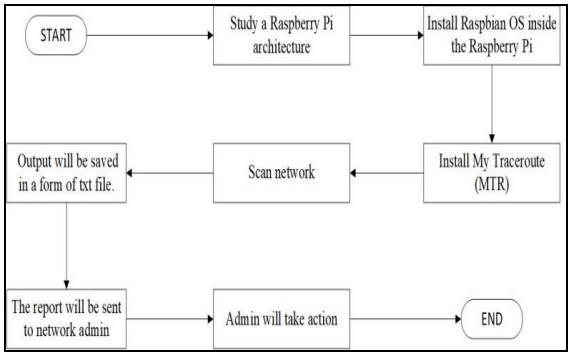

Fig. 2: Project Flowchart

Before the project can be started, the architecture of Raspberry Pi needs to be understood. Next, we need to install Raspbian OS into the Raspberry Pi. Raspbian OS is a free operating system based on Debian optimized for Raspberry Pi hardware. Other than that, it has a set of basic programs and utilities that make Raspberry Pi run.

In this project, the My Traceroute (MTR) tool is used to capture and analyze the network. MTR is a network diagnostic tool which combines ping and traceroute. By using this tool, the IP address of the AP can be obtained. In addition, this tool also capable to capture any packet loss in the network. Then, the output obtained from the scanning process will be saved in the form of flat file (e.g. file.txt).

Further, the generated output was sent to network admin via email. Python script is used to send the email. Then, the network admin will take action, e.g. to troubleshoot, if there is unusual activity is detected in the network.

4. Implementation on network traffic analysis

Raspberry Pi is a small, cheap, credit card sized computer board which can be connected together with computer LCD, keyboard and mouse to perform like a traditional computer. It is equipped with RAM, Hard Disk (SD card), audio and video ports, USB ports, HDMI port and Ethernet port. There are a lot things can be done by using this tiny computer. For example, the user can create spread sheet, word processing, browse the Internet, etc.

In this project, Raspberry Pi 3, B model is used. It has a built-in Wi-Fi adapter as an alternative to Wi-Fi dongle. Besides, this project use SD card with 8GB sized. Pi is Linux based, and the Raspbian OS is installed on the SD card. Raspberry Pi also provides default programming language, i.e. Python in addition to installation of other programming language such as C, C++, and Java.

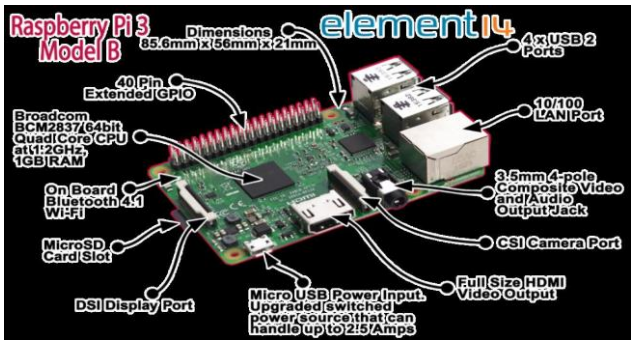

Fig. 3: Raspberry Pi Model B

MTR is a powerful network diagnostic tool that enables administrators to diagnose and isolate networking errors and provide meaningful information on network status to upstream providers. MTR is capable to capture IP address of the AP and show the packet loss percentage that occurs in the network.

Before installing the MTR, the Pi needs to be updated by using a ***$ sudo apt-get update*** command. This command is used to add repositories so that these repositories can be used to download and install any new software. Without this command, the installation of new software would fail as the location of the software will be unknown.

Next, the Pi needs to be upgraded by using the ***$ sudo apt-get upgrade*** command. This command installs latest versions of the software packages. After updating the lists, the package manager knows the available updates for the software that have been installed. Then, after running both commands, the MTR is ready to be installed in the Pi. The command to install the MTR is ***$ sudo apt-get install mtr***. After the installation is completed, the MTR is ready to capture the IP of the access point together with any packet loss incurred.

Further, a ***sudo mtr –r*** command is used to generate a report and save it in a flat file called mtr.txt. This command is used to capture details of all of the available access point in the range of the Raspberry Pi can capture. By using this command, the ESSID, Address, Frequency, Quality, and Cannel of the access point around the Pi can be captured and further can be stored in a flat file. This output will also be automatically updated in every scanning process.

Further, the information obtained from the capturing process is then sent to network admin via email. In this project, we use python language to send an email to the network admin. In order to do this, a new python file called ***myemail.py*** needs to be created with some python command in that python script.

5. Results and discussion

In this project, if the packet loss is more than 10%, the report will be automatically sent to network admin via email. In this process, the result which is a text file will be forwarded from 'mnhaziqsafri95@gmail.com' to the email address of network admin which is 'haziq.nabila@gmail.com' as shown in Fig. 4 and Fig. 5, respectively.

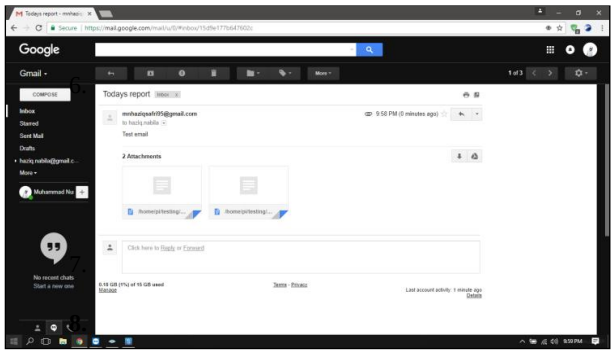

Fig. 4: Sender Email

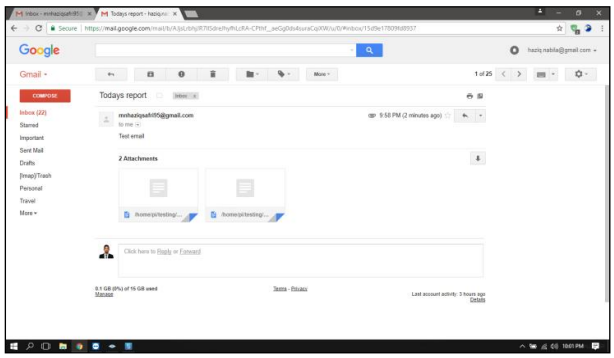

Fig. 5: Receiver Email

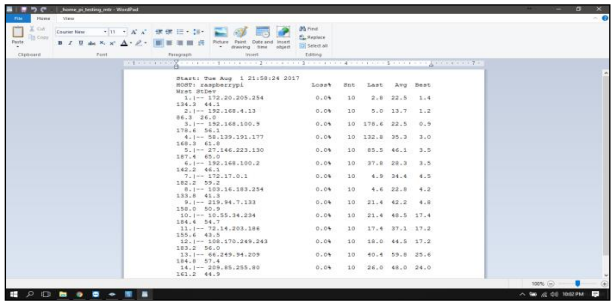

Fig. 6: Result from MTR scanning

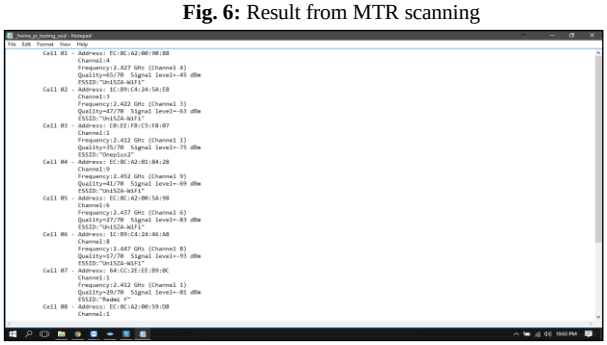

Fig. 7: Result from scanning the available access point

6. Conclusions

In conclusion, it is realized that a well-configured Raspberry Pi can be manipulated to become a powerful tool to capture useful information on wireless network signal provided by a respective AP and further provides an efficient monitoring, managing and maintaining wireless network traffics. Despite of this tiny *IOT* device, the Raspberry Pi is cheap and a portable device which can be allocated nearby AP.

Some future works can be done in order to improve this project. Some ideas that can be considered are: 1) Run the program automatically during start, and 2) Change the sending notification via email to web based so that the network administrator can get a real-time update on network access quality provided by the APs.

References

[1] Zhao, C. W. (2015). Exploring IOT Application Using Raspberry Pi. *International Journal of Computer Network and Application*, 27-34.

[2] Maksimovic, V. V. (2014). Raspberry Pi as a Wireless Sensor node: Performances and Constraints. *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 27-34.

[3] Ishida, Shigemi, et al. (2015). Wifi AP-RSS Monitoring using Sensor Nodes toward Anchor-free Sensor Localization. *IEEE*. Tokyo.

[4] Tariq, A.K., ZIyad, A.T., & Abdullah, A.O. (2013). Arduino Wi-Fi network analyzer. *Procedia Computer Science*, 522-529.

[5] Vamsikrishna, P., Kumar, S, D., Hussain, S. R., & Naidu, K.R. . (March, 2015). Raspberry Pi Controlled SMS-Update-Notification (Sun) System. *IEEE International Conference* (pp. 1-4). IEEE.

[6] Zafar, S., & Carranza, A. (n.d.). Motion Detecting Camera Security System with Email Notifications and Live Streaming Using Raspberry Pi. *Conference of American Society for Engineering Education*, (pp. 1-5).

[7] BIBLIOGRAPHY Biswas, J. (2014). An Insight to Network Traffic Analysis using Packet Sniffer. *International Journal of Computer Applications Volume 94*, 39-44.

[8] Cecil, A. (n.d.). A Summary of Network Traffic Monitoring and Analysis Techniques. 1-8.

[9] Danezis, G. (n.d.). Introducing Traffic Analysis Attacks, Defencs and Public Policy Issues. 1-12.

[10] Dhillon, N. K. (2012). Enterprise Network Traffic Monitoring, Analysis and Reporting using WINPCAP Tool with JPCAP API. *International Journal of Advanced Research in Computer Science and Software Engineering*, 95-101.

[11] Goldsmith, D. (1998). a Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists. *Cambridge Technology Partners*, 1-14.

[12] Hasan, M. K. (2012). IEEE 802.11b Packet Analysis to Improve Network Performance. *JU Journal of Information Technology (JIT), Vol 1*, 27 34.

[13] Hasib, M. (2006). Analysis of Packet Loss Probing in Packet Networks. 1-180.

[14] Patel, P. A. (2012). Network Traffic Analysis Using Packet Sniffer. *International Journal of Engineering Research and Application*, 854-856.

[15] Qadeer, M. A. (2010). Network Traffic Analysis and Intrusion Detection using Packet Sniffer. *Second International Conference on Communication Software and Networks*, 313-317.