International Journal of Advanced Science and Technology
Vol. 46, September, 2012

Impact of Duplicated ACK on TCP Performances in Wireless Chain Environment

Toni Janevski[1] and Ivan Petrov[2]

[1]Ss. Cyril and Methodius University in Skopje, Faculty of Electrical Engineering and Information Technologies, Karpos 2 bb, 1000 Skopje, Macedonia,
[2]Makedonski Telekom, Orce Nikolov bb, 1000 Skopje, Macedonia,
tonij@feit.ukim.edu.mk, Ivan.Petrov@telekom.mk

Abstract

The availability of smart hand held devices increases rapidly and the need for their interconnection is demand that can not wait. IEEE wireless standards are one of the leading factors of the wireless revolution. All IP networks with centralized management are reality. IMS platforms are providing advanced IP based voice and multimedia services. Technical standards are currently defined that address both the methods of wireless/wired interconnection. TCP remains as the most important and most used transport protocol used for non real time services such as web, email, ftp, which are basis for additional value-added or novel IP-based services. Not much work is done in direction of understanding how can TCP be adapted to handle this scenario for efficiency delivering digital contents. The focus should continue to be on the transport protocol development, its optimization and predetermined behavior in variety of environments. Hence, in this paper we conduct detailed analysis of the impact of the Duplicated ACK and Delayed ACK parameters on the throughput, congestion window change and average packet delay in wireless chain scenario with incorporated loss models under traffic load provided by different TCP versions and IFQ buffer size.

Keywords: *DelACK, DupACK, Internet, TCP, Throughput, Wireless Network*

1. Introduction

The rapid development of Internet and wireless technologies resulted in their integration. In that manner all IEEE wireless networks are IP native, i.e. they define physical and Medium Access Control (MAC) layers, while the network layer is reserved for IP. On one side Internet is based on TCP/IP protocol suite targeted for usage by non-real-time applications (e.g., web, ftp, email etc.) and UDP for real-time applications (e.g., voice over IP, streaming, etc.) From the beginning the transport protocols were designed and optimized for wired world where most of the losses and packet delays are caused by congestion. The first algorithms incorporated simple mechanisms like Go back N. With the time and the growth of the traffic the need for design optimization of the transport protocols was born and the first steps were made [1]. The need for congestion control, reliability and good resources utilization were required to provide best-effort Internet, with congestion control left to the end nodes (terminals and servers). Since then, there are several versions of TCP which have been widely implemented. Starting from old Tahoe [2], then continuing with Reno [3] and further NewReno, which was followed by Vegas, Sack [4-6], until today when we have huge number of different TCP versions that improve the behavior either in wired or in wireless environment. Hence, every parameter used for the TCP design has important role for the efficiency of the transport protocol. Such design parameters make the transport protocol robust at different environments. For sure every year we will have more and more efficient (more or less) modification of the transport protocols because the initial condition, under which the TCP protocol was born, has changed [7-13]. This is also the case for network access protocols (below the network layer). The behavior of the transport protocols is not the same in wired and wireless environments because of the different impact on the access control and environment parameters where losses caused in these environments have different nature. In this paper we focus on DupACK parameter in wireless environment which directly influences the congestion control mechanism in TCP versions. This paper is organized as follows. Section two gives brief overview of the transport protocols, discusses some related work and motivates the need for our approach. Section 3 describes our simulation environment and section 4 presents the simulation results. Section 5 concludes the paper.

2. The Transport Protocols

Internet is based on several factors and one of them is the ability to provide a reliable medium for file downloading and information sharing. TCP was initially designed to provide end to end, connection oriented and reliable service in ARPANET and later in the Internet. The first TCP implementations were using cumulative positive acknowledgements and required a retransmission timer expiration to send a lost data during the transport. They were following the go back n model in order to provide good user throughput. Today evolved versions of TCP contain variety of mechanisms that control the network congestion and maintain good user throughput. TCP continuously probes the link for higher transfer rates, eventually queuing packets in the buffers associated with the bottleneck connection. The network resources can be shared by several users at the same time, and TCP provides equal sharing of the given link between different TCP connections on longer time scales.

There are three basic phases that describe the behavior of the TCP protocols. Slow start is the initial phase (smallest congestion window), which is followed by congestion detection and congestion avoidance phase. One of the main TCP objectives is to create a reliable connection by retransmitting lost packets, for this purpose TCP incorporate acknowledgements (ACKs). The objectives of the acknowledgements are to regulate the transmission rate of the TCP ensuring that packets can be transmitted only when other have left the network and to render the connection reliable by transmitting to the source information if it needs so as to retransmit packets that have not reached the destination. So, the ACKs have major role in the behavior of the TCP protocol. A TCP packet is considered as lost if three repeated ACKs for the same packets arrive at the source or when there is a timer expiration meaning that ACK does not arrive in a given time period. Retransmitting after three duplicated ACKs is called fast retransmit. Further in this paper we will do analysis regarding the number of ACKs, i.e. what happens if we change the number of ACKs that triggers the fast retransmit phase of TCP, using different TCP versions under wireless environment which is not TCP native. In this environment TCP will be impacted by interference, errors, fading which may cause packet losses that are handled by MAC protocol. These local MAC retransmission hide the error losses from TCP and are increasing the reliability of the connection on one hand but on the other they are followed by the well known back off mechanism that increase the delay packet time. Wireless connection can be shared and this also impacts the file transfer by increasing the congestion level and causing queuing. IFQ buffer sizes has important rule so the large buffers helps the TCP based flows in keeping a high sending rate and the main reasons are: link successive to the buffer remains fully utilized for longer periods of time since there are always packets in queue waiting to be sent and the traffic burst can be easily accommodated thus reducing the packet losses and maintaining higher sending rates for longer periods. All of these parameters strongly influence the TCP behavior and performances so they should be carefully analyzed and predetermined.

3. Simulation Scenario

We will analyze two different chain scenarios the first presented in Figure 1 consisted of two nodes and the second presented in Figure 2 consisted of nine wirelessly linked nodes, (named n0-n8). The distance between the nodes is set at 200 m. We are using 802.11 g standard at MAC/Physical layer that limits our max link throughput at 54 Mbps. Configuration of the links between the nodes is given in Table 1. Two Ray Ground model [14] will be used in this scenario.

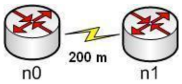

Figure 1. Simulation Scenario A Consisted of Two Nodes

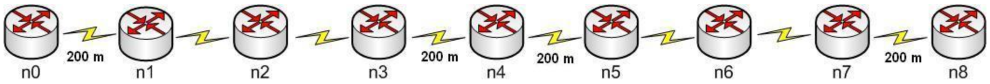

Figure 2. Simulation Scenario B Consisted of Nine Nodes

Table 1. Network Configuration

Link	**Distance**	**Throughput**
n0-n1....n8	200 m	54 Mbps
Propagation Model	**MAC Protocol**	**TCP Packet Size**
Two Ray Ground	802.11 g	512 byte

Table 2. Simulation Parameters

Parameter	**Value**	**Comment**
Duplicated ACK	1,2,3,4,5,6,7,8,9,10	Default val. is set to 3
Queue size (packets)	2,5,15,25,50	Default value is Bandwidth Delay product
TCP	Tahoe, Reno, New Reno, Vegas, Sack	Common used TCP types
Del ACK (d)	1,2	Del ACK mechanism,

We observe the following TCP protocols: Tahoe, Reno, New Reno, Vegas, and Sack. Different parameter values used in this scenario are listed in Table 2.

Analysis targets the throughput with/without 20% link losses. This throughput degradation is cumulative because every node incorporates degradation of the incoming and outgoing traffic. Simulation duration time is set at 150 seconds. TCP packet size is 512 bytes, while ACK packets have their well known value. NS2 is used to simulate the requested environments.

4. Analysis of Transport Control Protocols

In the following part we observe simulation results from the scenario presented in Figure 1 and Figure 2.

4.1 Two Node Wireless Scenario

Like we stated before in this case we analyze the throughput performance of the network scenario presented in Figure 1. Figure 3 shows the change of the throughput as a function of the number of duplicated ACK and the value of the IFQ buffer size. In the real world the default value of DupACK parameter is set at three but this value is optimized for wired environment. It is quite vivid Figure from which we can see overlapping of the throughput lines for most of the transport protocols and IFQ values. There is small variation and increment until DupACK receive value 4. For values grater than 4 the overlapping is obvious. Best throughput is achieved with Vegas regardless the IFQ value except when the DupACK receive value 4 and IFQ is set at 5 packets when is achieved the lowest throughput. If we average the throughput for all DupACK and IFQ values then the rank list will be: NewReno, Sack, Tahoe, Vegas and Reno. When IFQ is set at 25, 50 packets there is no variation of the throughput.

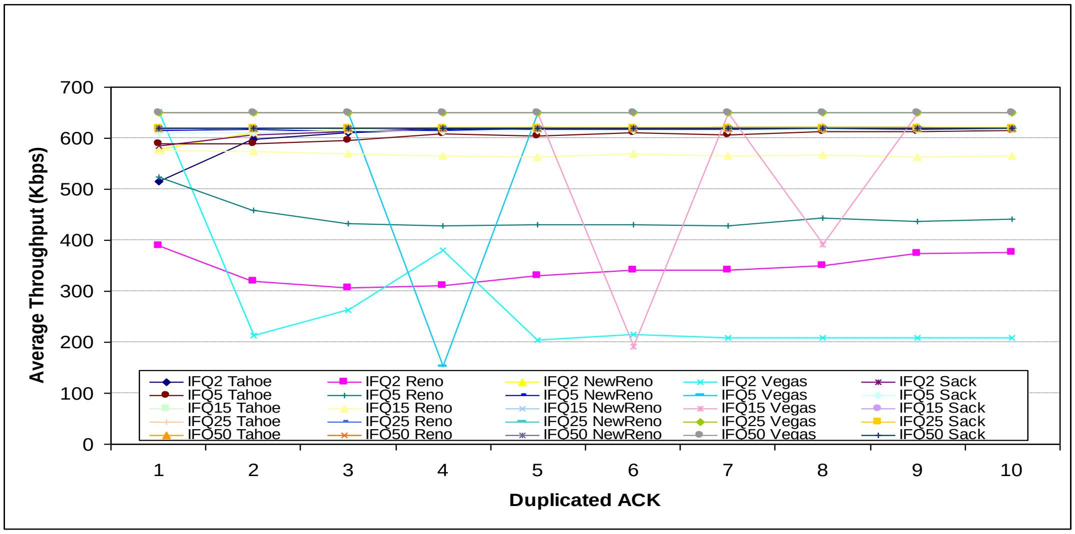

Figure 3. TCP throughput variation for different values of the duplicated ACK parameter with excluded losses

This is not the case when it receives values of 2, 5, 15 packets when we have increment of the throughput for all transport protocols. If we analyze the number of DupACK then the throughput receives larger value than in case when DupACK is set at three when DupACK is set at 6,8,9,10,4,7. The impact of the distance and the propagation model is obvious so the overall average achieved throughput is around 600 Kbps. This can be explained with the impact of the distance at the propagation parameters and the change of the modulation scheme of the IEEE 802.11 g protocol and the nature of the MAC layer [9], all of this parameters strongly influence at the change of the cwnd, that directly impacts the throughput value. Exceptions are the throughput lines when Reno is the used as transport protocol for IFQ values of 2, 5 and 15 packets. Figure 4 presents the change of the throughput when we have incorporated 20% losses. In this case the change is more colorful. The degradation of the throughput is obvious, it is around 200 Kbps. If we compare the overall average values of the throughput then Vegas will be at first place followed by NewReno, Sack, Tahoe and Reno. The throughput receives the same values when IFQ is set at values of 25 and 50 packets. When IFQ is set at 2, 5 packets there is increment of the throughput but for value of 15 packets the throughput degradation is obvious. DupACK should be as follows: 1, 5, 7, 2, 6, 9, 10, 8. Instead to acknowledge every packed there is a mechanism that ACKs every second sent packet. This is a way to reduce the reverse traffic in the network and to increase the throughput.

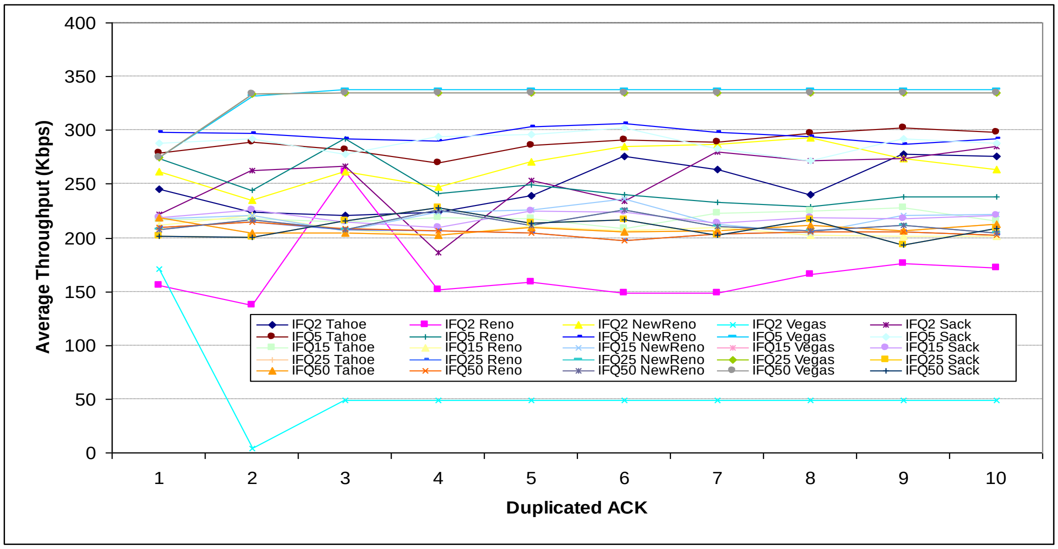

Figure 4. TCP Throughput variation for different values of the duplicated ACK parameter in case of 20 % losses

Figure 5 presents the traffic variation when the DelACK mechanism is activated and set at 2. In this case the best average throughput is achieved with Vegas second best is Tahoe, NewReno, Sack, and Reno. When IFQ is set at 25 and 50 packets there is no variations of the throughput, this comment stands for Vegas when IFQ is set at 15 packets. The highest throughput is achieved with Vegas except when IFQ is set at 2 when it shows worst performance. If we increase the value of IFQ 2, 5, 15 the throughput increase regardless the transport protocol. Better throughput values are achieved when DupACK is set at 1 and 2.

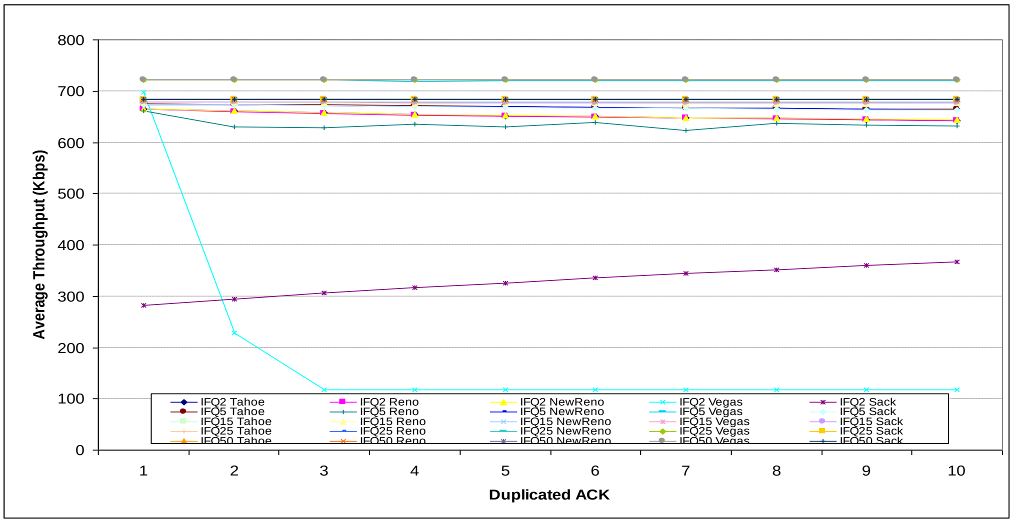

Figure 5. TCP throughput variation for different values of duplicated ACKs and 0% losses. The Number of delayed ACKs is set at 2

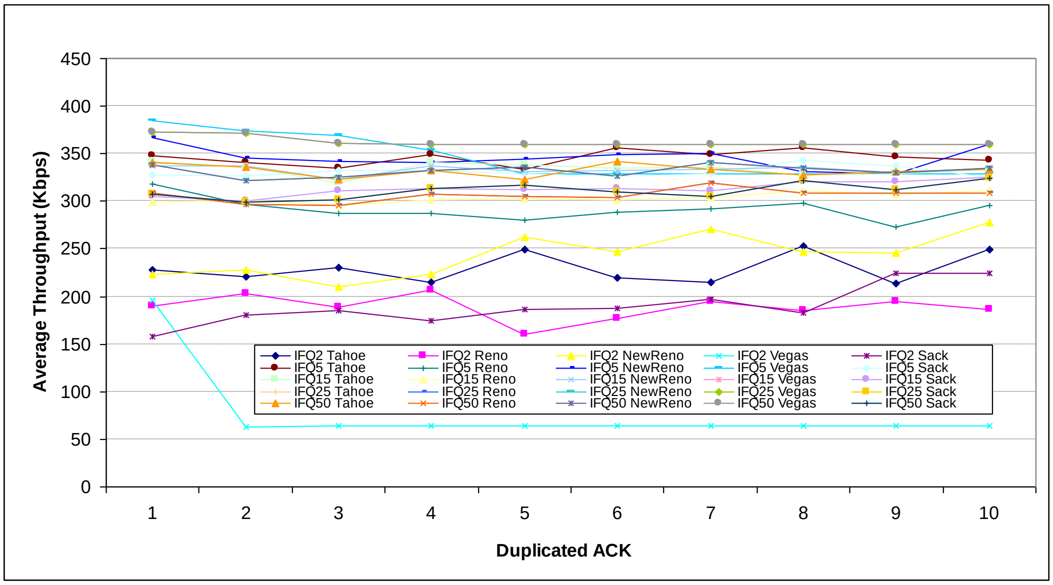

Figure 6. TCP throughput for different values of the number of duplicated ACKs and 20% of losses. The Number of delayed ACKs is set at 2

Variation of the throughput with 20% losses and activated DelACK mechanism is presented in Figure 6. In this case the best average throughput is achieved with Tahoe, followed by Vegas and NewReno at the second position, Sack is at the third and Reno at forth position. DupACK can be set at 1, 7, 10, 6, 8, 9, 5, and 4. For IFQ 25, 50 the throughput receive the same values. For IFQ 2, 5 there is an increment but when IFQ is set at 15 there is small decrease. It is obvious that all of these parameters strongly influence at the throughput, their adjustment should be done carefully, dynamical change is recommended in order to achieve better throughput.

After we have conducted detailed analysis of the average throughput we will observe the change of CWND as function of IFQ and DelACK parameters. DupACK parameter in this case will be fixed at value of three. This analysis may be conducted for all DupACK in this case it will be done for the default DupACK value.

Starting from Figure 7 where is presented the cwnd change for all observed protocols and IFQ values of 25, 50 packets in lossless system, иt can be concluded that cwnd increases with time. This explains the constant nature of the throughput so the max throughput is obtained. The same comment stands for Figure 11 when is activated the DelACK mechanism. There is degradation of the throughput from the predefined value because of the impact of the MAC layer, the wireless medium and the distances of the nodes (modulation schemes). CWND behavior is completely different when IFQ is set at 15 packets. In this case we can notice that TCP goes over all three phases. If we compare Figures 8 and 12 than we can say that they differ in the max value that cwnd receives. In the first case it is around 25 packets and the phase duration is longer so the window is wider that provides better throughput environment.

In the second case the max value is around 16 packets, it is noticeable that phases durations are shorter but this mechanism provides similar or better throughput behavior that can be seen from Figures 3 and 5. This throughput behavior can be explained with the lower cwnd variation in the second case. If we set IFQ at value of 5 packets than cwnd decreases its max and in this case is set at 10 packets or 7 packets when DelACK mechanism is activated. If we compare Figures 9 and 13 we can notice denser behavior of cwnd in case when DelACK is activated. Further decrease of IFQ value will result with decrease of the max cwnd value at 5 and 3,5 packets respectively. Like in the previous scenarios and now we can see that when DelACK is activated cwnd changes the phases more frequently and losses are grater so the throughput degradation is noticeable. If we compare the Figures 7, 8, 9, 10 we can see that as we decrease the IFQ value we have cwnd degradation of the max received value and its duration is shorter. TCP changes the phases more frequently in order to provide optimized throughput. The degradation is highest when IFQ is set at 2 which highly impact the throughput. We will continue with the cwnd analysis for the scenario with incorporated 20% of losses. In this case the cwnd behavior will be presented separately for the protocols because of the frequent phase change (congestion avoidance, slow start phase) of the TCP protocols caused by higher system losses and IFQ degradation. We will start with NewReno, Reno, Sack, Tahoe and will end with Vegas. At Figure 15 cwnd change for NewReno is presented we can notice overlapping of the cwnd curves when IFQ is set at 25 and 50 packets. When IFQ is set at 15 packets cwnd max value and the duration are decreased. Rapid decrease of cwnd can be seen at Figure 16 when IFQ is set at 5, the same comment stands when IFQ is set at 2 packets. When DelACK is incorporated the max values of the cwnd are almost similar (average value differs) like in the previous case, narrow cwnd behavior is noticeable when IFQ is set at 2 packets. Cwnd Reno has similar behavior like cwnd NewReno. Cwnd of Sack and Tahoe have similar behavior. There is cwnd overlapping when

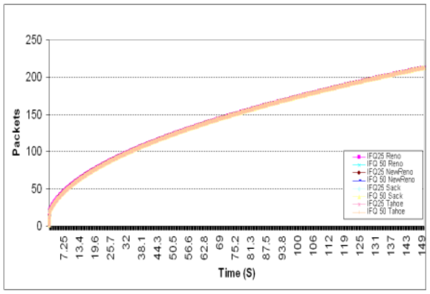

Figure 7. CWND change for 0 % of losses and IFQ values of 25, 50 packets (pkt)

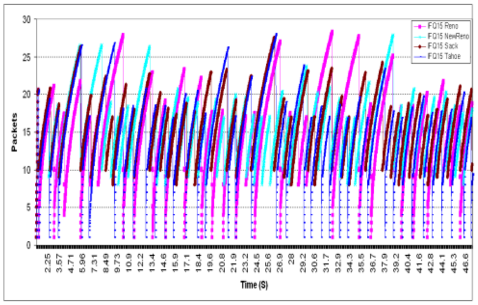

Figure 8. CWND change for 0 % of losses and IFQ value of 15 pkt

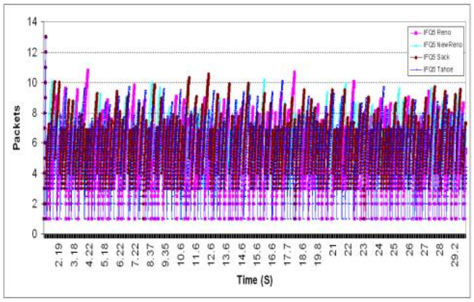

Figure 9. CWND change for 0 % of losses and IFQ value of 5 pkt

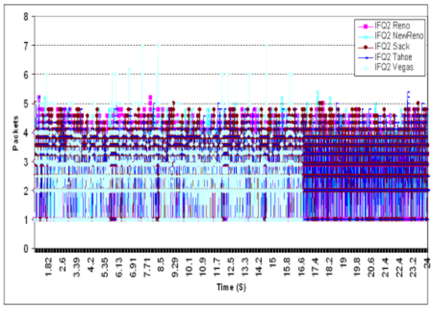

Figure 10. CWND change for 0 % of losses and IFQ value of 5 pkt

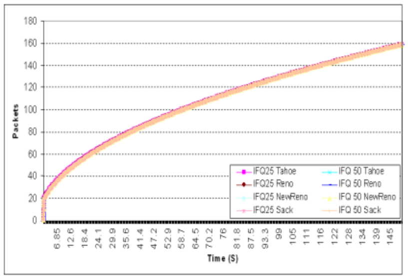

Figure 11. CWND change for DelACK on, 0 % of losses and IFQ set at 25, 50 pkt.

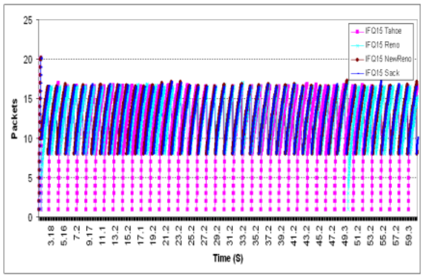

Figure 12. CWND change for DelACK on, 0 % of losses and IFQ set at 15 pkt.

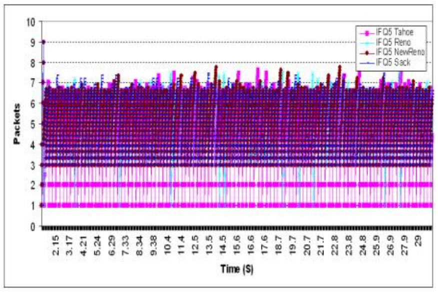

Figure 13. CWND change for DelACK on, 0 % of losses and IFQ set at 5 pkt

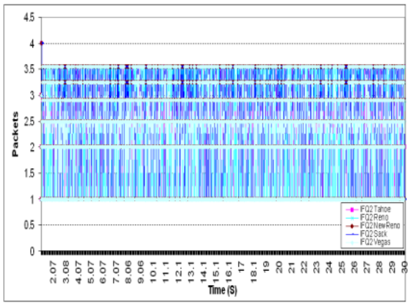

Figure 14. CWND change for DelACK on, 0 % of losses and IFQ set at 2 pkt

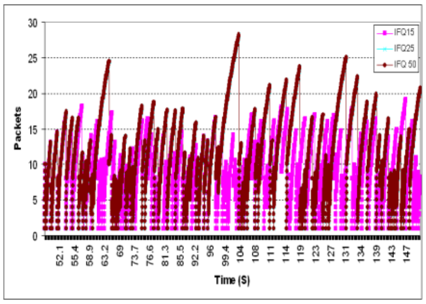

Figure 15. NewReno CWND change for 20 % of losses and IFQ set at 15,25,50 pkt.

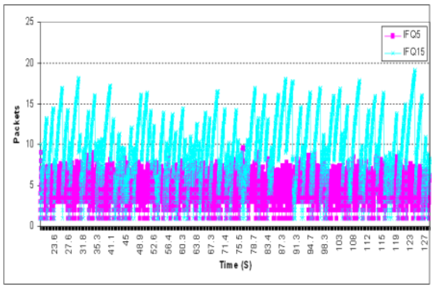

Figure 16. NewReno CWND change for 20 % of losses and IFQ values of 5, 15 packets

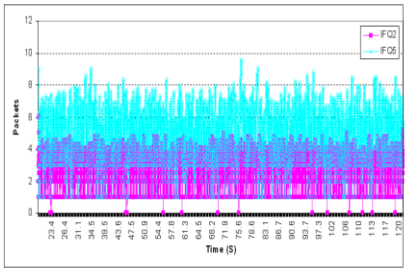

Figure 17. NewReno CWND change for 20 % of losses and IFQ values of 5, 2 pkt.

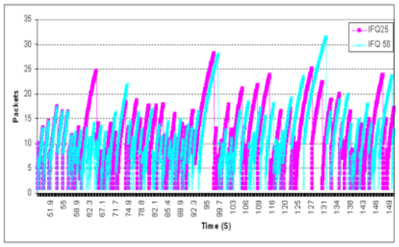

Figure 18. Reno CWND change for 20 % of losses and IFQ values of 25, 50 pkt.

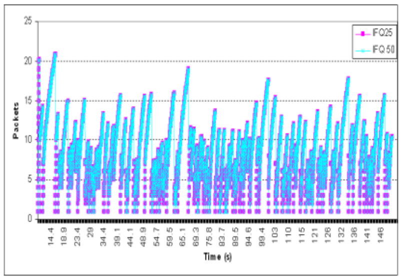

Figure 19. NewReno DelACK CWND change for 20 % of losses and IFQ set at 25, 50 pkt.

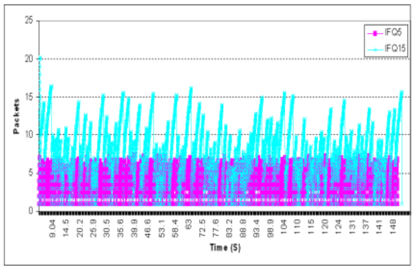

Figure 20. NewReno DelACK CWND change for 20 % of losses and IFQ set at 5, 15 pkt.

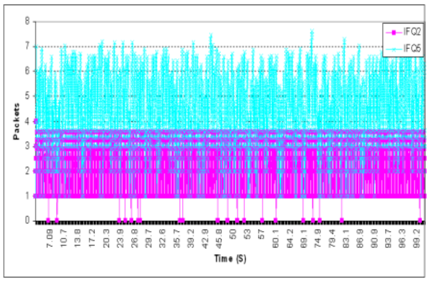

Figure 21. NewReno DelACK CWND change for 20 % of losses and IFQ set at 2, 5 pkt.

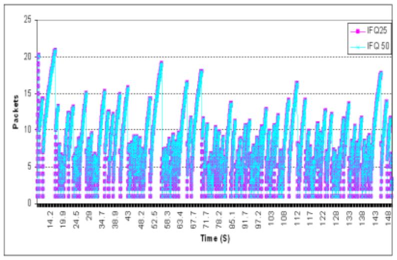

Figure 22. Reno DelACK CWND change for 20 % of losses and IFQ set at 25, 50 pkt.

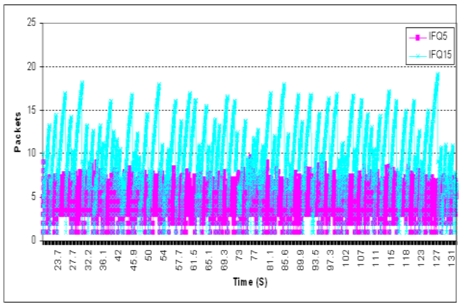

Figure 23. Reno CWND change for 20 % of losses and IFQ values of 5, 15 pkt

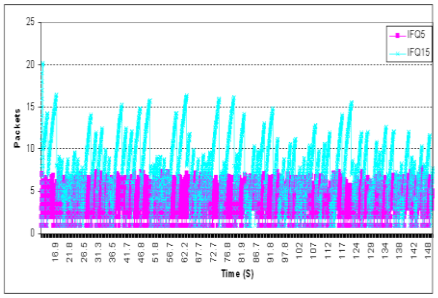

Figure 27. Reno DelACK CWND change for 20 % of losses, IFQ set at 5, 15 pkt

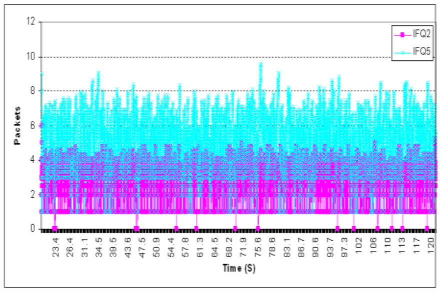

Figure 24. Reno CWND change for 20 % of losses and IFQ values of 2, 5 pkt

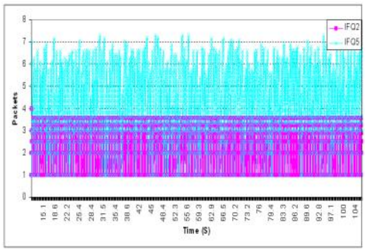

Figure 28. Reno DelACK CWND change for 20 % of losses and IFQ set at 2, 5 pkt

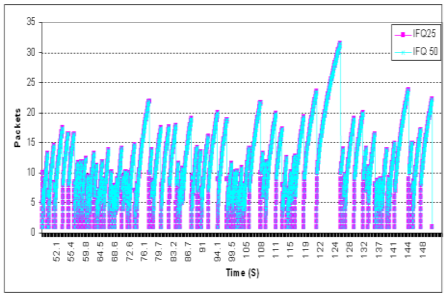

Figure 25. Sack CWND change for 20 % of losses and IFQ values of 25, 50 pkt

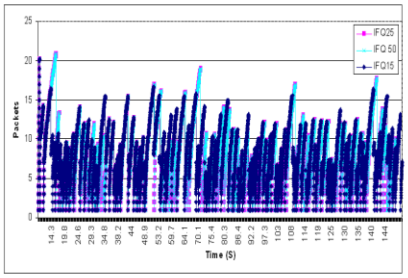

Figure 29. Sack DelACK CWND change for 20% of losses, IFQ set at 15,25,50 pkt

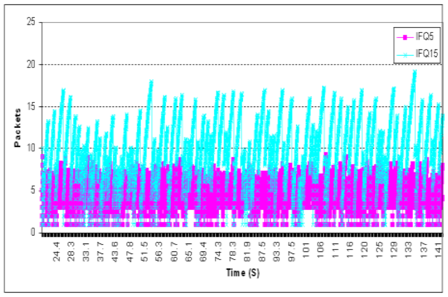

Figure 26. Sack CWND change for 20 % of losses and IFQ values of 5, 15 pkt

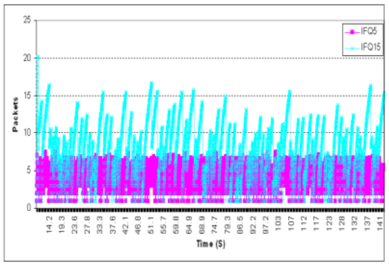

Figure 30. Sack DelACK CWND change for 20% of losses and IFQ set at 5, 15 pkt

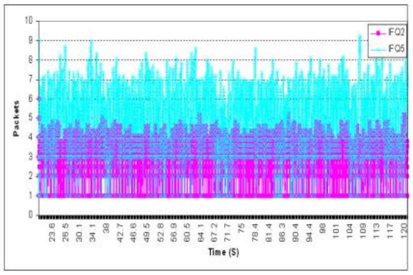

Figure 31. Sack CWND change for 20 % of losses and IFQ values of 2, 5 pkt

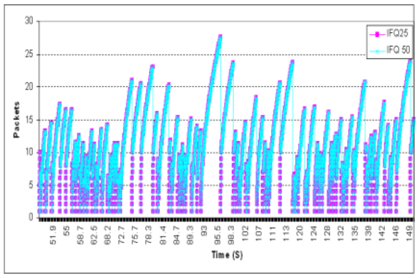

Figure 32. Tahoe CWND change for 20 % of losses and IFQ values of 25, 50 pkt

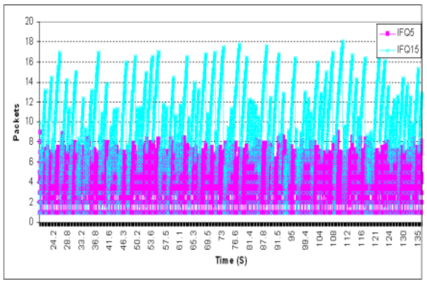

Figure 33. Tahoe CWND change for 20 % of losses and IFQ values of 5, 15 pkt

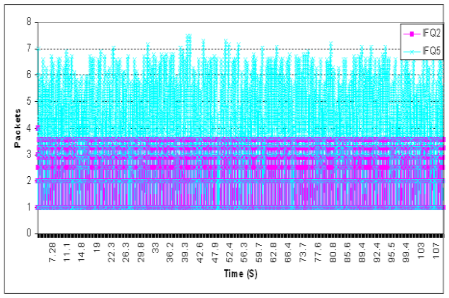

Figure 34. Tahoe CWND change for 20 % of losses and IFQ values of 2, 5 packets

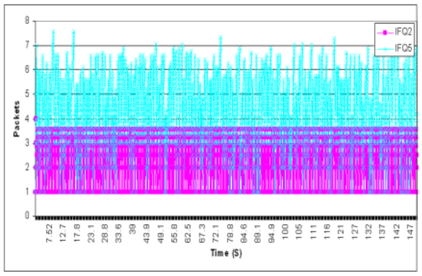

Figure 35. Sack DelACK CWND change for 20 % of losses and IFQ set at 2, 5 pkt

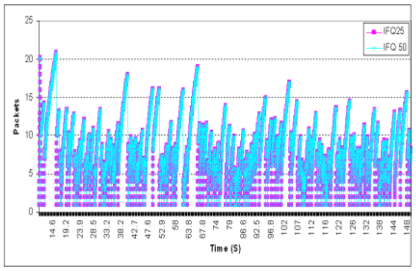

Figure 36. Tahoe DelACK CWND change for 20 % of losses and IFQ set at 25, 50 pkt

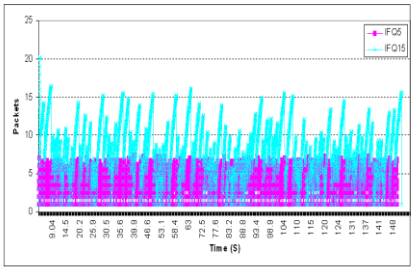

Figure 37. Tahoe DelACK CWND change for 20 % of losses and IFQ set at 5, 15 pkt

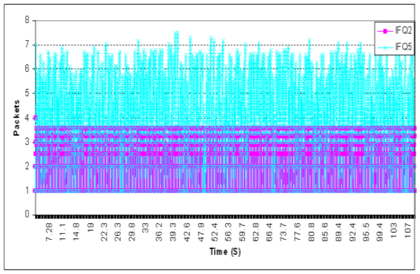

Figure 38. Tahoe DelACK CWND change for 20 % of losses and IFQ set at 2, 5 pkt

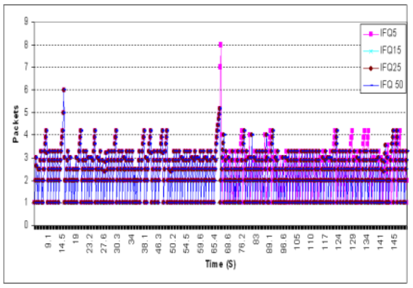

Figure 39. Vegas CWND change for 20 % of losses and IFQ set at 5, 15, 25, 50 pkt

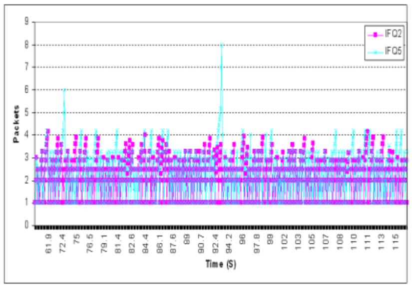

Figure 40. Vegas CWND change for 20 % of losses and IFQ values of 2, 5 pkt

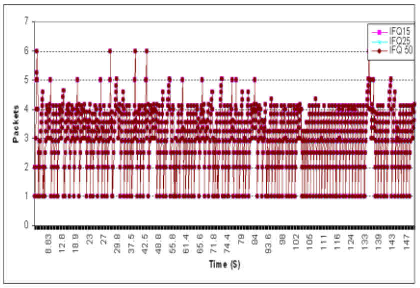

Figure 41. Vegas DelACK CWND change for 20 % of losses, IFQ set at 15,25,50 pkt

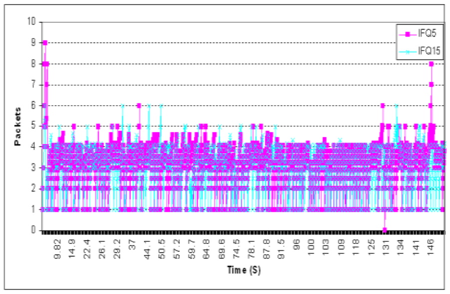

Figure 42. Vegas DelACK CWND change for 20 % of losses and IFQ set at 5, 15 pkt

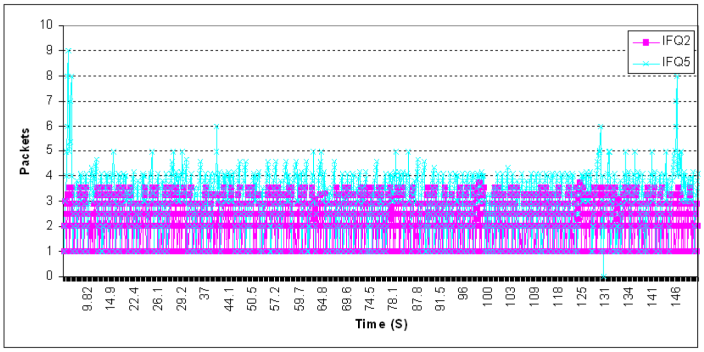

Figure 43. Vegas DelACK CWND change for 20 % of losses and IFQ values of 2, 5 packets

IFQ receives values of 25, 50 packets, max values of cwnd are identical when is activated DelACK mechanism, but the average value differs and impacts the throughput, it is in correlation with cwnd duration. Vegas has different cwnd behavior in this case there is overlapping of the cwnd curves for IFQ set at 5, 15, 25, 50 packets. When IFQ is set at 2 then cwnd has lower duration than in the previous case. This is the main reason why Vegas provides best throughput. It can be concluded that frequent phase changing can lead to better throughput or throughput optimization this is confirmed by the cwnd change when DelACK mechanism is activated in this case cwnd shows dense behavior. The analysis will be incomplete if we don't include the average packet delay. At Figure 44 it is presented the average packet delay from where we can see that lowest delay we have when IFQ is set at 2 packets, if we increase the IFQ value the delay will increase too, we have overlapping of the curves when IFQ receive values of 25 and 50 packets in that case the delay is highest. This comment stands for the Figures 45, 46, 47. It is important to be mentioned that Vegas provides lowest packet delay. DelACK mechanism under losses provides better packet delay than in case with no losses.

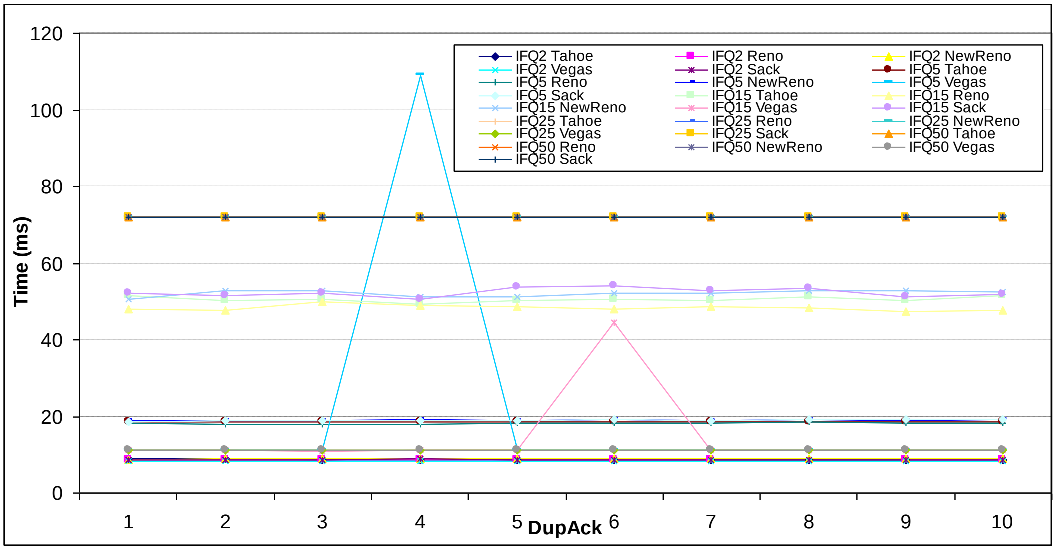

Figure 44. Average Delay for 0 % of losses and IFQ values of 2, 5, 15,25, 50 packets

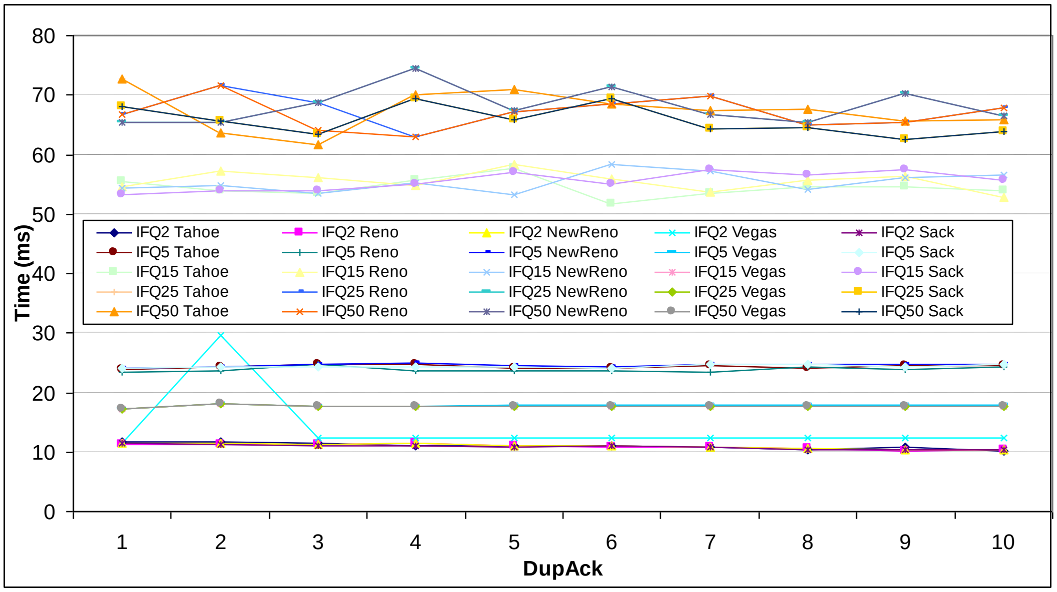

Figure 45. Average Delay for 20 % of losses and IFQ values of 2, 5, 15,25, 50 packets

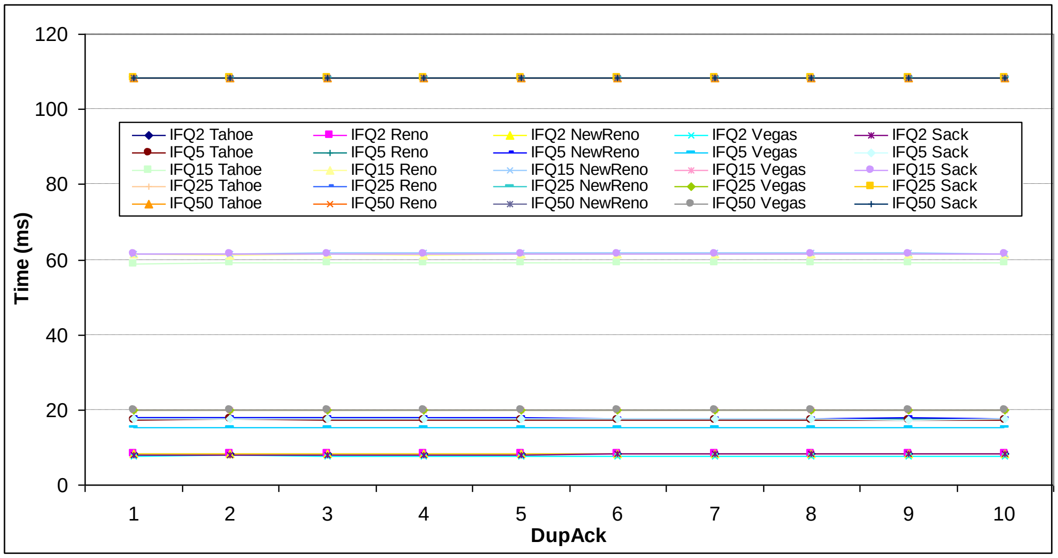

Figure 46. Average Delay for 0 % of losses, IFQ values of 2, 5, 15,25, 50 packets, and DelACK activated

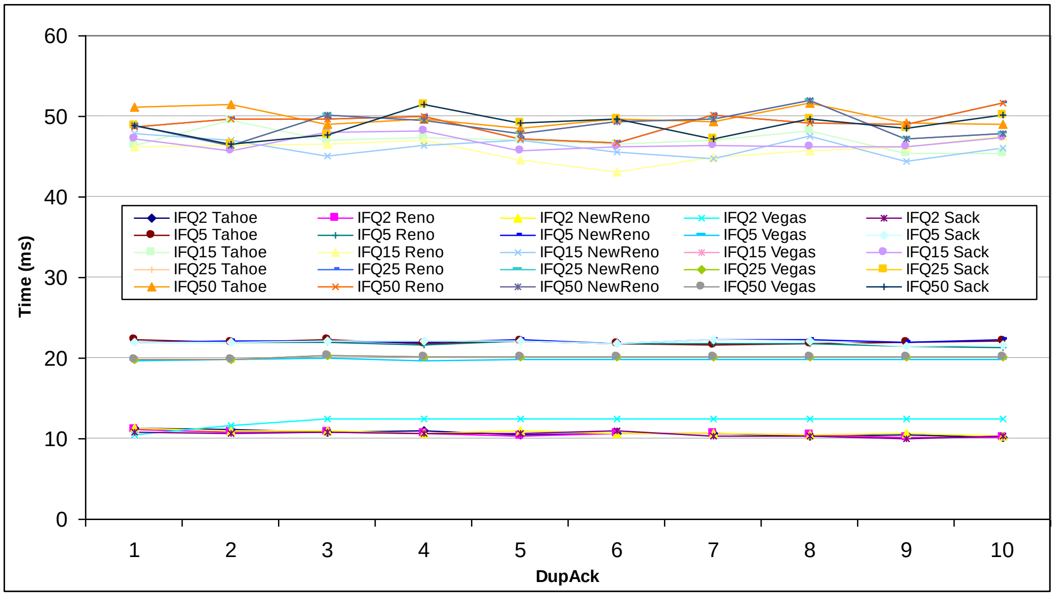

Figure 47. Average Delay for 20 % of losses, IFQ values of 2, 5, 15,25, 50 packets, and DelACK Activated

4.2 Nine Node Wireless Chain Scenario

From Figure 48 we can see that when IFQ receive values of 15,25,50 packets there is overlapping of the throughput lines for the used protocols, this is not the case when we use smaller values. Best throughput is obtained when Vegas is used as TCP protocol, second best is NewReno then is followed by Sack Tahoe and Reno when IFQ is set at 15,25,50 packets. Better throughput is achieved if we set DupACK at values of 6,9,1,2, except when we use Tahoe as a transport protocol. If we decrease the value of IFQ size at 5 packets than Sack and NewReno will switch their positions this comment stands for Reno and Tahoe the difference is that better throughput is achieved when DupACK is set at 6 and 9. If we set IFQ at value of 2 then Reno is the leading protocol which is followed by Sack, NewReno, Vegas and Tahoe. In this environment the best throughput values are achieved when DupACK receive values of 7,9,10. At Figure 49 we can see the change of the throughput when 20% losses are included. Like in the previous analysis and here there is an overlapping of the throughput lines for larger value of IFQ parameter. In this case the performances obtained with the TCP protocols are similar like when there are no losses so at the first two places there is no change, the third place is reserved for Reno followed by Sack and Tahoe. Improved throughput values are obtained when DupACK is set at 8, 7, and 9. For IFQ set at 5 we have similar behavior like when there are no losses first two places are reserved for Vegas and Sack then NewReno and Reno has switched their positions and the last one is reserved for Tahoe. The best throughput is achieved when DupACK is set at 8 and 10. Drastically new situation we have when IFQ receive value of 2, than the protocol ranking is as follows Vegas Reno Tahoe NewReno and Sack. The DupACK influence is greater when it receives values of 6, 8, 9, and 10. Obtained throughput in the scenario with no losses and activated DelACK mechanism is shown in Figure 50. From this Figure we can notice almost 100% improvement of the throughput than in the case when DelACK mechanism was excluded. There is no change of the statement that stands for larger IFQ values, however the overlapping of the curves is obvious. Protocol ranking is as follows Vegas, Sack, Reno, NewReno and Tahoe. DupACK should be set at 1,6,7,9. If we decrease IFQ value then we have status quo at the first two positions Tahoe is third followed by NewReno and Reno. In this case DupACK should be set at 5. Rapid change of the transport protocol behavior we have when IFQ is set at value of 2 packets. Winning combination is Sack, NewReno, Reno, Tahoe, Vegas, and DupACK should receive value of 1 or 2.

If we incorporate 20 % of losses in this scenario we obtain results presented in Figure 51. In this case we have different behavior of the protocols so the change of the ranking list is obvious Vegas, Reno NewReno, Sack and Tahoe. DupACK should be set at 2 or 7. If we decrease IFQ value at 5 then there is no change at the first three positions. Tahoe and Sack has changed their positions and now Sack has shown worst performances. Significant change is noticeable if we set IFQ at value of 2 packets. NewReno Tahoe Vegas Sack and Reno is the new rang list in this case maybe DupACK could be set at 3, 6 or 9. Other parameter that has major rule of traffic shaping is the cwnd. Cwnd change shows the TCP performance during the transmission. Detail graphic presentation of the cwnd variation of the analyzed protocols is presented at the Figures 52-95, from where we can exactly notice in which phase at the time is the current TCP protocol and how it influence at the throughput. Like we stated before the impact of the MAC layer is not minor it prevents flow losses caused by transmission errors. The analysis continues with NewReno cwnd change in case with no losses and activated DelACK mechanism. At Figure 52 we can notice that there is no change in the behavior of the cwnd when IFQ receives values of 15, 25, 50 packets. If we compare Figure 52 with Figure 57 the case when DelACK mechanism is activated the difference can be noticed in way that at Figure 57 cwnd has more dense, more narrow behavior that indicates that the change of the phases is more frequent.

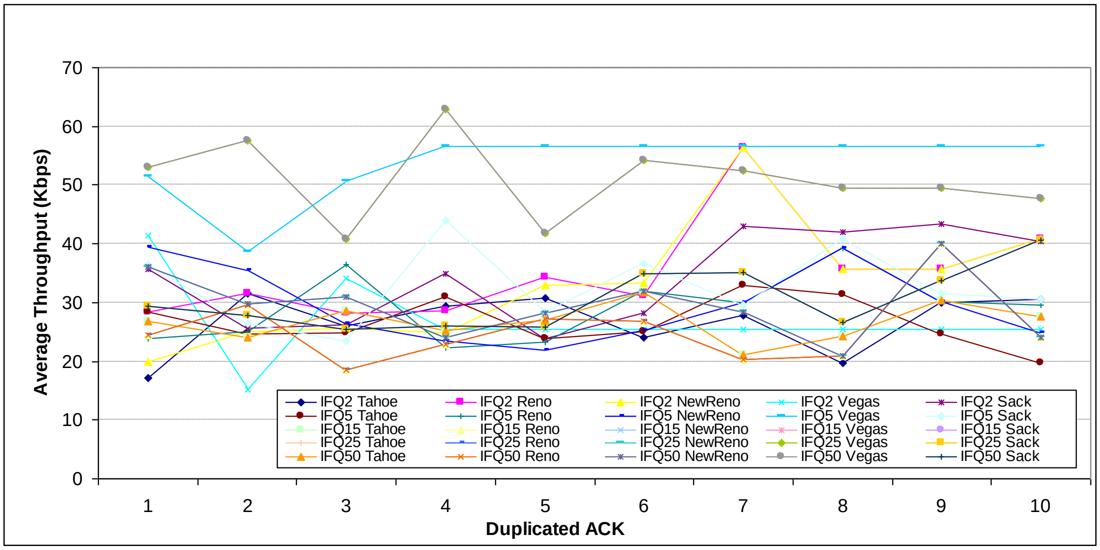

Figure 48. TCP throughput variation for different number of duplicated ACKs and excluded losses

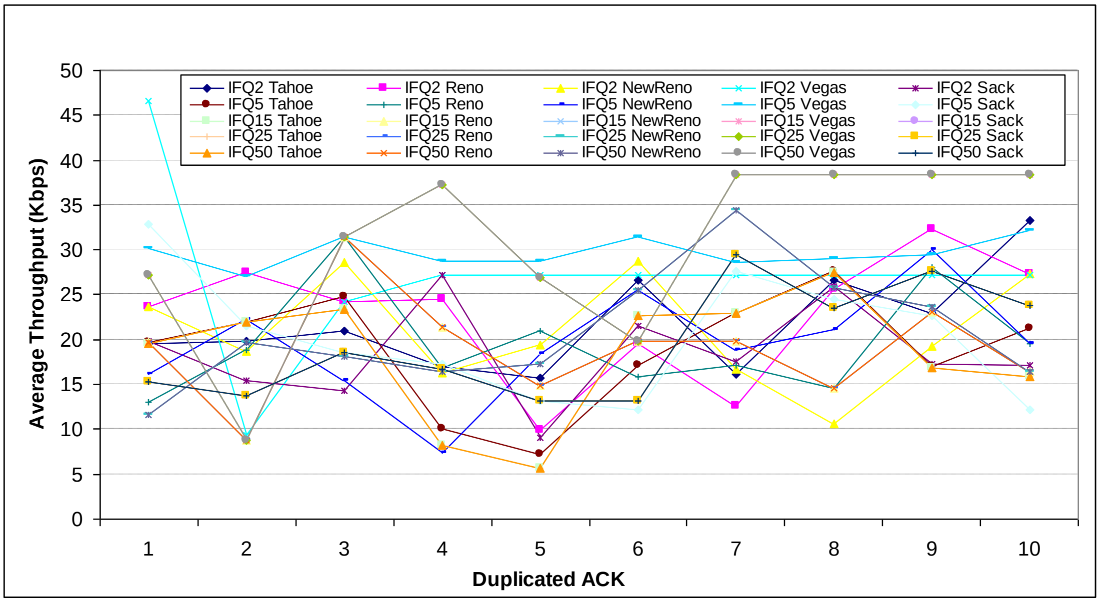

Figure 49. TCP Throughput variation for different number of duplicated ACKs in case of 20 % losses

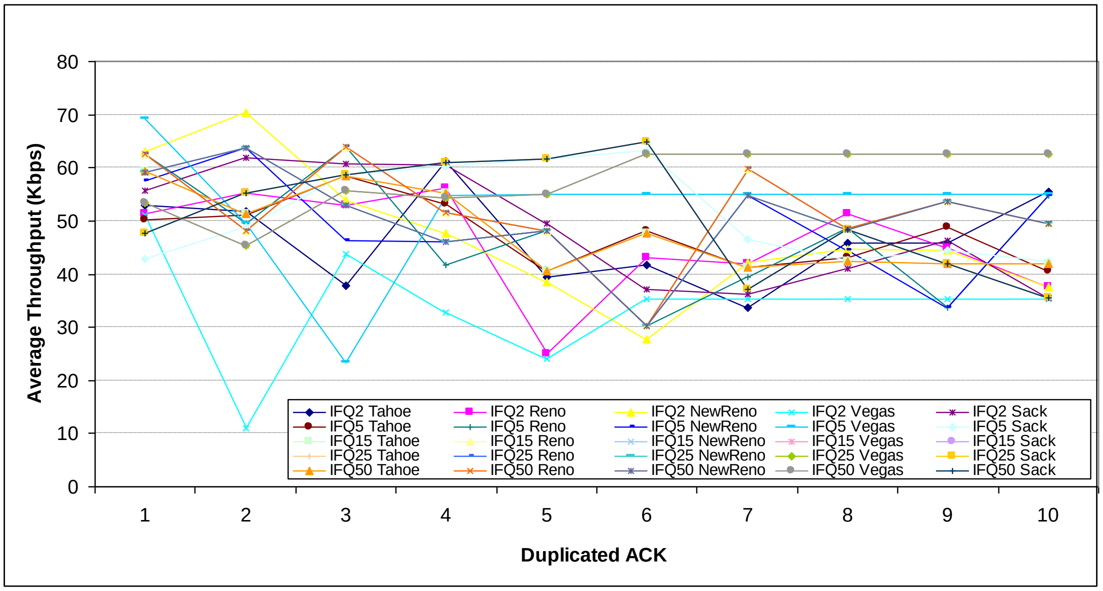

Figure 50. TCP throughput for different values of the number of duplicated ACKs and 0% losses. The Number of delayed ACKs is set at 2.

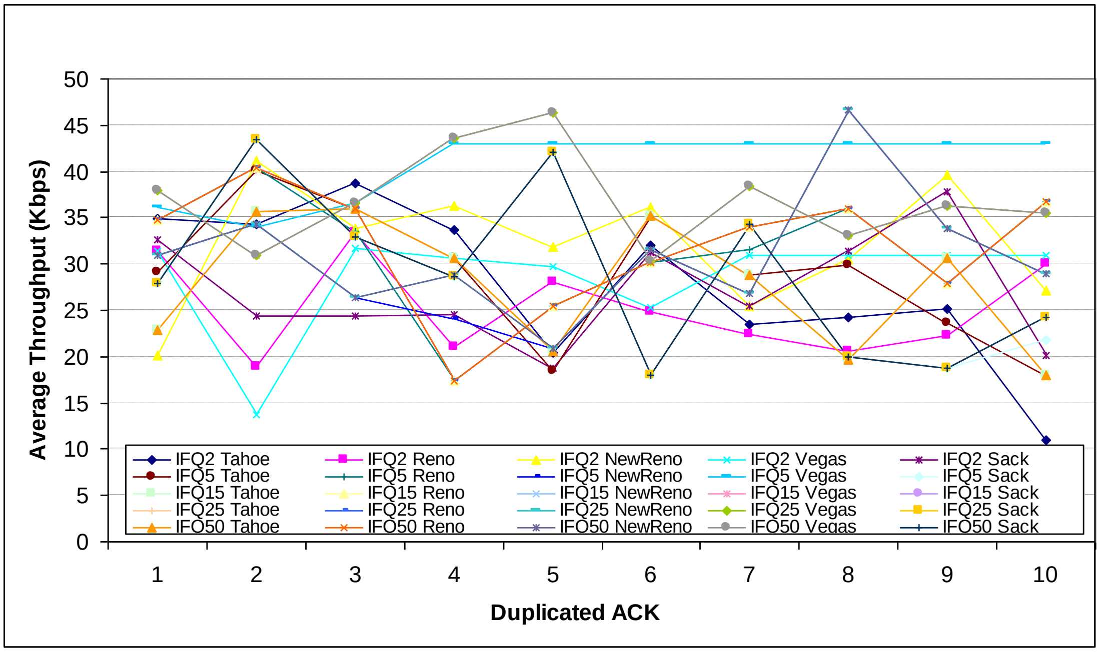

Figure 51. TCP throughput for different values of the number of duplicated ACKs and 20% of losses. The Number of delayed ACKs is set at 2.

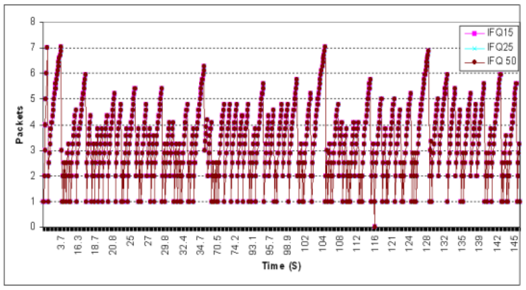

Figure 52. NewReno CWND change for 0 % of losses and IFQ set at 15, 25, 50 pkt.

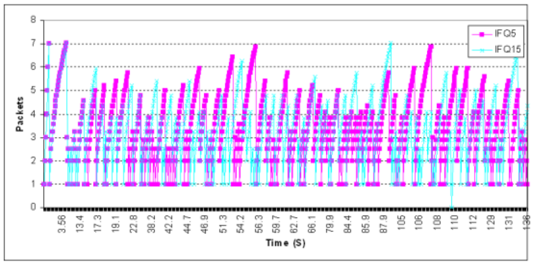

Figure 53. NewReno CWND change for 0 % of losses and IFQ values of 5, 15 pkt.

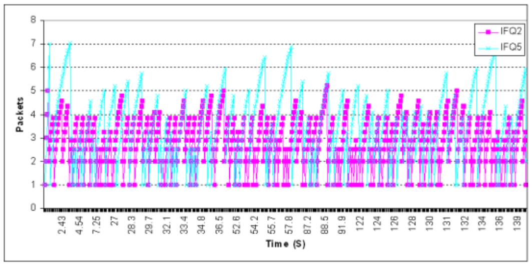

Figure 54. NewReno CWND change for 0 % of losses and IFQ values of 2, 5 pkt.

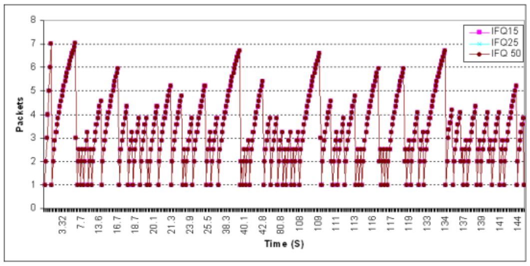

Figure 55. Reno CWND change for 0 % of losses and IFQ values of 15, 25, 50 pkt.

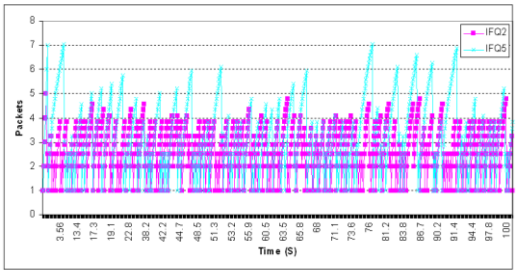

Figure 56. Reno CWND change for 0 % of losses and IFQ values of 2, 5 pkt.

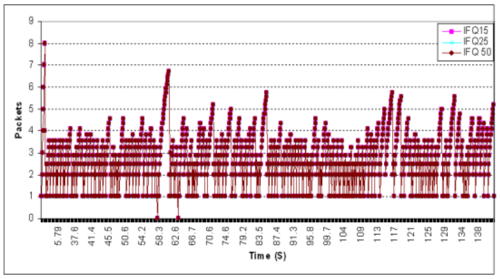

Figure 57. NewReno DelACK CWND change for 0 % of losses, IFQ set at 15, 25, 50 pkt.

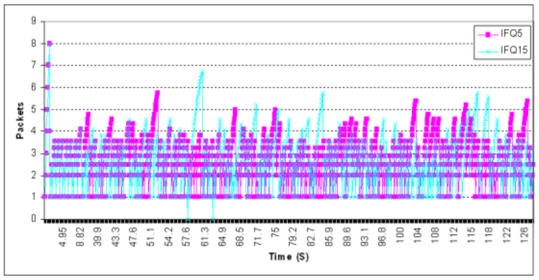

Figure 58. NewReno DelACK CWND change for 0 % of losses, IFQ set at 5, 15 pkt.

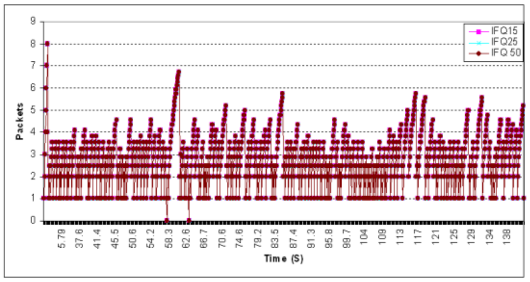

Figure 59. NewReno DelACK CWND change for 0 % of losses, IFQ set at 2, 5 pkt.

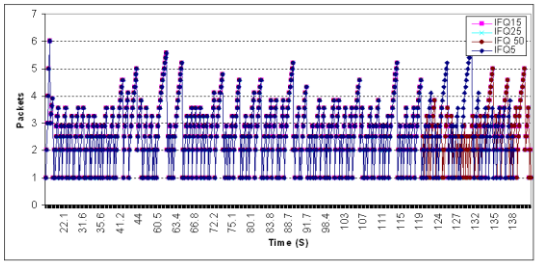

Figure 60. Reno DelACK CWND change for 0 % of losses, IFQ set at 5, 15, 25,50 pkt.

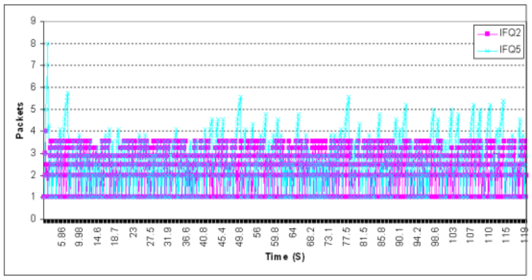

Figure 61. Reno DelACK CWND change for 0 % of losses and IFQ values of 2, 5 pkt.

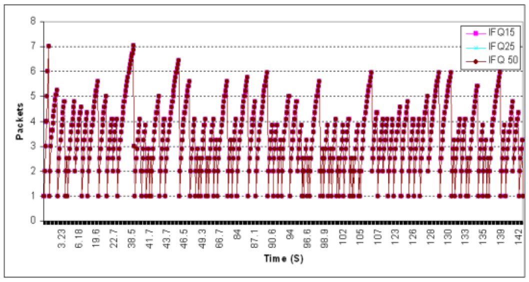

Figure 62. Sack CWND change for 0 % of losses and IFQ values of 2, 5 packets

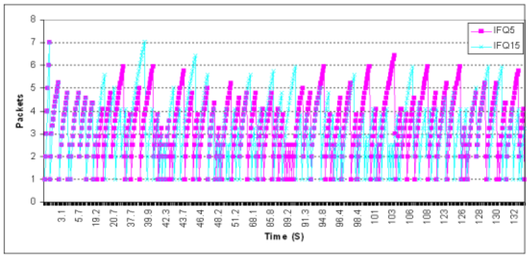

Figure 63. Sack CWND change for 0 % of losses and IFQ values of 2, 5 packets

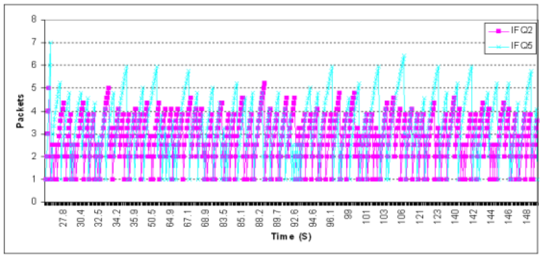

Figure 64. Sack CWND change for 0 % of losses and IFQ values of 2, 5 packets

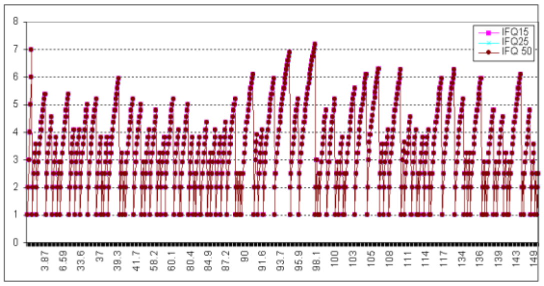

Figure 65. Tahoe CWND change for 0 % of losses and IFQ values of 15, 25, 50 packets

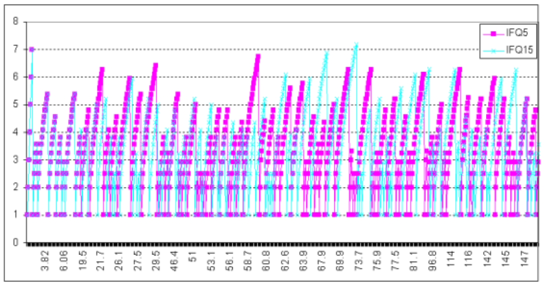

Figure 66. Tahoe CWND change for 0 % of losses and IFQ values of 5, 15 packets

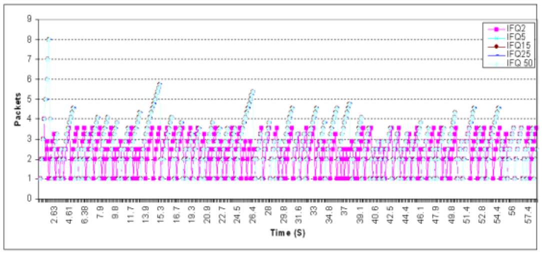

Figure 67. Sack DelACK CWND change for 0% of losses and IFQ values of 2- 50 packets

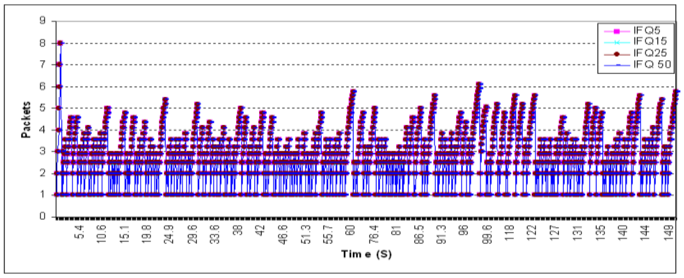

Figure 68. Tahoe DelACK CWND change for 0% of losses and IFQ values of 5-50 packets

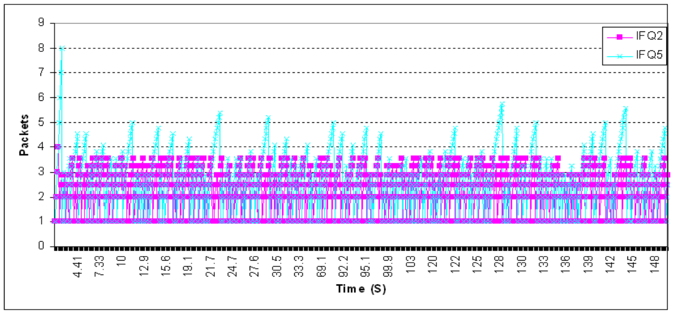

Figure 69. Tahoe DelACK CWND change for 0% of losses and IFQ values of 2-5 packets

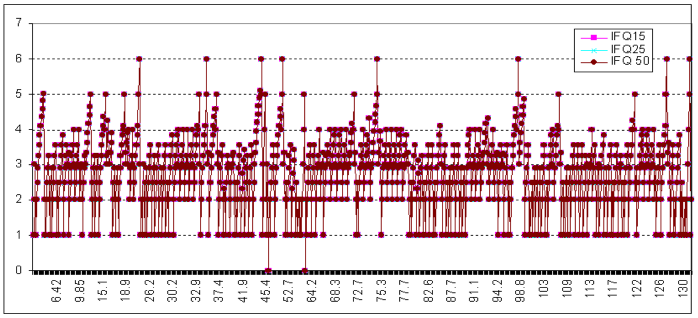

Figure 70. Vegas DelACK CWND change for 0% of losses and IFQ set at 15-50 packets

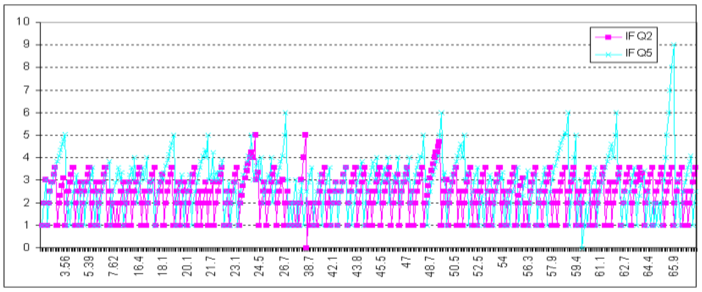

Figure 71. Vegas DelACK CWND change for 0 % of losses and IFQ values of 2-5 packets

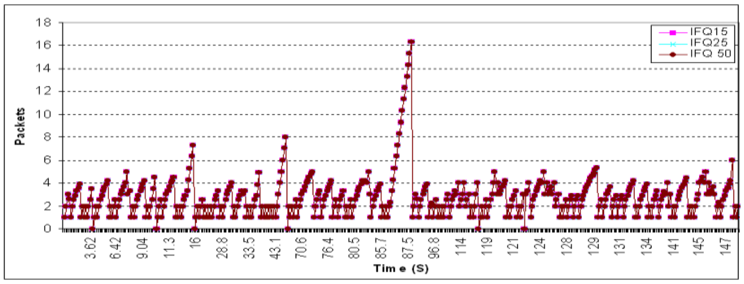

Figure 72. Vegas CWND change for 0 % of losses and IFQ values of 15, 25, 50 packets

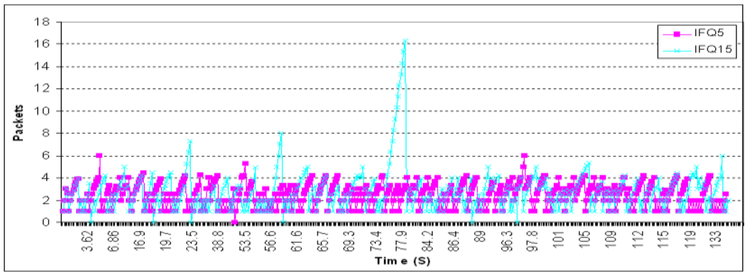

Figure 73. Vegas CWND change for 0 % of losses and IFQ values of 5, 15 packets

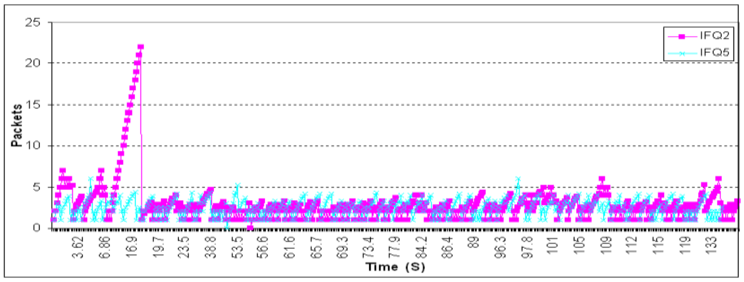

Figure 74. Vegas CWND change for 0 % of losses and IFQ values of 2, 5 packets

This cwnd behavior at first sight may lead us wrong and we can say that better throughput is achieved when DelACK is excluded but the throughput analysis showed that DelACK mechanism improves the throughput in this case. If we decrease the IFQ value at 5 and 2 packets that the peak cwnd value is decreased and the curve overlapping is less present because of more narrow shape of the window. This impacts the throughput badly. The change of the DupACK will influence at the throughput and cwnd change but in this scenario the analyses is conducted when it receives its default value of 3 duplicated ack. Cwnd change in case when IFQ is set at values of 5 and 2 packets is given in Figures 53, 54 and 58, 59 respectively. At Figure 55 is presented cwnd change when Reno is used as a transport protocol. If we compare Figures 52 and 55 the better Reno cwnd behavior is obvious that explains why better throughput is achieved when is used Reno as a transport protocol. In this case cwnd does not change the shape when IFQ is set at 15,25,50 packets. If we look at Figure 60 than in this case cwnd has the same change even when IFQ is set at value of 5 packets. From 55 and 60 can be concluded that when DelACK mechanism is activated the behavior of the cwnd becomes more narrow this cwnd behavior improves the achieved throughput. Further decrease of IFQ value leads us to the comment given when we analyzed NewReno cwnd. In Figures 62, 63, 64 we can see the Sack cwnd change. Visible difference of the behavior can be noticed when DelACK mechanism is activated and in this case IFQ does not impacts the cwnd change, Figure 67. Tahoe cwnd change is presented in Figures 65 and 66. In this case the overlapping exists when IFQ is set at 15, 25 and 50 packets when DelACK is activated he have cwnd overlapping and for IFQ set at 5 packets, like in the previous cases and in this when DelACK is used cwnd has more narrow shape.

In Figures 70-74 is presented the Vegas cwnd change we can notice that it has unique behavior that does not surplices if we have in mind the nature of this protocol. The overlapping of the curves is noticeable when IFQ receive values of 15, 25, 50 packets and there is not grate difference when it is set at values of 5 and 2. When DelACK is activated rapid change of the behavior is noticed it negatively impacts the throughput too. This behavior of the cwnd can be explained with the increased delay time that is present. If we incorporate 20 % losses in the scenario than the obtained cwnd shapes are presented at the rest of the Figures starting from 75 and ending with 94. In this case the analysis will bi similar like the previous in this case the lover value of the IFQ does not impacts the cwnd change strongly. When IFQ is set at value of 2 than cwnd receives most narrow shape in all scenarios. If we compare the change of cwnd when DelACK is activated there is no grate difference. Other important parameter that influences the throughput is the average packet delay. The graphical presentation of the average packet delay is given at the Figures 95-98.

From Figure 95 we can see that when IFQ is set at 2 packets the average packet delay is around 70 ms and the lowest delay is achieved when is used Reno as a transport protocol. As we increase the IFQ value the average delay increases too. For IFQ set at value of 5 packets it is set at 130ms lowest is achieved with Vegas and is around 90 ms. When IFQ receive values of 15, 25, 50 packets the average delay has same values for the used protocol and is in range of 126ms-178ms and the lowest is 90ms achieved with Vegas.

If we incorporate 20 % of losses than the average delay change is shown at Figure 96. In this case the lowest delay is achieved when IFQ is set at 2 packets and it is in range of 100 and 160 ms. New Reno has highest packet delay and lowest is achieved with Sack. If we increase the IFQ value at 5 packets then Tahoe has highest delay of 218 ms and the lowest is achieved with Vegas. NewReno Reno and Sack are between the border values.

If we incorporate DelACK mechanism the average packet delay is presented in Figures 97 and 98.

Like can be seen from the Figures there is no grate variation of the average delay parameter. The range of values starts from 70 ms and ends with 125 ms. Tahoe, Reno, Sack and Vegas show similar behavior. Highest delay values are achieved when IFQ receives value of 5 packets and lowest when IFQ is set at 2 packets. Overlapping of the values is obvious when IFQ is set at 15, 25 and 50 packets. This comment stands in the case when is activated DelACK mechanism as presented at Figure 98. The only thing that is changed is the range of values that is increased and starts with 96 ms and ends with 165 ms. When IFQ receives value of 2 NewReno provides lowest delay and Tahoe has worst time. If IFQ is increased then Reno takes the first, Sack, Vegas, NewReno and at the last place is Tahoe. As we can notice from the stated before this analysis and protocol ranking can be done by different parameters.

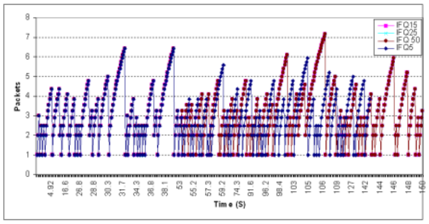

Figure 75. NewReno CWND change for 20% of losses, IFQ set at 5,15,25,50 pkt.

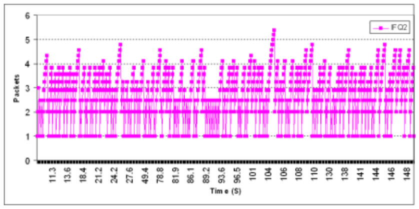

Figure 76. NewReno CWND change for 20% of losses, IFQ set at 2 pkt.

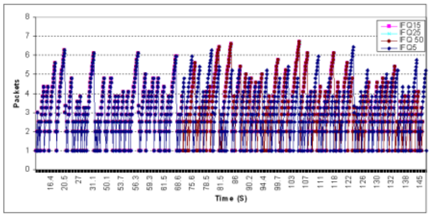

Figure 77. Reno CWND change for 20 % of losses, IFQ set at 5,15,25,50 pkt.

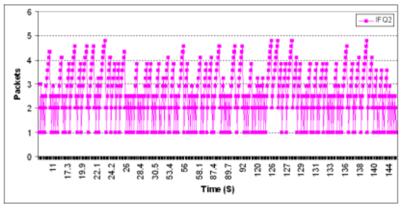

Figure 78. Reno CWND change for 20 % of losses, IFQ set at 2 pkt.

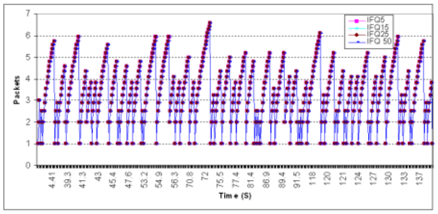

Figure 79. Sack CWND change for 20 % of losses and IFQ values of 5-50 packets

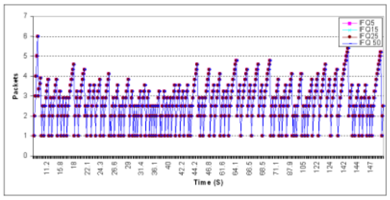

Figure 80. NewReno DelACK CWND change for 20 % of losses, IFQ set at 5,15,25,50 pkt.

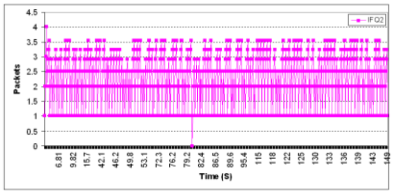

Figure 81. NewReno DelACK CWND change for 20 % of losses, IFQ set at 2 pkt.

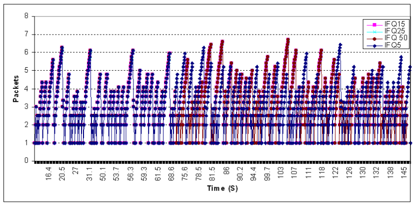

Figure 82. Reno DelACK CWND change for 20% of losses, IFQ set at 5,15,25,50 pkt.

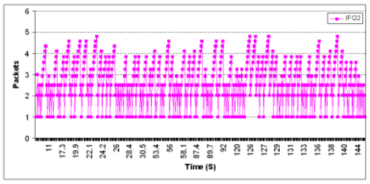

Figure 83. Reno DelACK CWND change for 20% of losses and IFQ value of 2 packets

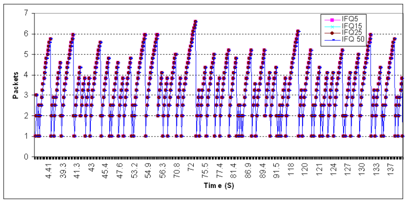

Figure 84. Sack DelACK CWND change for 20% of losses, IFQ set at 5,15,25,50 pkt.

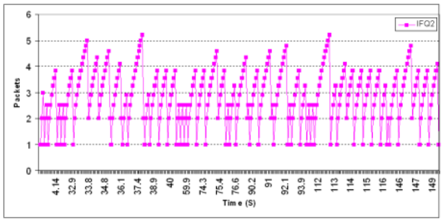

Figure 85. Sack CWND change for 20 % of losses and IFQ values of 2 pkt.

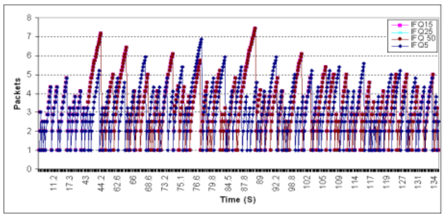

Figure 86. Tahoe CWND change for 20 % of losses and IFQ values of 5,15,25,50 pkt.

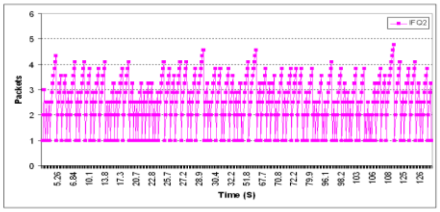

Figure 87. Tahoe CWND change for 20 % of losses and IFQ values of 2 pkt.

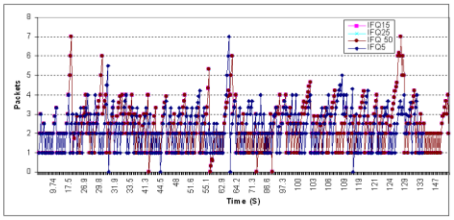

Figure 88. Vegas CWND change for 20 % of losses and IFQ values of 5,15,25,50 pkt.

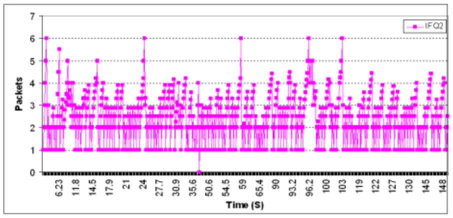

Figure 89. Vegas CWND change for 20 % of losses and IFQ values of 2 pkt.

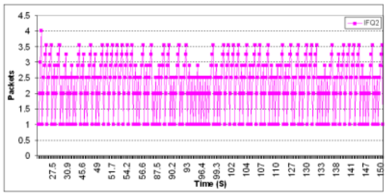

Figure 90. Sack DelACK CWND change for 20 % of losses and IFQ value of 2 pkt.

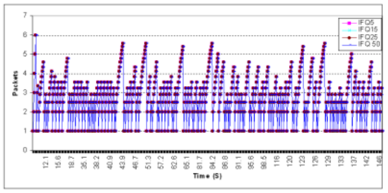

Figure 91. Tahoe DelACK CWND change for 20 % of losses, IFQ set at 5,15,25,50 pkt.

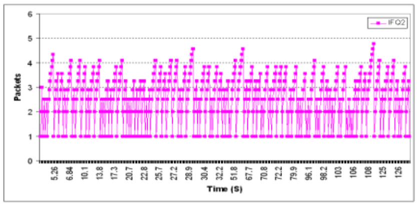

Figure 92. Tahoe DelACK CWND change for 20 % of losses and IFQ values of 2 pkt.

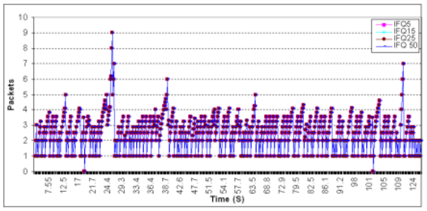

Figure 93. Vegas DelACK CWND change for 20 % of losses, IFQ set at 5,15,25,50 pkt.

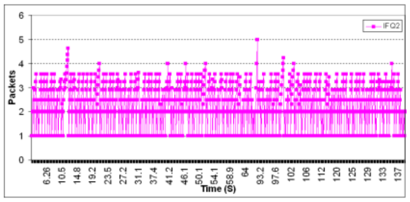

Figure 94. Vegas DelACK CWND change for 20 % of losses and IFQ values of 2 pkt.

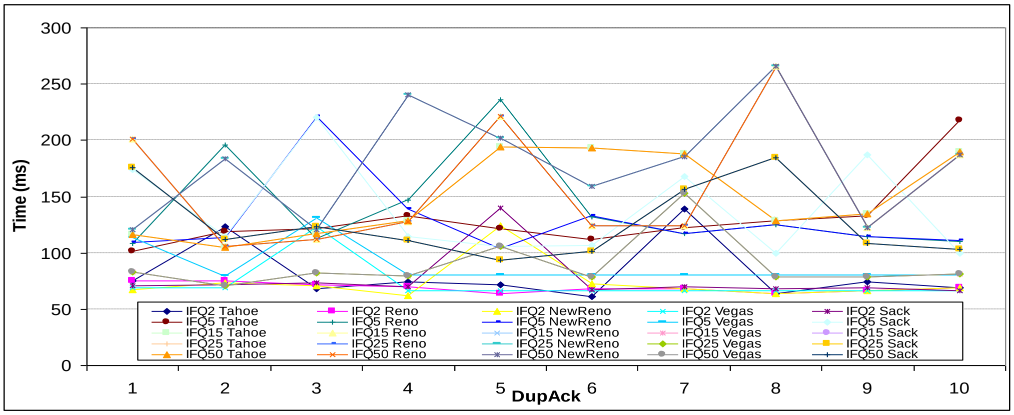

Figure 95. Average Delay for 0 % of losses and IFQ values of 2, 5, 15,25, 50 Packets

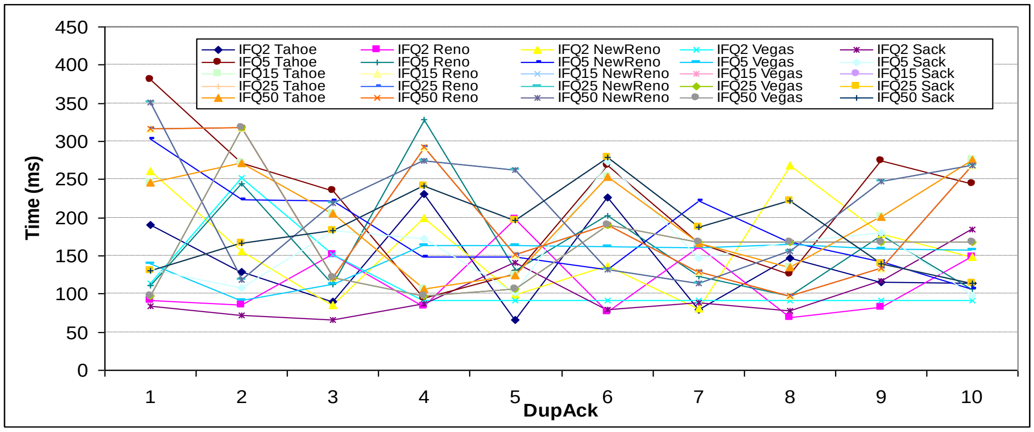

Figure 96. Average Delay for 20 % of losses and IFQ values of 2, 5, 15,25, 50 Packets

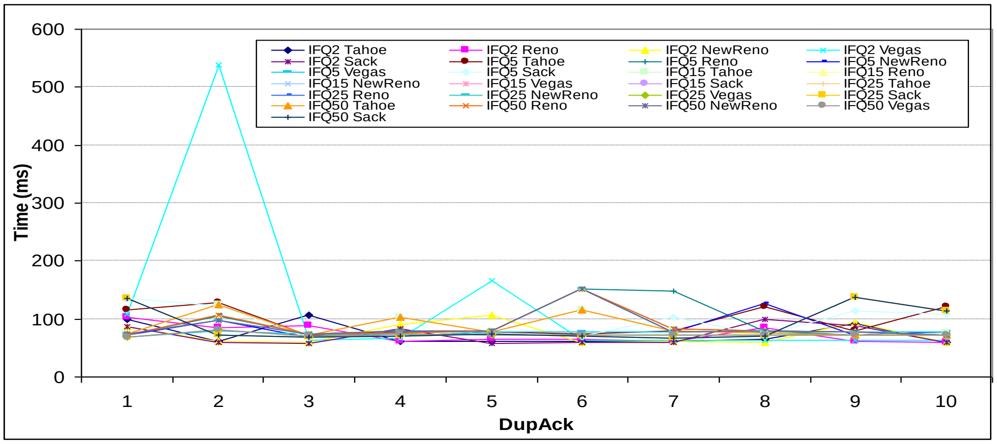

Figure 97. Average Delay for 0 % of losses, IFQ values of 2, 5, 15,25, 50 Packets, and DelACK Activated

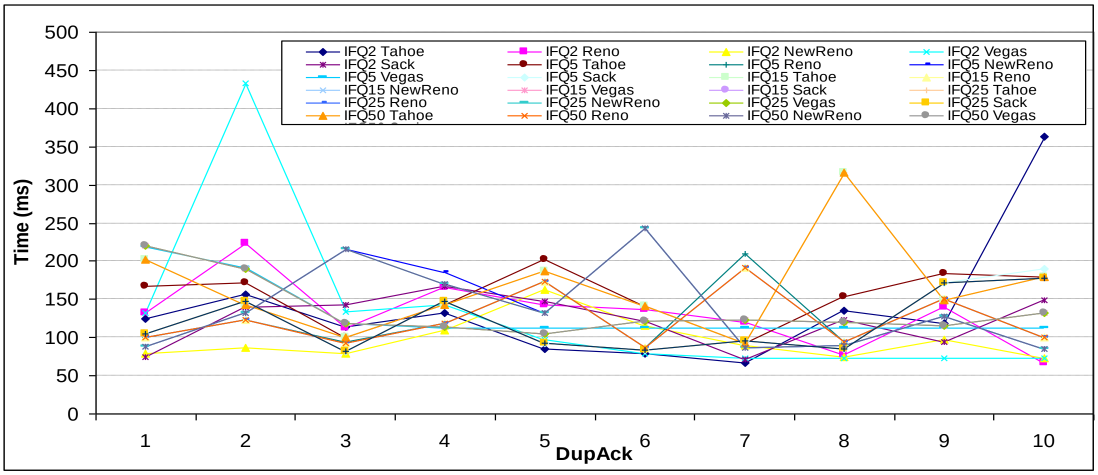

Figure 98. Average Delay for 20 % of losses, IFQ values of 2, 5, 15,25, 50 Packets, and DelACK Activated

5. Conclusion

In this paper we have performed detailed traffic analyses regarding the performances of different TCP versions in wireless chain network scenario. We have used the throughput as a merit. The results showed high importance of the DupACK parameter. If we analyze the system in conditions without any incorporated losses then we can say that there is a need to optimize DupACK value [15], which is by default set at 3. In the nine node scenario this parameter can be set at values of 1, 9, 2, 6, or 7, and the obtained throughput will be better than in the case when this parameter is set at 3. When we incorporated the DelACK mechanism in scenario with no losses then we have concluded the same.

On the side, when we have included 20% losses in the scenario then we have obtained that the best value of DupACK parameter to be 8, 7, 9, 10, and 2, which was also the case when the DelACK mechanism was activated. However, if we have different loss probability then we may expect different optimized DupACK value for the fast retransmission mechanism of the TCP. Overall best performance of the analyzed protocols when IFQ receives values of 15, 25, 50 packet was achieved by Vegas, followed by NewReno, Reno, Sack and Tahoe. Another key point is that increased IFQ does not improve the throughput that was provided in all scenarios when IFQ was set at 15, 25, 50 packets. One may conclude that higher throughput is achieved for smaller values of IFQ. In the case of two node scenario we can notice 10 times higher throughput than in the case with nine nodes, there is no throughput variation for IFQ values of 25 and 50 packets, increment of the throughput is noticed when IFQ is set at 2, 5 packets in some cases there is small degradation when IFQ is set at 15 packets. DupACK has influence on the throughput, and better throughput is achieved when it is set at 1, 2, 5, 7, and 6. Finally, DupACK and the transport protocol should be dynamically adjusted in wireless environment. DelACK mechanism should be incorporated in wireless scenarios. IFQ value should be dynamically adjusted and should be in correlation with the distance of the nodes and the propagation model. It was shown that change of cwnd is important parameter that describes the TCP behavior and can help during the TCP optimization. Hence, dynamic change of the cwnd and the phase behavior can lead to new improved TCP protocol.

References

[1] W. R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols", Addison Wesley, **(1994)**.

[2] V. Jacobson, "Congestion Avoidance and Control", SIGCOMM Symposium on Communications Architectures and Protocols, **(1988)**, pp. 314–329.

[3] V. Jacobson. "Modified TCP Congestion Avoidance Algorithm", Technical report, **(1990)** April 30.

[4] K. Fall and S. Floyd, **"**Simulation-based comparisons of Tahoe, Reno and SACK TCP**",** ACM SIGCOMM Computer Communication Review, vol. 26, Issue 3, **(1996)** July, pp. 5 - 21.

[5] B. Moraru, F. Copaciu, G. Lazar and V. Dobrota, "Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno", 2nd RoEduNet International Conference, **(2003)**.

[6] J. Mo, R. J. La, V. Anantharam and J. Walrand, "Analysis and comparison of TCP Reno and Vegas", IEEE INFOCOM '99, vol. 3, **(1999)** March, pp. 1556-1563.

[7] S. Floyd, "Metrics for the Evaluation of Congestion Control Mechanisms," IRTF, Informational RFC 5166, **(2008)** March.

[8] S. Floyd and M. Allman, "Comments on the Usefulness of Simple Best-Effort Traffic," IETF," Internet Draft version 3, **(2008)** January.

[9] T. Janevski and I. Petrov, "Optimization of TCP/IP over 802.11 Wireless Networks in Home Environment", EuropeComm, London, **(2009)** August.

[10] J. -M. Chen, et. al., "Improving SCTP Performance by Jitter-Based Congestion Control overWired-Wireless Networks", EURASIP Journal on Wireless Communications and Networking, **(2011)**.

[11] J. Domzał, N. Ansari and A. Jajszczyk, "Congestion Control in Wireless Flow-Aware Networks", IEEE ICC 2011, Kyoto, **(2011)** June.

[12] H. Jung, et. al., "Adaptive Delay-based Congestion Control for High Bandwidth-Delay Product Networks", IEEE INFOCOM 2011, Shanghai, China, **(2011)** April 10-15.

[13] J. Wang, J. Wen, J. Zhang and Y. Han, "TCP-FIT: An Improved TCP Congestion Control Algorithm and its Performance", IEEE INFOCOM 2011, Shanghai, China, **(2011)** April 10-15.

[14] T. Janevski and I. Petrov, "Analysis of Mobile IP for NS2", Telfor 2008, Belgrade, Serbia, **(2008)** November.

[15] T. Janevski and I. Petrov, "Analysis of the Impact of Duplicated ACK on the Performances of TCP", Telfor 2011, Belgrade, Serbia, **(2011)** November.

Authors

Toni Janevski

Dr. Toni Janevski is a Full Professor at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, Macedonia. He received his Dipl. Ing., M.Sc. and Ph.D. degrees in electrical engineering all from Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, in 1996, 1999 and 2001, respectively. In the past, during 1996-1999 he has worked at the Macedonian mobile operator Mobimak (currently T-Mobile, Macedonia), contributing to the planning, dimensioning and implementation of the first mobile network in Macedonia. From 1999 he is with Faculty of Electrical Engineering and Information Technologies in Skopje. In 2001 he has conducted research in optical communication at IBM T. J. Watson Research Center, New York. Also, during 2005-2008 he was a Member of the Commission of the Agency for Electronic Communications (AEC) of the Republic of Macedonia, where he contributed towards the introduction of new technologies in Macedonia, such as WiMAX and 3G mobile networks, as well as new operators and services. During the periods 2008-2012 and 2012-2016 he is an elected member of the Senate of the Ss. Cyril and Methodius University in Skopje. In 2009 he has established Macedonian ITU (International

Telecommunication Union) Centre of Excellence (CoE) as part of the Europe's CoE network, and serves as its Coordinator since than. He is an author of the book titled "Traffic Analysis and Design of Wireless IP Networks", which is published in 2003 by Artech House Inc, Boston, USA. Also, he is author of the book "Switching and Routing", written in Macedonian language, published in September 2011 by the Ss. Cyril and Methodius University in Skopje, for which in May 2012 he has won the "Goce Delchev" award, the highest award for science in the Republic of Macedonia. He has published more than 130 research papers and has led research and applicative projects in the area of Internet technologies and mobile and wireless networks. He is a Senior Member of IEEE. His research interests include Internet Technologies, Mobile, Wireless and Multimedia Networks, Traffic Theory, Quality of Service, Design and Modeling of Telecommunication Networks, as well as Next Generation Networks.

Ivan Petrov

Ivan Petrov is with telecom operator Makedonski Telekom, Skopje, Macedonia. He received his Dipl.Ing. and M.Sc. from the Faculty of Electrical Engineering, Ss. Cyril and Methodius University in Skopje, in 2005 and 2009, respectively. From 2009 he is working towards his Ph.D. degree under mentorship of Prof. Toni Janevski. His research interests include transport protocols and cross-layer optimization techniques in heterogeneous wireless IP networks.