Wireless VPNs: An Evaluation of QoS Metrics and Measures

Kumudu S. Munasinghe and Seyed A. Shahrestani
School of Computing and Information Technology
University of Western Sydney, Locked Bag 1797
Penrith South DC, NSW 1797, Australia
kmunasin@cit.uws.edu.au and seyed@computer.org

Abstract

In this paper, the analysis and experimental results for an evaluation of the Quality of Service (QoS) of Virtual Private Networks (VPNs) over an IEEE 802.11b wireless network are presented. A comprehensive set of measurements on the application throughput, packet loss, CPU utilisation, and round-trip delay for different traffic conditions are obtained and analysed. These results and analysis further indicate the degree of contribution made by the packet generation rate, the payload data size, and the CPU processing power on the overall performance of a wireless VPN under general enough circumstances.

1. Introduction

With the rapidly growing adoption of the wireless networking technology, for many implementers, security and quality of service (QoS) are issues of utmost priority. The main reasons for growing concerns in security are the insufficiencies of the basic security services offered by the IEEE 802.11 standard [1],[2]. Then again, QoS is increasingly becoming important for delivering next generation wireless multimedia applications. This motivated research into exploring alternate avenues for the enhancement of the required security solutions. One such method is based on the implementation of Virtual Private Networks (VPNs) over the wireless infrastructure.

In our earlier works, performance issues relevant to the use of a single Internet Protocol Security (IPSec) VPN in a practical and experimental IEEE 802.11b Wireless Local Area Network (WLAN) were analysed and reported [3]. The results indicated that even a single VPN may have a significant impact on the throughput, packet loss, and delay for the end-system. The analysis also indicated that, under such circumstances, CPU limitations may contribute to the generation of a possible bottleneck in the performance of an IPSec VPN.

In this paper, the results of expanding our work for inclusion of QoS limitations and their causes of IPSec VPNs in a wireless environment are reported. Using the light-weight User Datagram Protocol (UDP), we investigate the QoS limitations by increasing the number of simultaneously operating IPSec VPNs. The remainder of this paper is organized as follows. The next section is an overview of wireless VPNs. Following that, the setup used for measurements and the results of experiment are discussed. In Section 4 and 5 the analysis of the performance results are presented. The last section gives the concluding remarks.

2. VPNs in a wireless environment

Implementing VPNs over the wireless infrastructure is a generally well-received solution for securing a wireless communications. IPSec, standardized by the Internet Engineering Task Force (IETF), is a suit of protocols that is widely used in VPNs to provide encryption, authentication and integrity services. Two of the main protocols defined in IPSec are Authentication Header (AH) [4] and Encapsulating Security Payload (ESP) [5]. The AH ensures the integrity and authenticity of the IP payload or the entire IP datagram. The ESP ensures the confidentiality of the IP payload or the entire IP datagram and may optionally provide authenticity and integrity.

Amongst many open research questions on wireless VPNs remain the issue of Quality of Service (QoS). As the existing IEEE 802.11 standard does not address the above issue, wireless networks are susceptible to errors owing to the nature of its link layer environment. A VPN over a wireless link may negatively impact on its performance levels. Therefore, it is highly important to understand the behaviour of the QoS parameters.

Proceedings of the International Conference on Mobile Business (ICMB'05)
0-7695-2367-6/05 $20.00 © 2005 IEEE

3. Research methodology

The experimental setup includes two desktop PCs, referred to as wired client (CPU 1.7GHz) and wireless client (CPU 300MHz) as shown in Figure 1. The wired client and the IEEE 802.11b 2.4 GHz Access Point (AP) are both connected to the network via 100 Mbps Fast Ethernet interface cards. The wireless client has an IEEE 802.11b 11Mbps interface configured in infrastructure mode. The distance between the AP and the wireless client is around 3 meters.

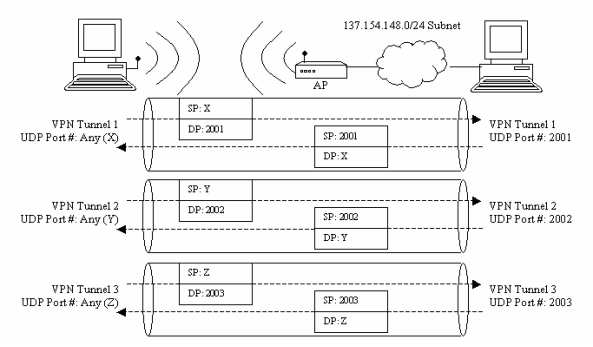

Figure 1. Experimental Setup.

Both the wireless and the wired clients use MS Windows 2000 SP2. For traffic generation and capturing the fully featured LanTraffic V2 is used. The CPU utilisation at the wireless client is measured by the MS Windows 2000 Performance Monitoring utility. A separate IPSec policy is configured for each VPN with Pre-shared key authentication.

The first stage of the investigation is to establish the baseline conditions. The wireless client is configured to generate streams of UDP traffic to the wired client. The destination captures the incoming traffic and then echoes it back to the source as shown in Figure 1. Ten thousand UDP packets are generated with fixed repeating packet content and an inter-packet generation gap of 1ms. Starting from 25 bytes, the UDP payload is gradually increased up to a maximum of 1,600 bytes to generate higher traffic levels. A comprehensive set of measurements on the application throughput, packet loss, the round-trip delay and the CPU utilisation at the source are taken. Each reading is repeated a minimum of five times and the mean values of these samples are used in the subsequent analysis.

In the next stage a single IPSec VPN is setup and the same steps are repeated. Subsequently, as shown in Figure 1, the number of simultaneously operating IPSec VPNs are increased to two and then to three respectively. The same steps as in the original experiment were repeated on the multiple VPN setups. All other parameters are kept without any change. Finally, the complete experiment is repeated for an increased inter-packet generation gap of 5ms.

4. Results analysis

In this section a detailed analysis of the performance results from the experimentation described in the previous section is present. In most cases, the performance results are plotted, as graphs provide for easy comparisons and quick references.

4.1. Throughput

The measured average throughput can be defined as the average amount of data payload transferred over the time duration between two points in the same service area [6]. Figure 2 and Figure 4 represent the average throughput graphs for multiple IPSec VPNs for UDP traffic flows with 1ms and 5ms inter-packet generation gaps respectively.

The results in Figure 2 indicate that the baseline value for maximum achievable average throughput, under the given conditions, is 5.97 Mbps. This result is in line with some of the previously published results [7], [8]. When a single IPSec VPN is activated, this value drops to 4.94 Mbps. Figure 4 shows that for increased inter-packet generation gaps, the corresponding highest achievable average throughput for baseline and the single IPSec VPN setups have relatively decreased. Nevertheless, it is worth noting that as a result of the excessive overheads of the media access control mechanism (i.e., CSMA/CA) of the IEEE 802.11 standard, a substantial amount of throughput is also compromised. Such overheads include preambles of the transmitted frames, MAC headers, ACK frames, transmission protocol overheads, processing delays in stations, forwarding delays around the APs and the random back off periods [8], [9].

Figure 2 illustrates that as the number of simultaneously operating VPNs is increased, a considerable reduction in the total average throughput can be noticed. The graphs in Figure 2 also reflect that there may be a possibility for reduction in the per VPN average throughput as the total number of simultaneously operating VPNs increase. However, Figure 3 reveals that it need not always be true. In fact, Figure 3 indicates that for UDP flow with payload sizes up to 150 bytes, the throughput graphs behave in the opposite direction. The highest total average throughput value in Figure 3 is recorded for the setup of 3 simultaneously operating IPSec VPNs. The second highest total average throughput value corresponds to the setup of 2 simultaneously operating IPSec VPNs.

The baseline and the single VPN show the lowest throughput values in Figure 3.

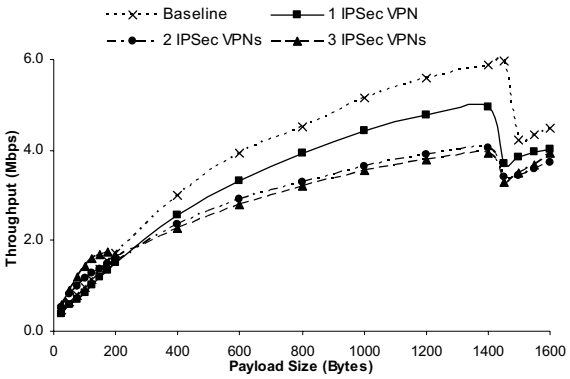

Figure 2. The Average Throughput Graphs for Traffic Generated with a 1ms inter-packet Generation Gap.

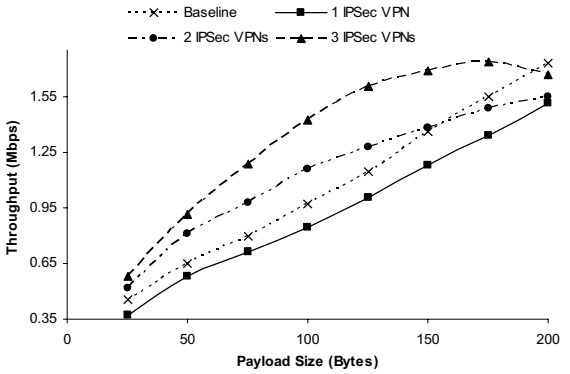

Figure 3. Expanded Section of Figure 2; the Average Throughput Graph for Payloads from 25 to 200 bytes.

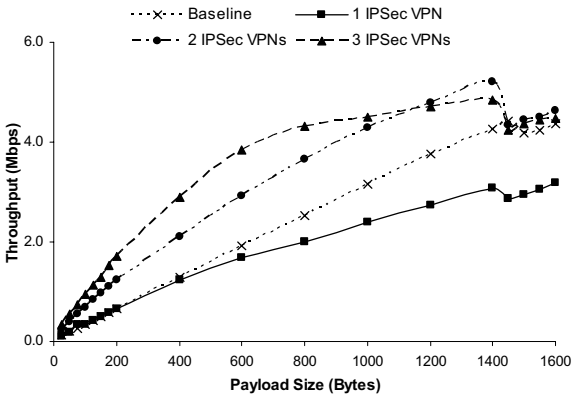

Figure 4. The Average Throughput Graphs for Traffic Generated with a 5ms Inter-packet Generation Gap.

Furthermore, the throughput graphs in the region of 1000 bytes and lower in Figure 4, closely resembles the graphs in Figure 3. Both of these show how the peak performing multiple VPN setup increasingly suffers reductions in the rate of change (increase) in throughput. Refer to case in Figure 4; this effect is experienced by UDP traffic with payloads of 600 bytes and over. Figure 3 and Figure 2 also experience a similar effect, at a much earlier stage. Form these two cases; it is clear that the packet generation rate and the packet payload size are the two dominant factors affecting the performance of simultaneously operating multiple VPN setups.

In all the above cases, as the throughput values reach a maximum point, a sudden (but temporary) drop is experienced. This is related to the fragmentation of the IP datagrams. This phenomenon can be expected when the payload increases beyond 1472 bytes (i.e., 1500 bytes for Ethernet frame – 20 bytes for IP header – 8 bytes for UDP header) for the baseline curve. Payloads larger than 1472bytes get fragmented into more than one datagram reducing the net throughput rapidly [14]. A similar behaviour can be noticed for the average throughput graph of the IPSec VPN. However, this happens somewhat earlier when the payload increases beyond 1438 bytes. This is due to the additional headers introduced by the ESP encapsulation.

4.2. Packet loss

The packet loss metric studied here, specifically investigates the average per tunnel packet loss in transmission (outbound) and receiving (inbound) processes. These relate to the wireless client for UDP traffic generated at two different rates represented by Figures 5 to 8.

As it is clear from Figures 5 and 6 there is a high packet loss (up to approximately 35% in transmission and 20% in receiving) experienced for UDP datagrams with light payloads (i.e., 25 to 50 bytes), generated at 1ms intervals. Similar trends in packet loss have already been identified and published [10],[11],[12]. One such argument is that the buffer shortages at the UDP level, due to the very fast generation rate of short packets, may have caused the datagrams to drop [10]. In spite of this, such a condition is most likely to arise for bursty traffic. Hence, for the type of UDP flow used in this experiment, UDP buffer overflow cannot be ruled out as the primary cause for packet loss. A second explanation to this phenomenon points towards the behaviour of the lower network layers, data link layer or physical layer [11], [12]. These researchers argue that, when a wireless interface card deals with frames having such short payload sizes, the overheads occupy most of the frame. The bidirectional traffic arriving continuously causes increase in the total traffic

and contention. When such frames arrive continuously at the wireless interface the processing ability of the interface card decelerates and the packets eventually drop off. Figure 5 further indicates that as the number of simultaneously operating VPNs is increased, it is obvious that this situation becomes worse.

Comparing the results in Figure 5 and Figure 6 against those in Figure 7 and Figure 8 indicates that, for traffic with smaller inter-packet generation gaps, packet loss is higher. Furthermore, as the UDP payload size is gradually increased, the packet loss also increases for baseline as well as for IPSec VPN setups. This is again in line with previous works in the field [10]. The cause can be explained in the following manner. As the payload data size increase the transmission delay at the interface increases. As a result, at high packet generation rates (i.e., at 1ms inter-packet delays), relatively larger packets may experience increased queuing delays. Consequently, a bottleneck situation is formed at the wireless interface. In general, when a UDP datagram is delayed as a result of queuing up to the maximum delay limit, it may get dropped [13].

Despite the high packet generation rate, it can be noted that IPSec VPNs in Figure 5 achieve relatively low packet losses compared to its baseline situation. This condition only applies to UDP traffic with payloads of 400 bytes and over. This can be explained by noting that, at the network layer, a UDP datagram spends a comparatively longer time for the IPSec encryption process. This causes the fast UDP flow to actually slow down. This prevents or at least reduces the queuing at the interface. As a result, the packet losses are relatively less for an IPSec VPN, even at high packet generation rates. Figure 7 shows how packet loss reduces to a minimum as the inter-packet delay of the UDP traffic is increased. As discussed before, as the inter-packet generation gap increases to 5ms, queuing at the interface may eventually be eliminated. This results in very low levels of packet loss.

In the previous discussions, we mainly concentrated on one aspect of packet loss. That is, the packet loss during transmitting the data from the wireless client to the wired client as shown in Figure 5 and Figure 7. Figure 6 and Figure 8 show the second aspect of packet loss. This relates to the loss in receiving the inbound echoed packets, from the wired client to the wireless client. The analysis of inbound traffic leads to some interesting results. Based on the results for the baseline situation, shown in Figure 6, it can be assumed that without any VPNs the interface is capable of processing inbound traffic with minimal losses.

This can be taken as an indication that the interface is also able to process all the outbound traffic. If that is the case, the high level of packet loss depicted in Figure 5 needs to be investigated. This may be qualitatively explained as follows. As a result of the dropping of UDP packets due to congestion and buffer overflows before reaching the interface, it may eventually end up processing less outbound data than what was originally generated. A side observation from this is that, the interface is not performing at its peak when processing the outbound traffic. In fact, this was also verified by analysing some dump files captured using the Windump (the Windows version of TCP Dump) utility.

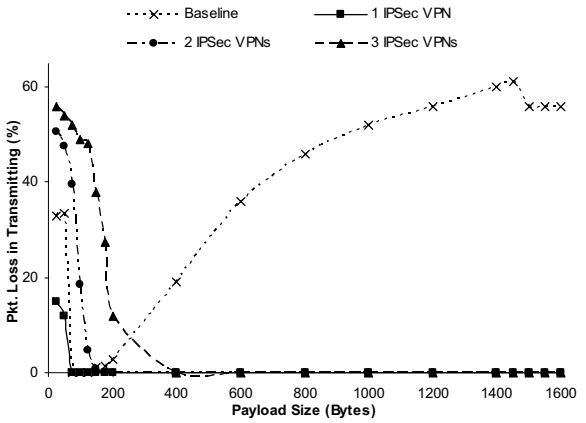

Figure 5. Packet Loss in Transmitting Traffic Generated at 1ms Inter-packet Delay.

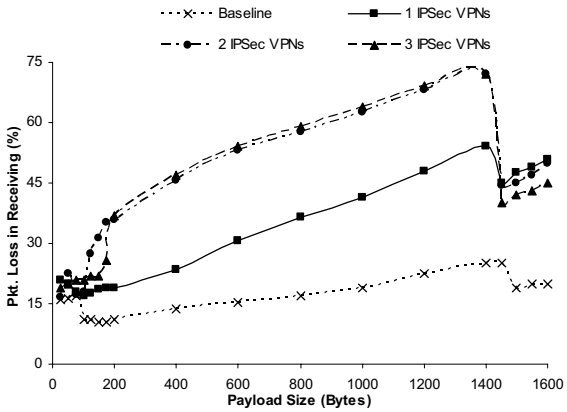

Figure 6. Packet Loss in Receiving Traffic Generated at 1ms Inter-packet Delay.

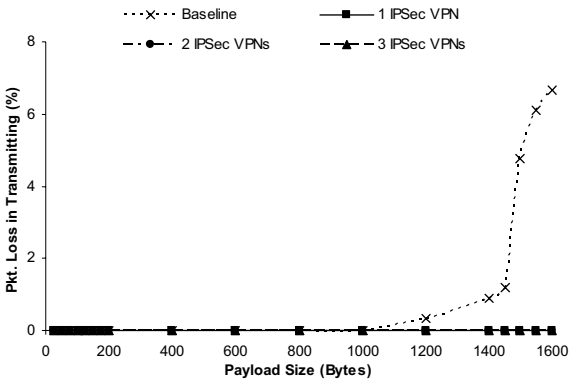

Figure 7. Packet Loss in Transmitting Traffic Generated at 5ms Inter-packet Delay.

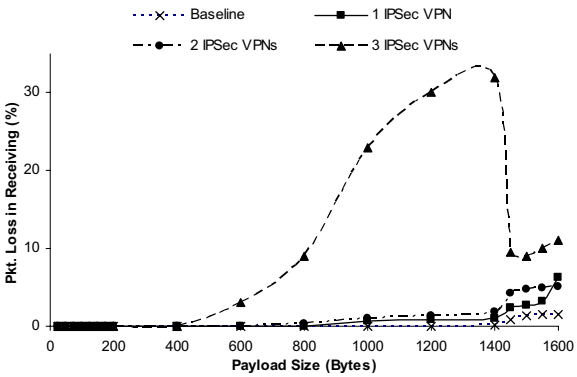

Figure 8. Packet Loss in Receiving Traffic Generated at 5ms Inter-packet Delay.

As discussed previously, Figure 7 shows a minimal outbound packet loss during transmission. However, in the case of inbound traffic, Figure 8 indicates that one of the VPN implementations is experiencing a significant growth in packet loss. This happens to be the implementation of 3 simultaneous IPSec VPNs. For payload data over 600 bytes, these VPNs experienced increasing packet losses. Furthermore, referring back to Figure 4, it can be noted that for payload sizes over 600 bytes, the same VPN setup experienced a reduction in the rate of change (increase) in throughput. Therefore, as per the argument given in section 4.1, it can be assumed that the above setup has reached its performance limits.

Finally, all packet loss results shown in Figure 5 to Figure 8 indicate sudden, but temporary, changes at the point of fragmentation similar to the results in Figure 2 and Figure 4. This can be attributed to the delay and overheads introduced at the network layer as a result of fragmentation. This will reasonably slow down the UDP flow, resulting in temporary drop in the packet loss.

4.3. CPU utilisation

The results of average CPU utilisation at the wireless client are shown in Figures 9 and 10. It is evident that a considerable number of CPU cycles are necessary for the functioning and implementation of even a single IPSec VPN.

As the number of simultaneous VPN implementations is increased, the CPU utilisation further increases. The highest average CPU utilisation rate shown in Figure 9 is 99.2%. However, Figure 10 shows 100% CPU utilisation for payloads of 1200 and 1400 bytes when 3 VPNs operate simultaneously. It also shows an average CPU utilisation of 98% and over for all UDP traffic with payloads of more than 600 bytes. From this point onwards, the graph indicates that the CPU is in full utilisation. This can be classified as the CPU having reached a saturation point as predicted in [3]. The behaviour shown by the graphs for 3 simultaneous VPNs in Figure 4 and Figure 8 clearly illustrate the overall VPN performance degradation, as a result of the CPU reaching its processing limits.

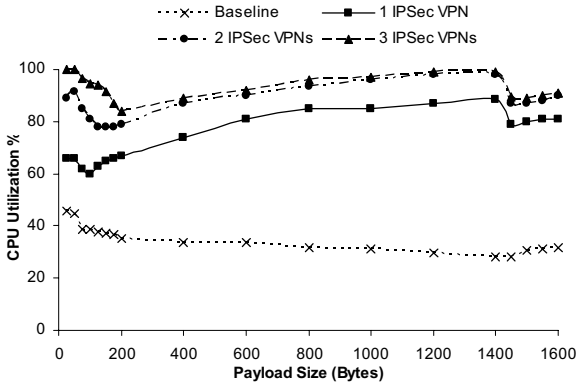

Figure 9. CPU Utilisation for Traffic Generated at 1ms Inter-packet Delay.

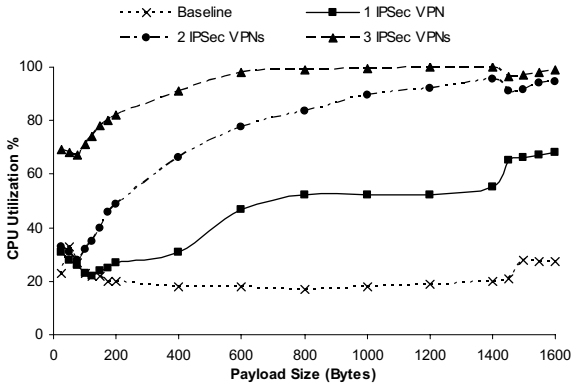

Figure 10. CPU Utilisation for Traffic Generated at 5ms Inter-packet Delay.

The graphs in Figure 9 and 10 also show relatively high CPU utilisation rates for lighter UDP payloads. The bottleneck situation discussed in section 4.2 can be considered as the primary cause of that in this situation, as well. Again, at the point of fragmentation, the CPU utilisation for the IPSec VPNs in Figure 9 and Figure 10 (except for the baseline and single VPN graphs in Figure 10) shows a momentary drop. As discussed in section 4.2, fragmentation and the extra overheads of the IPSec VPN at the network layer help to relatively slow down the UDP flow. This temporarily reduces the queuing at the interface and as a result, the average CPU utilisation reduces.

4.4. Round-trip delay

Figure 11 and Figure 13 represent the round-trip delay curves for UDP traffic with inter-packet delays of 1ms and 5ms respectively. The above graphs show that, at faster packet generation rates, the average time for a given UDP datagram to complete a round-trip is relatively longer. As mentioned in section 4.2, at faster packet generation rates, there is a relatively larger queue forming at the interface. Therefore, an average packet usually experiences a higher amount of queuing delay before it is actually transmitted by the interface. Thus, relatively high round-trip delays can be noted for VPNs in Figure 11 in comparison to those shown in Figure 12.

Figure 11 also shows a sudden increase in the round-trip delay for UDP packets with light payloads at higher packet generation rates. This is related to the bottleneck situation described in section 4.2. Figures 11 and 12 show significant jumps in the round-trip delay graphs beyond point of fragmentation. This is mainly due to the variation in queue lengths as a result of the overheads of fragmentation.

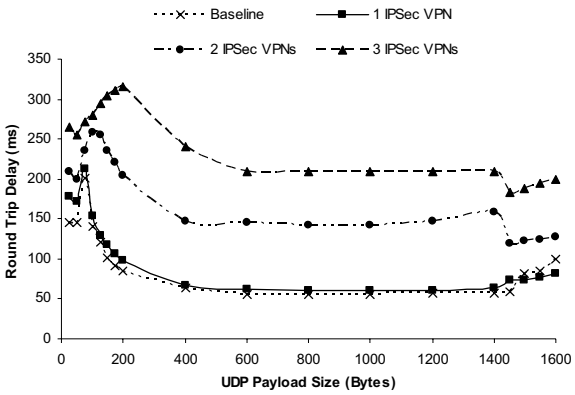

Figure 11. Round-Trip Delay for Traffic Generated at 1ms Inter-packet Delay.

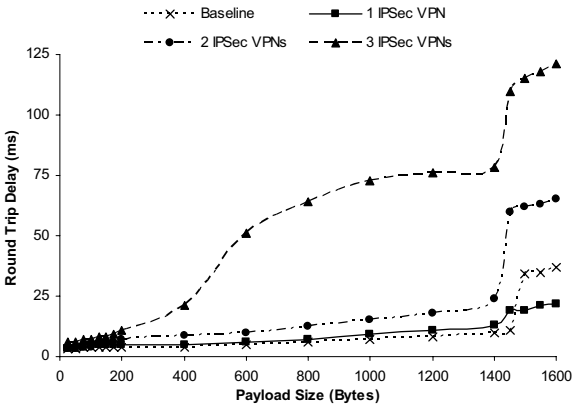

Figure 12. Round-Trip Delay for Traffic Generated at 5ms Inter-packet Delay.

5. Performance Limitations and Causes

This section investigates some of the performance limiting factors of a wireless IPSec VPN. Section 4.1 clearly establishes the fact that the effective net throughput depends on the packet generation rate. These results also show that the throughput increases with the payload data size and the maximum achievable throughput is reached prior to fragmentation.

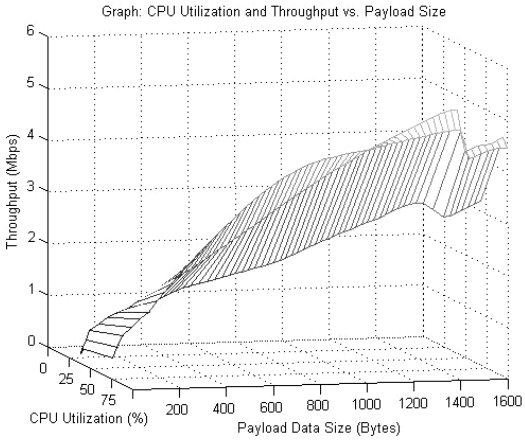

Figure 13. Variation of Throughput and CPU Utilisation against Payload Data Size.

Figure 4 indicates that when simultaneously operating multiple VPNs are increased beyond a certain point, the rate of increase in throughput against the payload data size drops gradually. The analysis of the CPU utilisation also shows that the CPU is fully exhausted at this stage. Figure 13 illustrates the effect of the CPU on the throughput of simultaneously operating IPSec VPNs. The 3D mesh graph represents the three throughput curves; a single, two and three

simultaneous IPSec VPNs respectively. As the CPU approaches full utilisation (i.e., close to 100 %), the rate of increase in throughput shows a noticeable reduction. Therefore, it is clear that the CPU is a major contributing factor to the throughput performance of an IPSec VPN. It also indicates that as the number of simultaneously operating tunnels are increased, the per tunnel throughput performance degrades.

The analysis on packet loss shows that the payload data size and the inter-packet generation rate of a data flow may contribute to the degree of packet loss. The CPU reveals another potential cause for packet loss. The packet loss graph in Figure 8 shows an exceptional increase in packet loss for 3 IPSec VPNs. An investigation into the CPU utilisation graph (Figure 10) shows that the CPU is fully exhausted at this point.

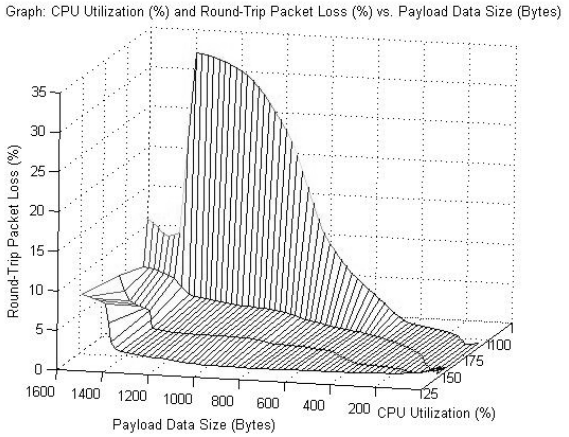

Figure 14. Variation of Packet Loss and CPU Utilisation against Payload Data Size.

Thus due to the overheads of the increasing numbers of VPNs the CPU becomes over exhausted resulting in a sudden increase in packet loss. Figure 14 illustrates the effect of CPU on packet loss of simultaneously operating IPSec VPNs. The 3D mesh graph represents the three packet loss curves; a single, two and three simultaneous IPSec VPNs respectively. As the CPU approaches full utilisation (i.e., close to 100 %), the packet loss shows a noticeable increase. Therefore, it is clear that the CPU may eventually become a major limiting factor to the packet loss of an IPSec VPN.

6. Conclusions

In this paper, the analysis and experimental results for an evaluation of the QoS of VPNs over an IEEE 802.11b wireless network are presented. The analysis clearly establishes the limitations and causes affecting the performance of VPNs. Some of such limitations are related to data flow arising from high packet generation rates, relatively small or large payload data sizes and the CPU processing power. Due to heavy overheads of maintaining multiple VPNs, larger payload data sizes may lead to a reduction in the QoS levels. As the multiple VPNs eventually reach their limitations, reductions in the rate of change (increase) in throughput and increases in packet losses are noted. Beyond this point the CPU indicates a fully utilized state. Hence, the CPU can be ruled out to be one of the primary limiting factors in affecting the QoS of simultaneously operating multiple VPNs. The results also indicate how optimal QoS levels can be achieved by regulating the above-mentioned parameters for a wireless VPN.

7. References

[1] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," Wireless Communications, IEEE, vol. 9, 2002, pp. 44-51.

[2] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in Proc. 7th Int. Conf. Mobile Computing and Networking: ACM Press, 2001, pp. 180-189.

[3] K. S. Munasinghe and S. A. Shahrestani, "Evaluation of an IPSec VPN over a Wireless Infrastructure," Proc. Australian Telecommunication Networks and Applications Conf., 2004, pp. 315-320.

[4] S. Kent and R. Atkinson, "IP Authentication Header," in IETF: RFC 2402, 1998.

[5] S. Kent and R. Atkinson, "IP Encapsulating Payload," in IETF: RFC 2406, 1998.

[6] S. Ci and H. Sharif, "An link adaptation scheme for improving throughput in the IEEE 802.11 wireless LAN," Proc. 27th IEEE Conf. LCN, 2002, pp. 205-208..

[7] M. G. Arranz, R. Aguero, L. Munoz, and P. Mahonen, "Behavior of UDP-based applications over IEEE 802.11 wireless networks," Proc. 12th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications, vol. 2, 2001, pp. F72-F77.

[8] B. Bing, "Measured performance of the IEEE 802.11 wireless LAN," Proc. 23rd IEEE Conf. LCN, 1999, pp. 34-42.

[9] A. Kamerman and G. Aben, "Throughput performance of wireless LANs operating at 2.4 and 5 GHz," Proc. IEEE PIMRC, vol. 2, 2000, pp. 190-195.

[10] G. Xylomenos and G. C. Polyzos, "TCP and UDP performance over a wireless LAN," Proc. IEEE INFOCOM, vol. 2, 1999, pp. 439-446

[11] J. Wu and J. Ilow, "A wireless multimedia LAN testbed," Proc. Conf. Electrical and Computer Engineering, vol. 2, 2000, pp. 826-830.

[12] J. Lee, G. Kim, and S. Park, "Optimum UDP packet sizes in ad hoc networks," Proc. Merging Optical and IP Technologies workshop on High Performance Switching and Routing, 2002, pp. 214-218.

[13] M. Petrovic and M. Aboelaze, "Performance of TCP/UDP under ad hoc IEEE802.11," Proc. 10th Int. Conf. ICT, vol. 1, 2003, pp. 700-708.