QoS Evaluation of VPN in a Raspberry Pi devices over Wireless Network

L. Caldas-Calle, *Student Member, IEEE*, J. Jara *Member, IEEE*, M. Huerta, *Senior Member, IEEE and* P. Gallegos.

***Abstract*— The different benefits of WLANs have achieved their massive implementation in different sectors: domestic business and intelligent cities, among others. However, security in data transmission is one of their biggest problems, as they are more susceptible to suffer different types of wireless network attacks. This disadvantage motivates users to use Virtual Private Networks (VPN) in order to reinforce the security of the data in the network. VPN overload data traffic including another layer , in this article presents the evaluation of VPN QoS parameters over a Wireless Network (WLAN), using different Raspberry Pi models. Additionally, we analyzed the QoS parameters with different traffic conditions and the CPU consumption in the VPN. The contribution of the results and their analysis determine the correlation between the parameters and the Raspberry Pi models. Getting better performance on Raspberry Pi Zero and Raspberry Pi 2.**

***Index Terms*— Quality of Service, Virtual Private Networks, Raspberry Pi.**

I. Introduction

Wireless Local Area Networks (WLANs) are popular because of the low cost of installation, easy configuration and accessibility. These characteristics allow their proliferation in homes, small businesses and public places [1]. Nowadays, cities have implemented public wireless networks in order to provide better services to citizens, becoming smart cities. The city of Cuenca in Ecuador does not escape this reality and has initiated a project to provide Internet access, using WLANs [2].

However, wireless networks are more susceptible to suffer attacks: unauthorized access, man-in-the-middle, dissemination of information, Denial of Service, among others [3] [4]. This disadvantage motivates the users to use Virtual Private Networks (VPN) to strengthen network security. VPN provides confidentiality and data integrity by encrypting and authenticating the traffic of a link between two or more network devices. The link between the devices is performed by a communication channel called a tunnel, which is protected using extreme to extreme encryption. Integrity and authentication is obtained through authentication algorithms, key exchange mechanisms or certificates [5]. For this reason, VPNs are used to connect remote devices to private networks through a public network.

However, using a VPN overload data traffic including another layer, causing effects on throughput, latency, frame loss rate, among other parameters, affecting the Quality of Service (QoS) of the network [6].

Different researchers studied the behavior of the QoS parameters in a VPN over a wireless network. In [7], Kolahi et al. determined the behavior of throughput and Round Trip (RTT) parameters in VPN protocols (SSL and IPSec) with Windows. In [8] the results of the measurements determined that throughput, latency, frame loss rate and packet delay variation parameters decreased the QoS of the network.

In [9] the analysis of the use of the CPU when applying software for the creation of VPN is presented. In [10] the authors analyzed the use of OpenVPN in a Tablet with Android Operating System. Analyzing VPN QoS and CPU consumption on a low-cost Single Board Computer (SBC), Raspberry Pi, would allow users to be referred to the effect of their use on them. However, there are no studies to determine the effect of wireless VPN on Raspberry Pi.

This article presents the evaluation of VPN QoS parameters over a WLAN, using the Raspberry Pi models: Pi Zero v1.3 (RPZ), Model B Rev 2 (RPB), Pi 2 Model B v1.1 (RP2) and Pi3 Model B (RP3). Additionally, we analyzed the QoS parameters with different traffic conditions and the CPU consumption in the VPN.

II. QoS In A Wireless VPN

VPN allows creating a secure extension of a private network over a public network. There are several protocols for establishing a VPN, given by the Internet Engineering Task Force (IETF), two of them are: SSL and TLS. TLS consolidates its predecessor's SSL certificates into a standard protocol. SSL/TLS is extensively developed to correct its vulnerabilities, providing security and trust, working in a transparent way for the user [11]. Consequently, users use this protocol to create wireless VPN.

In the development of technologies for Wireless Networks, such as 802.11, prioritizes connectivity, throughput and other functionalities, giving security as a second space [8].

The Benchmarking Methodology Working Group (BMWG) states in RFC 2544 [12], the parameters definitions that are

L. Caldas-Calle, J. Jara, M. Huerta and P. Gallegos was with Universidad Politécnica Salesiana, Cuenca, Azuay in Ecuador. (e-mail: lcaldascalle@gmail.com, jjaras@ups.edu.ec, mhuerta@ieee.org and pgallegos@ups.edu.ec).

978-1-5386-1962-9/17/$31.00 ©2017 IEEE

used to describe behavioral characteristics in the interconnection of networks, these are: Throughput, Latency and Frame Loss Rate (FLR).

Throughput is the real rate at which information is transferred over a period of time. It is controlled by factors such as loss of packets or retransmissions, transport layer protocols, use of shared media, signal to radio, hardware limitations, among others. These factors establish that throughput is less than the bandwidth.

Latency is defined as the time interval that the last bit of the incoming frame reaches the input port at the beginning and when the first bit of the same frame is seen at the output port at the end. Latency is considered as the delay between it sent from the information, from the sender, and the decryption in the receiver, this being the RTT. WLAN have higher latency than LANs because of their wireless access.

FLR is the percentage of frames lost between the interface of the transmitter and the interface of the receiver. The loss is caused by network congestion and wireless link signal level due to lack of resources. Having a higher percentage of FLR causes a negative impact on throughput and latency, which causes a lower perception of the connection speed. In wireless networks, a high FLR affects the throughput of a transmitter. Similarly, when using reliable transport protocols, packet loss increases packet retransmission and increases latency.

The analysis of the behavior of these QoS parameters allows determining the level of impact of the VPN in a wireless link. These parameters are dependent on the level of the Central Processing Unit (CPU) consumption during VPN usage [9] [13]. The CPU consumption in the wireless network by the VPN depends on the protocol that uses the tunnel and the encryption algorithm. Thus processes such as encapsulation, routing, and encryption are transformed into additional tasks by increasing the sum of the number of processes being executed and the number of processes that are waiting to be executed [14].

III. Design Of The Evaluation System

The block diagram of the designed test architecture is show in Fig. 1. It describes the devices connected by a 2.4 GHz IEEE 802.11n wireless link, which was supplied by a wireless router at a distance of three meters from the devices. The architecture of evaluation was configured in three blocks: transmitter, receiver and communication channel.

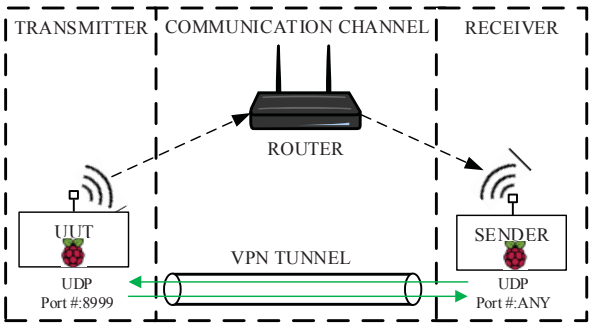

Fig. 1. Test Architecture

A. Transmitter

The transmitter, referred to as Unit Under Test (UUT), is the SBC that was used for the evaluation system, its technical characteristics are detailed in TABLE I.

TABLE I
SBC Technical Specifications

SBC: Raspberry Model	SoC Tipe	Core Tipe	Core No	CPU Clock	RAM
Pi Zero v1.3 (RPZ)	Broadcom BCM2835	ARM 1176JZF-S	1	1 GHz	512 MB
Model B Rev 2 (RPB)	Broadcom BCM2835	ARM 1176JZF-S	1	700 MHz	512 MB
Pi 2 Model B v1.1 (RP2)	Broadcom BCM2836	ARM Cortex-A7	4	900 Mhz	1 GB
Pi 3 Model B (RP3)	Broadcom BCM2837	ARMv8 64-bit	4	1.2 GHz	1 GB

The UUT generated and captured the packages using the Iperf software. This Open Source software is extensively developed and tested to return reliable performance measures. Iperf is used as a primary tool to produce continuous flows of UDP datagrams, in time cycles of 120s, and to establish throughput and FLR (packet sizes are set in [12]). The throughput that was previously established was used to determine the latency and average CPU consumption. CPU consumption was measured using the Linux uptime command. Latency was measured by using the ping command and using Eq. (1). This equation determined the values placed in Packets Per Second (PPS) of the command to obtain the RTT.

$$V_{TX} = \left(\frac{TotalPacke\ tSize\left[Bytes\right] \times 8\left[bytes\right]}{1000} \right) \times PacketRate\left[PPS\right] \quad (1)$$

B. Receiver

The sender was a Pi 3 Model B. It and the UUT were synchronized with the same port for transmission in order to operate in a client-server mode. The UUT and the sender used the Raspbian Jessie Lite O.S.

C. VPN Tunnel

This study usedOpenVPN an Open Source SSL/TLS, OpenVPN establishes a VPN with security methods (RSA certificates and keys) allow authentication and security, providing mechanisms of reliability, stability and encryption [15]. Moreover, this software gives accessibility to configure several parameters of the tunnel, which allows customizing its behavior. In this study several relevant parameters to maximize tunnel performance were chosen: such as an upper limit of the packet size of 1500 bytes, fast compression LZO, encrypt data channel packets with the AES-256-CBC algorithm. The other features were chosen by OpenVPN documentation recommendations: package encapsulation in layer three with the Tun interface, UDP transport level protocol that provides faster throughput.

IV. RESULTS

This section shows the measurements obtained from the experiments that were performed to analyze and compare the throughput, FLR, latency and CPU consumption level for UDP datagram flows on a SSL/TLS wireless VPN when using the Raspberry Pi models and the platform described in section III. For ease of analysis, the throughput, FLR, and CPU consumption level results are plotted and the latency results are detailed in TABLE II.

The results of the measurements for the throughput for the RPZ, RPB, RP2, and RP3 models are show in Fig. 2.

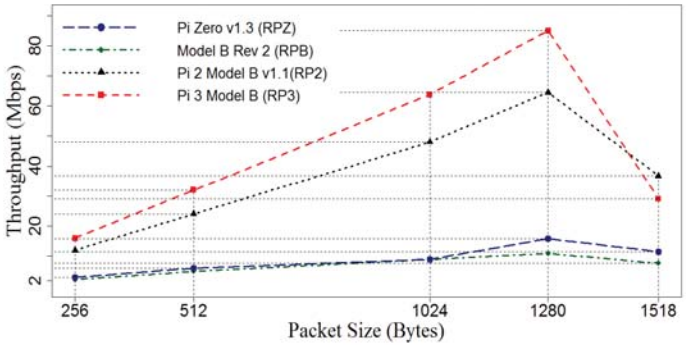

Fig. 2. Comparison of UDP Throughput in models: Pi Zero v1.3 (RPZ), Model B Rev 2 (RPB), Pi 2 Model B v1.1 (RP2) and Pi 3 Model B (RP3).

The measurements show the comparison of maximum throughput for each model of Raspberry Pi, under conditions described in section III. The best performance corresponds to RP3 which is 85 Mbps. The lowest performance was the RPB, it presented a maximum of 11 Mbps which represents a rate of decrease of 87.05% compared to RP3. Both results are obtained for the packet size of 1280 bytes.

On all devices, the throughput values reach a maximum, then a sudden drop is experienced. This is due to the fragmentation of the packets. Fragmentation occurs when the maximum payload is greater than a value beyond 1472 bytes (1500 bytes minus 20 + 8 bytes of IP header and UDP [16]). In this case, the 1518 byte frame will be fragmented into more than one frame, which reduces throughput.

The results obtained from latency (RTT) for the traffic generated with the throughput of Fig. 2, in the specified frame sizes and the PPS when using Eq. (1) are shown in Table II. Increases in RTT times for packets that are larger than the fragmentation point can be evidenced, being more evident in RPZ and RPB models. It is due to the variation in queue sizes as a result of fragmentation overheads and the wireless interface hardware [13]. Additionally, the results show the correlation between the rate of packet generation at a given size and the average time for the packet to complete the RTT presenting the buffer dependency of each model.

TABLE II
RTT FOR GENERATED TRAFFIC

Raspberry Model	Frame Sizes (bytes)	Packages Per Second (FPS)	RTT (ms)
Pi Zero v1.3 (RPZ)	256	1321	8,40
	512	1389	8,80
	1024	1070	8,94
	1280	1529	8,78
	1518	970	116,99
Model B Rev 2 (RPB)	256	968	27,85
	512	1157	8,35
	1024	1069	8,69
	1280	1051	8,73
	1518	946	165,99
Pi 2 Model B v1.1 (RP2)	256	5281	4,47
	512	5555	4,46
	1024	5703	4,44
	1280	6211	4,58
	1518	3153	5,19
Pi 3 Model B (RP3)	256	7042	4,81
	512	7407	4,79
	1024	7604	4,87
	1280	8123	4,70
	1518	2425	4,91

The results for FLR of reception for each of the models can be observed from Fig. 3 to Fig. 6.

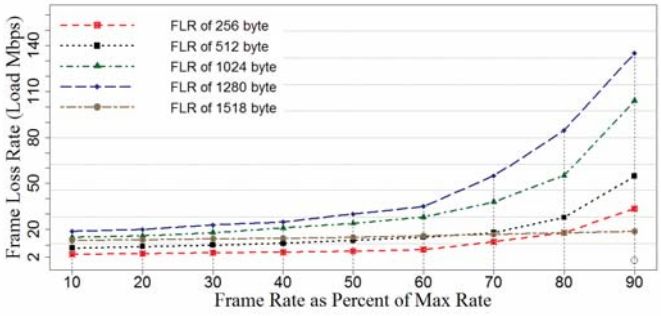

Fig. 3. Loss of packets in the traffic reception Pi Zero vl.3 (RPZ)

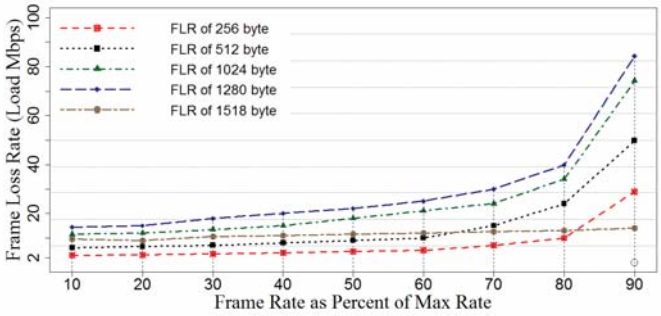

Fig. 4. Loss of packets in the traffic reception Model B Rev 2 (RPB).

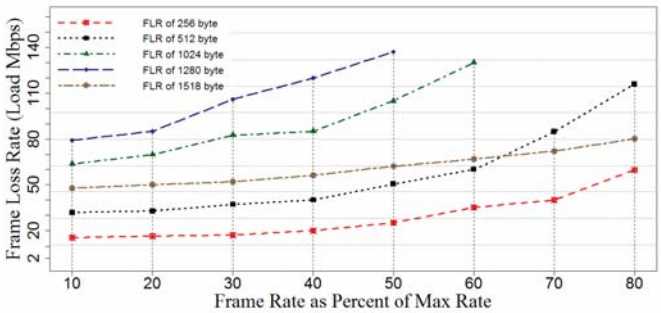

Fig. 5. Loss of packets in the traffic reception Pi 2 Model B v1.1 (RP2).

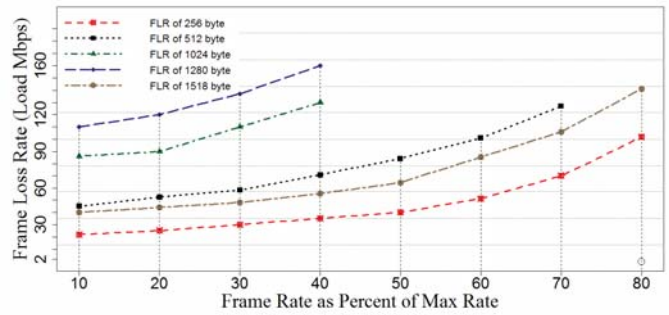

Fig. 6. Loss of packets in the traffic reception Pi 3 Model B (RP3).

The FLR results were determined to obtain in sequence (every 10%) the levels of packet loss for each model. In Figures 3, 4, 5 and 6, it can be observed that there is a positive correlation between the generated traffic load and the FLR, as the load increases the FLR.

In Figures 5 and 6, the RP2 and RP3 models do not have all percentages of losses because they exceed the maximum data transfer capacity of 150 Mbps of the router.

The FLR measurements indicated the abrupt changes after the fragmentation point similar to the results obtained in the throughput and latency parameters. This is attributed to the RTT and packet transmission cycle per second leading to the drop in UDP bandwidth. Similarly, the loss of packets was caused by the capacity of the data buffer of each model.

The results of the average CPU utilization for the RPZ, RPB, RP2, and RP3 models are shown in Fig. 7.

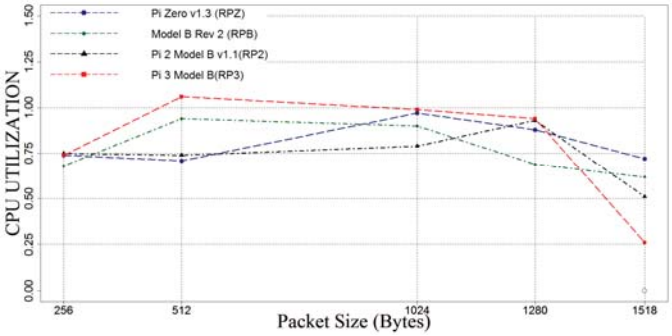

Fig. 7. Average CPU Consumption in models: Single-Core of the Pi Zero v1.3 (RPZ), Single-Core of the Model B Rev 2 (RPB), Quad-Core of the Pi 2 Model B v1.1 (RP2) and Quad-Core of the Pi 3 Model B (RP3).

The results of the CPU consumption level show the average usage for traffic generated with the specified frame sizes and the throughput obtained. A considerable number of processing cycles are required for the operation of the wireless VPN. For the RP2 and RP3 models, the fragmentation point gives CPU consumption a noticeable drop. This phenomenon is given by the FLR produced in the layers: physical, link and network. The loss of packets decreases the bandwidth which results in a smaller number of packets to be processed and reducing the length of the execution queue of the core.

The highest CPU consumption is 0.97 and 1.02 (max CPU utilization is 4, because RP3 have 4 cores) for the RPZ and RP3 models respectively with packet flows in 1024-byte and 512-byte sizes. This indicates that the Single-Core RPZ was at its maximum capacity and one of the cores of the RP3 exceeded its capacity. As the CPU approaches its maximum capacity, the performance of the VPN show a reduction being notorious in the models RP2 and RP3. Therefore, the CPU is an important factor that helps to determine the QoS of a wireless VPN.

V. Conclusion

This article was presented the evaluation of VPN QoS parameters over a WLAN, using different Raspberry Pi models. The results obtained indicate that the highest throughput is for RP3 and the lowest for RPB. Values indicate dependence latency buffer each model based on the average time RTT, It is more evident in RPZ and RPB. Packages with size beyond the fragmentation point suffer QoS decrease, due the need to fragment packets. The CPU power of each Raspberry Pi model is an important factor affecting the QoS parameters of a wireless VPN. Introducing VPN to secure communication, implies more complex process in communication that requires more from hardware.

References

[1] Wang Shunman, Tao Ran, Wang Yue, and Zhang Ji, "WLAN and it's security problems," 2003, pp. 241–244.

[2] ETAPA EP, "Noticias." [Online]. Available: http://www.etapa.net.ec/Noticias/newid/105/title/ETAPA_EP_AMPLIA_COBERTURA_WI_FI_EN_PARQUES_Y_PLAZAS. [Accessed: 13-Jan-2017].

[3] P. Feng, "Wireless LAN s ecurity issues and solutions," 2012, pp. 921–924.

[4] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 180–189.

[5] B. Hoekstra, D. Musulin, and J. J. Keijser, "Comparing TCP performance of tunneled and non-tunneled traffic using OpenVPN," *Univ. Van Amst. Syst. Netw. Eng. Amst.*, pp. 2010–2011, 2011.

[6] A. K. Agarwal and W. Wang, "Measuring performance impact of security protocols in wireless local area networks," 2005, pp. 625–634.

[7] S. S. Kolahi, Y. Cao, and H. Chen, "Impact of SSL security on bandwidth and delay in IEEE 802.11n WLAN using Windows 7," 2016, pp. 1–4.

[8] P. Likhar, R. S. Yadav, and M. Keshava Rao, "Securing IEEE 802.11g WLAN Using Open VPN and its Impact Analysis," *Int. J. Netw. Secur. Its Appl.*, vol. 3, no. 6, pp. 97–113, Nov. 2011.

[9] C. J. C. Pena and J. Evans, "Performance evaluation of software virtual private networks (VPN)," 2000, pp. 522–523.

[10] J. Qu, T. Li, and F. Dang, "Performance Evaluation and Analysis of OpenVPN on Android," 2012, pp. 1088–1091.

[11] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley, 2001.

[12] S. Bradner and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices," RFC Editor, RFC2544, Mar. 1999.

[13] K. S. Munasinghe and S. A. Shahrestani, "Wireless VPNs: An Evaluation of QoS Metrics and Measures," 2005, pp. 616–622.

[14] S. Narayan, S. S. Kolahi, K. Brooking, and S. de Vere, "Performance Evaluation of Virtual Private Network Protocols in Windows 2003 Environment," 2008, pp. 69–73.

[15] OpenVPN Community Software, "HOWTO." [Online]. Available: https://openvpn.net/index.php/open-source/documentation/howto.html. [Accessed: 13-Jan-2017].

[16] S. Garg and M. Kappes, "An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks," 2003, vol. 3, pp. 1748–1753.