

Cross-layer TCP Performance Analysis in IEEE 802.11 Vehicular Environments

Toni Janevski, *Senior Member, IEEE*, and Ivan Petrov

Abstract — In this paper we provide a performance analysis of TCP in IEEE 802.11 vehicular environments for different well-known TCP versions, such as Tahoe, Reno, New Reno, Vegas, and Sack. The parameters of interest from the TCP side are the number of Duplicate Acknowledgements - DupAck, and the number of Delayed Acknowledgements - DelAck, while on the wireless network side the analyzed parameter is the interface queue - IFQ. We have made the analysis for the worst-case distance scenario for single-hop and worst-case multihop vehicular environments. The results show that the number of wireless hops in vehicular environments significantly reduces the TCP throughput. The best average performances considering all scenarios were obtained for TCP Vegas. However, the results show that the interface queue at wireless nodes should be at least five packets or more. On the other side, due to shorter distances in the vehicular wireless network, results show possible flexibility of using different values for the DupAck without degradation of the TCP throughput. On the other side, the introduction of the DelAck parameter provides enhancement in the average TCP throughput for all TCP versions.

Keywords — DelACK, DupACK, IEEE 802.11, Multihop, TCP, Throughput, Vehicular, Wireless.

I. INTRODUCTION

THE rapid development of Internet and wireless technologies has resulted in their integration, [1]. 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 a 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 a 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 time and the growth of traffic the need for design optimization of the transport

protocols was born and the first steps were made [2]. 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, several versions of TCP have been widely implemented. Starting from old Tahoe [3], then continuing with Reno [4] and further New Reno, which was followed by Vegas, Sack [5]-[7], until today when we have a huge number of different TCP versions that improve their behaviour either in wired or in wireless environment. Hence, every parameter used for the TCP design has an important role for the efficiency of transport protocol. Such design parameters make the transport protocol robust in different environments. For sure, every year we will have more and more efficient (more or less) modifications of the transport protocols because the initial condition, under which the TCP protocol was born, has changed [8]. Work on TCP has continued over the years for more efficient congestion control algorithms. On the other hand, this is also the case with network access protocols (below the network layer), such as IEEE 802.11 standards. The behaviour of 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 a cross-layer analysis of WiFi-based vehicular environments, where we consider the InterFace Queue for the WiFi interfaces and TCP parameters DupACK and DelAck, which directly influence the congestion control mechanism in different TCP versions.

This paper is organized as follows. Section two gives a brief overview of the analyzed transport protocols. Further, section III describes our simulation environment and results. Finally, section IV concludes the paper.

II. BASICS OF TRANSPORT PROTOCOLS

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's evolved versions of TCP contain a 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

Paper received September 16, 2013; revised June 16, 2014; accepted June 18, 2014. Date of publication July 31, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aleksandar Nesković.

Toni Janevski, PhD, is a Professor at the Ss. Cyril and Methodius University in Skopje, Faculty of Electrical Engineering and Information Technologies, Karpos 2 bb, 1000 Skopje, Macedonia, Email: tonij@feit.ukim.edu.mk.

Ivan Petrov, MSc, is a PhD student at the Ss. Cyril and Methodius University in Skopje, Faculty of Electrical Engineering and Information Technologies, and employed as an engineer at Makedonski Telekom, Orce Nikolov bb, 1000 Skopje, Macedonia, Email: Ivan.Petrov@telekom.mk.

TCP connections on longer time scales.

There are three basic phases that describe the behaviour of TCP protocols. A Slow Start is the initial phase (the smallest congestion window), which is followed by congestion detection and Congestion Avoidance phase. Further, congestion detection is standardized via the Fast Retransmission and Fast Recovery mechanisms. One of the main TCP objectives is to create a reliable connection by retransmitting lost packets; for this purpose TCPs incorporate the mechanism of acknowledgements (ACKs). TCP receiving side acknowledges each TCP segment (or a series of them), by using the acknowledgement by the last TCP segment received in order (as sent). So, when a loss occurs (or reordering of segments), and then the following TCP segments (after the lost one) continue to arrive at the receiver, the receiver sends ACKs for each of them, but carrying the number of the last successfully received octet in the order. So, that way we get duplicate ACKs when losses occur in a TCP connection. However, this also causes Head of the Line (HoL) blocking at TCP socket on the receiving side, since the TCP uses streaming API (Application Programming Interface) sockets, which means that application on the receiving side expects the octets in the exactly same order, without any losses, as they were sent from the application on the sending side via its stream socket used by TCP. Then, in most of TCP versions, a packet is considered as lost if three repeated ACKs from receiver arrive at the source (loss triggered), or when there is a timer expiration meaning that ACK does not arrive in a given time period (time triggered). Retransmission of the lost segments after three duplicated ACKs is in fact the Fast Retransmission mechanism, which is usually followed by the Fast Recovery mechanism, which halves the congestion window instead going into the Slow Start. Further in this paper we will analyse the number of ACKs, i.e. what happens if we change the number of ACKs that triggers the fast retransmission phase of TCP, using different TCP versions in multihop wireless environment for which TCP is not initially created.

Also, there is a DelAck parameter, which determines the number of TCP packets confirmed with a single acknowledgement, thus reducing the ACK traffic from receiver towards the sender. In the following simulation analysis, we also include changing of the DelAck parameter for the TCP.

III. SIMULATION ANALYSIS OF TRANSPORT PROTOCOLS IN IEEE 802.11 VEHICULAR ENVIRONMENTS

We analyse two different scenarios A) a single hop wireless scenario, and B) a multihop (9 hops) wireless scenario. In both scenarios the wireless nodes are wireless routers integrated in vehicles, which are moving at an average city speed of 20 kmph (normal distribution), a typical speed average for vehicles in urban environments as a focus of interest in the following analysis. We are using 802.11g standard at MAC/Physical layer that limits our max link throughput to 54 Mbps. The configuration of links between the nodes is given in Table I. We use Two Ray Ground model in simulations. We observe the

following TCP protocols: Tahoe, Reno, New Reno, Vegas, and Sack. Different parameter values used in this scenario are listed in Table II. The 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 250 seconds. TCP packet size is 512 bytes, while ACK packets have their well known value. For simulations, we use the Network Simulator 2 (NS-2).

We observe the following TCP protocols: Tahoe, Reno, New Reno, Vegas, Sack (Selective acknowledgements). Different parameter values used in this scenario are listed in Table II. The analysis is targeting the maximum throughput in case with and without incorporated loss model that causes 20% throughput degradation. Simulation duration is set at 250 seconds. TCP packets' size is 512 bytes while ACK packets have their well known value.

TABLE 1. NETWORK CONFIGURATION.

<i>Link</i>	<i>Distance</i>	<i>Max. throughput</i>
$n_i-n_{i+1}, i=1, 2, \dots, 8$	200 m	54 Mbps
<i>Propagation Model</i>	<i>MAC Protocol</i>	<i>TCP Packet Size</i>
Two Ray Ground	802.11g	512 byte

TABLE 2. SIMULATION PARAMETERS.

<i>Parameter</i>	<i>Value</i>	<i>Comment</i>
DupACK	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Default val. is 3 DupAck
IFQ size (in packets) n_i-n_{i+1}	2, 5, 15, 25, 50	Default value is Bandwidth Delay product
TCP	Tahoe, Reno, New Reno, Vegas, Sack	Used types of TCP
DelACK (in packets)	1, 2	DelACK mechanism

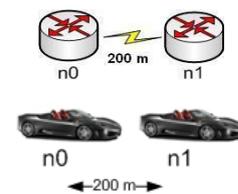


Fig. 1. Single hop vehicular wireless scenario (two nodes).

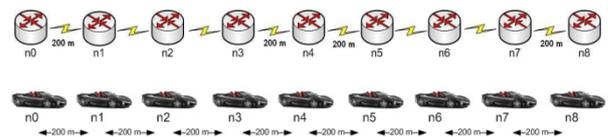


Fig. 2. Multihop vehicular scenario (nine nodes).

A. Singlehop Wireless Scenario

In the following part we observe simulation results from the scenario presented in Figure 1, where we have a single hop wireless scenario between two nodes at a distance of 200m, which results in the maximum possible TCP throughput (no background traffic) of almost 650 Kbps

using the IEEE 802.11g standards. However, at 200m distances WiFi has only the mode of 1Mbps available (due to a very low signal-to-noise ratio), and typical TCP utilization of WiFi is usually below 8/11 of the wireless network deterministic throughput (determined by the modulation and error-control coding). So, this is the worst case analysis regarding the available throughputs of WiFi, which, in fact, are a real situation in a vehicular environment, where not all vehicles will have WiFi routing capabilities. We analyze 2 different situations, when we do not use DelAck and when we use DelAck=2, meaning that two consecutive TCP packets are acknowledged with a single ACK. Then, in each of the two cases we have different error ratios, one error-free (0% losses due to errors) and the other scenario with 20% losses, introduced by using a two-state on-off error model. One may note that 20% seems to be a high error ratio, but we are looking towards the worst case, and in a vehicular scenario where it may happen in many situations (although the error ratio may vary from user to user, from one link to another, etc.). So, we have 4 sub-scenarios in this single-hop analysis (i.e. wireless transfer between two nodes). We analyze the well-known TCP versions, which are most used today in the data plane, and they are: Tahoe, Reno, New Reno, Vegas, and Sack (note: for carrying signalling traffic, we have SCTP on the transport layer, but control plane and SCTP are out of the scope of this paper). Input parameters which are changing are IFQ size (in the number of TCP packets), and the number of Duplicated Acknowledgements. The main motivation in this single-hop scenario analysis and in the following multihop scenario is the analysis regarding the influence of the DupAck parameter on the TCP throughput as the main performance parameter for the TCP (since TCP is lossless due to retransmissions of the lost segments, either due to congestion or bit errors). Increasing DupAck threshold only makes sense if there are packet reordering and therefore spurious retransmissions. In a vehicular environment, packet reordering may vary in the network due to changing of the nodes (cars) on the path of TCP packets, which results in a change of TCP path and the length of TCP path over vehicular ad-hoc environment. So, DupAck analysis is quite important in a vehicular environment to optimize TCP regarding the possible throughput. In the following part let us analyze the obtained results in a single-hop scenario.

Figure 3 shows the case when we have no losses. Regarding the IFQ size, results show that a larger queue size leads to a higher throughput since there is a higher possibility to buffer packets and acknowledgements. TCP Reno is the one that needs bigger buffers and also TCP Vegas shows high degradation for small IFQ (e.g., IFQ=2). So, for this case, where there is no loss, we need IFQ above 20 packets to have a maximum throughput regardless of the TCP version, although there are small variations per TCP version as discussed above. On the other side, if we introduce a 20% error ratio in the link (Figure 4), in exactly the same scenario, we obtain different results regarding the IFQ influence on the TCP throughput. The best results are obtained in this case

(Figure 4) with TCP Vegas and they are constant for $IFQ \geq 5$ packets, while for all other analyzed TCP versions, the optimal IFQ size is 5 packets (for a maximum throughput) since the throughput slightly decreases with an increasing IFQ size. Of course, smaller IFQ sizes (e.g., IFQ=2) are unacceptable due to significant TCP throughput degradation. Regarding the DupAck threshold, from Figures 3 and 4 it can be seen that TCP throughput is almost constant for different DupAck thresholds above the value of 2. It is well known that the role of DupAck threshold is to avoid going into a timeout after TCP packet loss and then Slow Start. However, if we have a continuous TCP stream with an average packets size of 500 Bytes, then the time needed for sending 10 TCP packets over a 650 kbps link is around 62 ms, which is lower than TCP timeout values on short distances (a typical value is 500 ms). So, we may use a higher DupAck threshold in a vehicular scenario, of course, when the TCP traffic does not travel over the global Internet, but TCP senders and receivers are limited to cars (and fixed nodes near streets).

There is also a known mechanism for the reduction of reverse traffic in the network which helps and leaves more space for TCP, called a Delayed ACK mechanism. The main idea in this case is that not every TCP packet is acknowledged, but every second one (or every third, etc.). This way we may achieve the reduction of reverse traffic, and also reduce unnecessary Fast Retransmissions due to DupAcks resulted from packet reordering. The analyses of the TCP throughput for the same simulation scenarios, but now with DelAck=2, are given in Fig. 5 and Fig. 6 for the cases with 0% and 20% losses, respectively. The influence of IFQ size is similar to the case where DupAck=2 (Fig. 3 and Fig.4). Also, the throughput is almost not dependent upon the number of duplicated ACKs, due to the same reasons as discussed before, so we may use higher values for DupAck. However, an interesting result is the increment in the average TCP throughput in both cases, without and with errors in the link, when DelAck=2 is used (Fig. 5 and Fig. 6) compared to scenarios where DelAck=1 is used (Fig. 3 and Fig. 4). This can be explained by the reduction of the unnecessary retransmissions caused by the TCP packet reordering. So, in the single-hop scenario delayed acknowledgements give a better throughput. Also, TCP Vegas gives the best throughput results in all 4 cases in this simulation scenario, or course, for larger IFQ sizes.

B. Multihop Wireless Scenario

In the following part we provide simulation results from the scenario presented in Figure 2, where we have 8 hops between 9 wireless nodes (i.e. cars). Cars are moving at a speed of 20 kmph and have a distance of 200 m between each pair of them. Further, we generated the same results as in the single hop scenario, with the aim to obtain the TCP performance when TCP traffic travels multiple hops (each hop is an IEEE 802.11 link). We consider this one as the worst-case scenario regarding the number of hops and the distance between the nodes (i.e., cars), because a further increase in either of the parameters leads to

unusable communication based on IEEE 802.11 standards.

First of all, multiple hops decrease the overall TCP throughput, something that is expected due to more nodes to buffer packets, and more wireless links to travel. So, the available throughput (out of maximum 1 Mbps) is maximum 60-70 kbps in the cases without losses (e.g., in Fig. 7), which is nearly 10 times less than single-hop bit rates for the TCP. And, the bit rates are even lower when we introduce an error ratio in the wireless interface, something that is also expected due to a higher probability of packet loss because of bit errors. Again, we perform two types of simulations, one with DelAck=1 (Fig. 7 and Fig. 8) and the other with DelAck=2 (Fig. 9 and Fig. 10). Different DupAck thresholds provide similar behaviour due to shorter end-to-end distances for the TCP packets (which are 8 hops in this scenario, each hop equals 200 m). In the case without losses (Fig. 7), it is obvious that Vegas gives the best throughput for TCP for $IFQ \geq 5$ packets. Such a better performance of TCP Sack can also be seen in the case of a scenario with 20% losses (Fig. 8), but the bit rates are lower due to losses. In both of these cases DelAck=1, i.e., a TCP receiver acknowledges each received TCP packet.

Further, we set the DelAck to 2 and generate the same simulations, which give the results shown in Fig. 9 and Fig. 10. It is noticeable that we have a better TCP throughput for all TCP versions (on the average) compared to the case with DelAck=1, i.e., Fig. 7 and Fig. 8, respectively. Again, TCP Vegas has the best average results regarding the throughput, although in these two cases other TCP versions show similar and comparable performances to the TCP Vegas.

Finally, the results show that there is space to improve and readjust the values of certain parameters associated with IEEE 802.11 and TCP for better performances in wireless vehicular environment.

IV. CONCLUSION

In this paper we have performed analyses regarding the performances of different TCP versions in wireless vehicular environments. The analyses were focused on the worst-case scenarios for TCP. The worst-case scenario in a single-hop wireless network is the situation with a maximum distance between the wireless nodes (which are within vehicles moving in the urban area) where communication is still possible, but at the lowest possible data rates in the wireless link (which is IEEE 802.11, i.e. WiFi). Further, we have added the worst-case scenario regarding the number of hops, which resulted in the analyses of 8 hops wireless scenarios (i.e., multihop) for the TCP. For the TCP, we have used a continuous stream in all cases. The parameters of interest on the TCP side

were DupAck threshold and DelAck, while the parameter of interest on the IEEE 802.11 side was the IFQ length (i.e. interface buffer size). With this approach we have performed the cross-layer analyses in the worst-case scenarios for the transmission of TCP over WiFi in a vehicular environment. The analyses were performed for well-known TCP versions, including Tahoe, Reno, New Reno, Vegas and Sack.

If we are going to summarize, we may give four outcomes, regarding the DupAck, DelAck, IFQ size, and TCP version, regarding the maximization of the TCP throughput as a performance merit. The results show that IFQ length should be at least 5 packets or more, with the aim to limit losses due to congestion. DupAck threshold has a similar impact for all values up to DupAck=10 in the analyses, since TCP packets travel a short distance in both cases, in a single-hop as well as in a multihop scenario (they do not travel over the global Internet). Finally, DelAck showed impact on the obtained TCP throughput, a conclusion based on the results obtained with DelAck=2 compared to the results obtained with DelAck=1. This is a result of avoiding unnecessary retransmissions due to TCP packets which arrive at the destination out of order. Hence, DelACK mechanism should be incorporated in wireless vehicular environments. Additionally, WiFi IFQ values should be dynamically adjusted and should be in correlation with a node distance and propagation model.

Our further work is towards the creation of a cross-layer mechanism which will provide optimal adjustment of the transport layer protocol to the wireless network interface in vehicular environments.

REFERENCES

- [1] T. Janevski, *Traffic Analysis and Design of Wireless IP Networks*. Artech House Inc., Boston, USA, 2003.
- [2] W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Addison Wesley, 1994.
- [3] V. Jacobson, "Congestion Avoidance and Control", *SIGCOMM Symposium on Communications Architectures and Protocols*, 1988, pp. 314-329.
- [4] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm", Technical report, 30 Apr. 1990.
- [5] K. Fall, S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," *ACM SIGCOMM Computer Communication Review*, vol. 26, Issue 3, pp. 5-21, July 1996.
- [6] B. Moraru, F. Copaciu, G. Lazar, V. Dobrota, "Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno," *2nd RoEduNet International Conference*, 2003.
- [7] J. Mo, R.J. La, V. Anantharam, J. Walrand, "Analysis and comparison of TCP Reno and Vegas," *IEEE INFOCOM '99*, pp. 1556-1563, vol. 3, March 1999.
- [8] S. Floyd, "Metrics for the Evaluation of Congestion Control Mechanisms," IRTF, Informational RFC 5166, Mar. 2008.

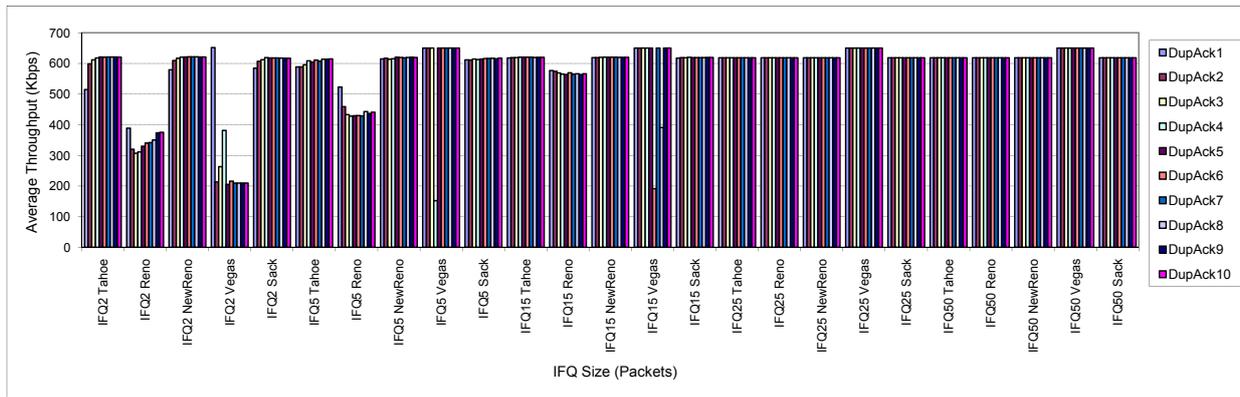


Fig. 3. Single-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (0% error ratio, DelAck=1).

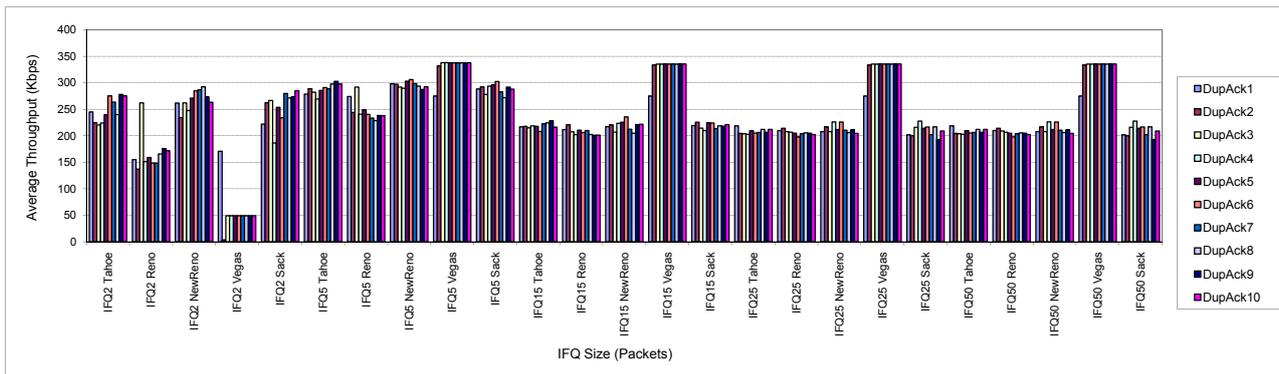


Fig. 4. Single-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (20% error ratio, DelAck=1).

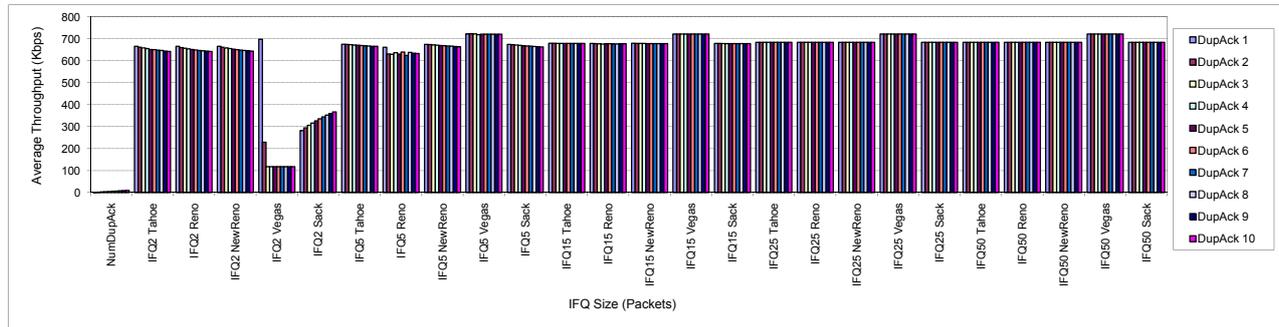


Fig. 5. Single-hop vehicular environment TCP throughput for different DupAcks and IFQ lengths (0% error ratio, DelAck=2).

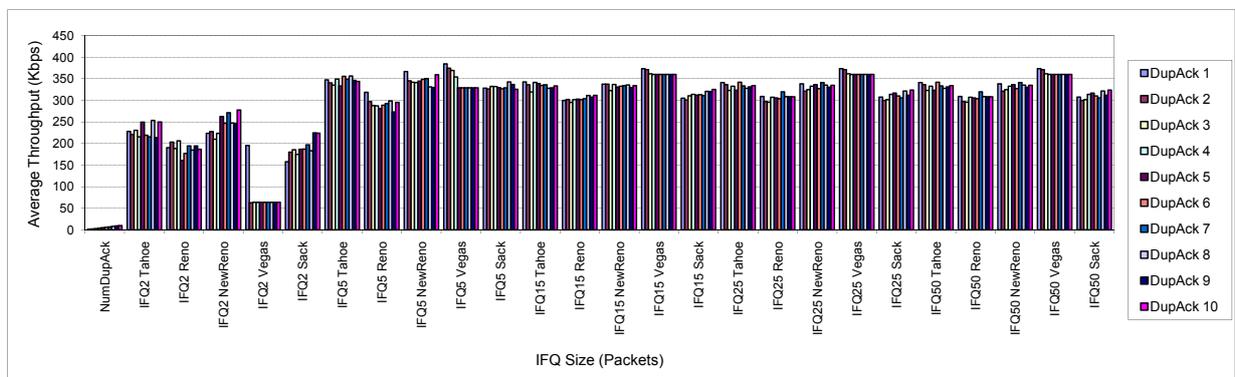


Fig. 6. Single-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (20% error ratio, DelAck=2).

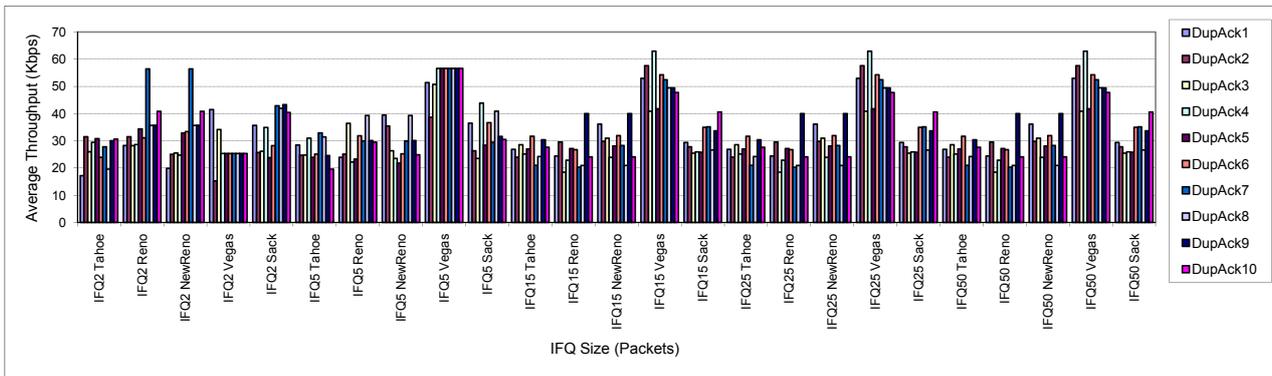


Fig. 7. Multi-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (0% error ratio, DelAck=1).

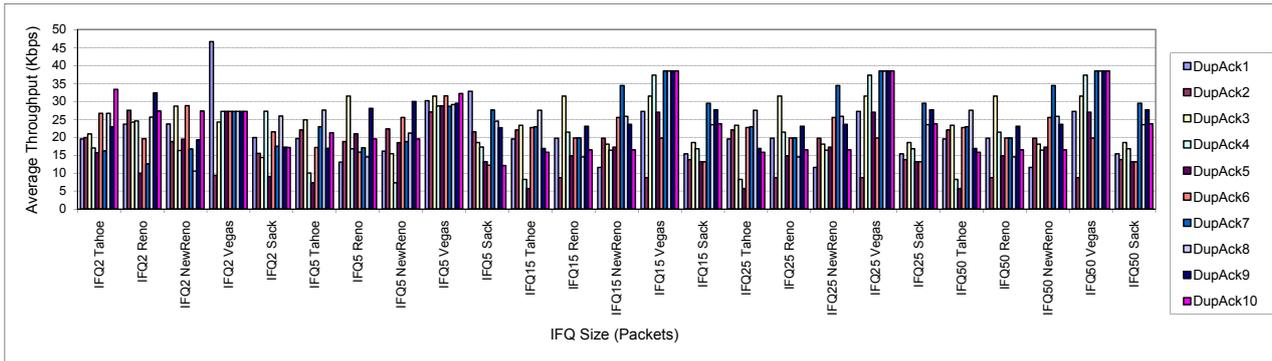


Fig. 8. Multi-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (20% error ratio, DelAck=1).

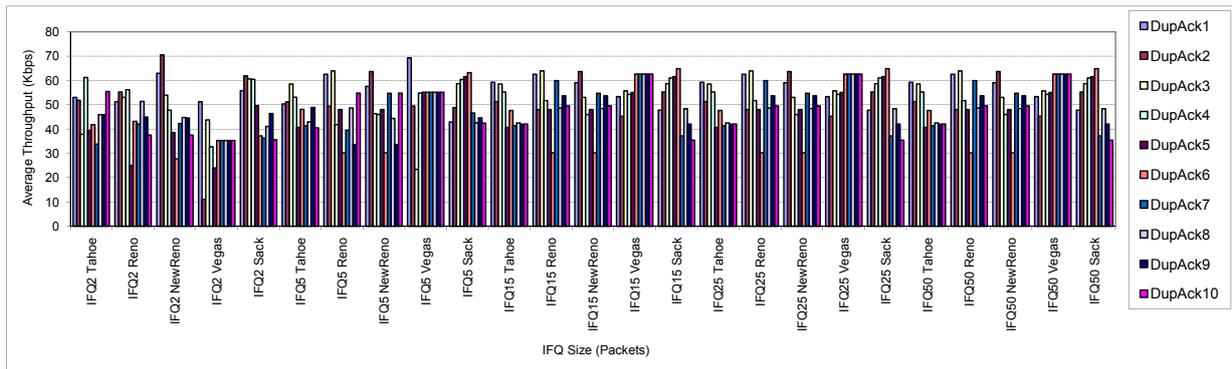


Fig. 9. Multi-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (0% error ratio, DelAck=2).

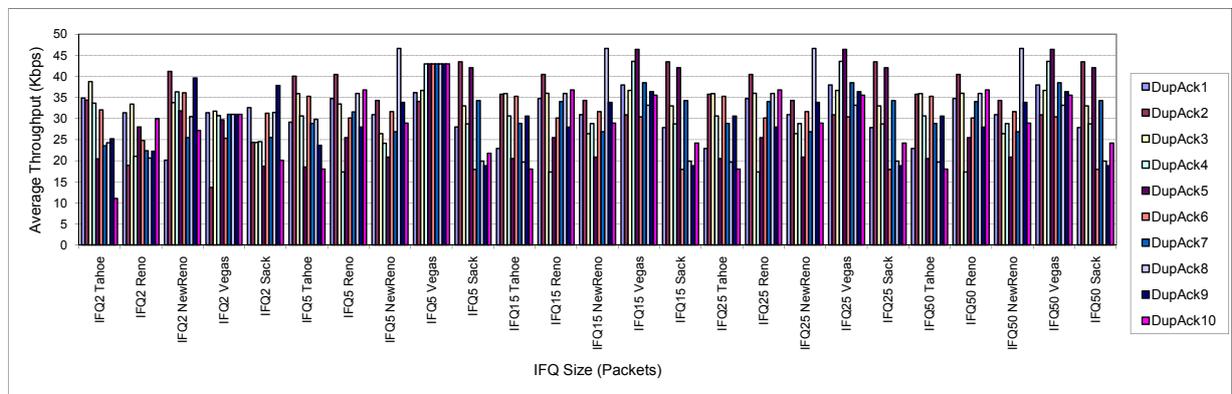


Fig. 10. Multi-hop vehicular environment: TCP throughput for different DupAcks and IFQ lengths (20% error ratio, DelAck=2).