A Survey on TCP Performance for Mobile and Wireless Networks

Rihabb Salamen and Dr. Mohammed Aldasht
Master of Informatics Department
Palestine Polytechnic University
Hebron, Palestine

Abstract: Transmission Control Protocol or shortly (TCP) is considered to be the fundamental cell for carrying the ultimate traffic of the internet, thus its performance is crucial for defining the performance of the internet. Previously, where the wired networks were the dominant in the world of the networks communications, TCP worked just fine. Actually TCP worked efficiently, since it was first introduced to serve the wired networks, and in wired network the only cause of packet losses is congestion. Wired connections are highly reliable. However, when the idea of both wireless and mobile networks were first introduced, the performance of TCP started to shake and started to hinder. Thats happened because of the TCPs inability to identify the true reason behind the losses in the transmitted packets [1]. At one hand, in a regular wired network where the links are highly reliable, the losses in transmitted packets is attributed to congestion. On the other hand, for wireless and mobile networks, the matter of packet losses is a little bit more complicated than that. Accordingly, a number of techniques were introduced to enhance the TCP performance for wireless and mobile networks. In our survey paper we attempt to summarize and discuss the problems that affect TCP performance for wireless and mobile networks, also we are going to compare the different proposed techniques to improve the TCP performance for wireless and mobile networks.

keywords. TCP, congestion control, mobile adhoc networks.

1 Introduction

As its described in [2], TCP is the most commonly used transport protocol. TCP provides congestion control and reliable delivery of data. Accordingly, TCP performs greatly in wired networks where the losses in transmission can be attributed to one and only reason, which is congestion. However, for wireless and mobile networks the equation is not straightforward. There are several factors that affect the transmission process. The above discussion has created an interesting spot for researchers. Over the past three decades, the efforts were extensively focused to provide some sort of technique that might conquer one or more of the issues that affect the performance of TCP in wireless and mobile networks. Our efforts in this paper are directed to create a survey that will include a list of these factors, plus a number of the proposed techniques. Most of our work is accomplished for single connection models. According to [3] there are three types of models, single, multiple and fluid models. The rest of this paper is divided as

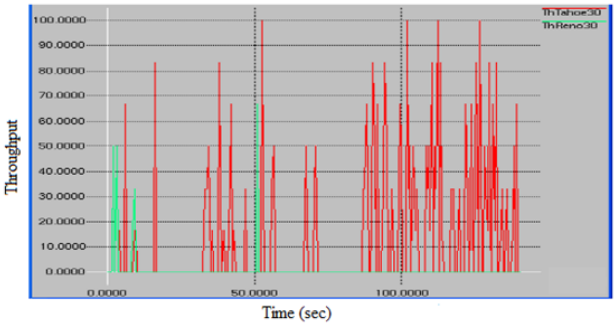

Figure 1: *Throughput of TCP Taho and Reno with loss (30% noise)*

follows: Section 2, lists the different issues that affect TCP performance. Section 3, contains a description for the suggested techniques and protocols. A summerization for the proposed techniques is given in section 4. The conclusion and further studies are provided in section 5 and 6 respectively.

2 TCP Performance Factors

Communications in wireless and mobile networks suffer greatly in comparison to wired networks. Several factors can affect the transmission in such an environment. Some of these factors are listed below:

- Random Loss [4]: The links in this environment are noisy, which will produce errors and hence, packets to be dropped. Noisy links can impact the strength and quality of transmitted signals, which in turn will produce high rate of interference and fading [5]. The authors of [4], studied the impact of the random loss caused by noisy links on the performance of TCP. The study was conducted by running a simulation for TCP variants (Reno and Taho). Figure 1 shows the performance of TCP in terms of throughput when there is 30 % noise. As it is shown in figure 1, the throughput is badly influenced with the noise. The figure shows that the performance of TCP Taho is better than Reno in case of multiple losses. Thats happened because in Taho doesnt apply fast recovery mechanism. Fast recovery involves dividing the congestion window by two each time a duplicate acknowledgment is received. When the cwnd is reduced here its value closes to zero, and thus blocking the communication. However, in Taho the window starts to increase exponentially as its suggested by the slow start.

- Handover [6]: In mobile networks, nodes are allowed to move freely. This movement implies that a node will travel from the range of a base station to another. This requires all current information to be transferred as well. This is called handover. Handover creates a disconnection or a temporary route failure that will be reestablished after a period of time. During this disconnection, each time sender fails to retransmit, it will double the value of RTO. After a number of failed attempts, the RTO value is too long and even if the link is restored before this time elapses, the sender will not try to retransmit until the expiration of the timer [7]. This is a waste of time which is in turn degrades the TCP performance [8].

- Bandwidth: The assigned bandwidth can greatly affect the performance of TCP and depends on the number of users in any given cell. The assigned bandwidth to any given node may change frequently, especially in mobile networks. In mobile networks, to protect against failures, a node can be within the range of several base stations [3]. So, when a base station fails, the node will switch to the neighboring base station which implies less bandwidth for the already connected nodes. In [6], an experiment was conducted to study the effect of load increase in a cell. The assigned bandwidth decreases when there is an increase in the number of nodes. Theyve started the experiment by having only two nodes down-

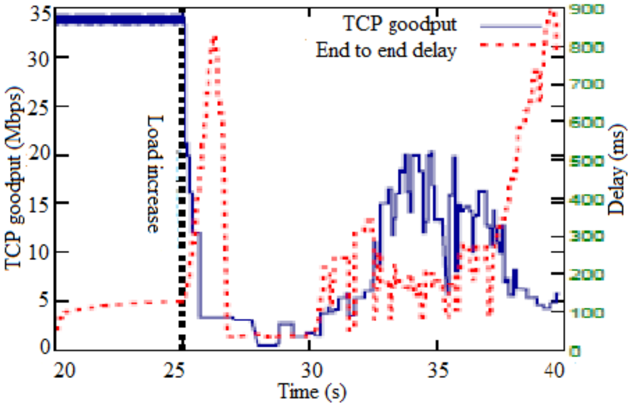

Figure 2: *Load increase results (Bandwidth reduction and max delay increase)*

loading data, then the number of nodes changed. Figure 2 shows their results. Figure 2 shows that after 25 seconds new nodes started to download data simultaneously. As the number of nodes increased the available bandwidth decreased and the delay is increased to become 850ms. Well that caused a degradation in the goodput of TCP.

- High Latencies: The higher the latencies caused by network, the more savoir the problem. Wireless and mobile networks are known to have higher latencies than wired networks [9]. This is in turn affects the time needed by a massage to reach its destination and the time for retransmitting the packets are set to high values [8].

3 Improving TCP Performance

- **Cross Layer Feedback**
  According to [10], the concept of Cross Layer Feedback is the use of the information that are provided from layers above and below in the layered protocol stack. Two mechanisms were provided. These mechanisms require little modification to the mobile host, but for the sender or any other network entities no changes are made. The first mechanism is called Receiver Window Control (RWC). The RWC employs the feedback that is provided by the user. A user feedback can be the prioritizing of applications dynamically. For instance, if we consider two running applications, like a ftp and a video conferencing application. At one hand, for the system where heuristics are used, the video conference will take the higher priority. On the other hand, when the user is involved, he might take better decisions based on experience. So, the user might anticipate a disconnection and decides that the ftp should take the higher priority in this case scenario. The user feedback is blended into the protocol stack in which the throughput of the running applications can be controlled dynamically. The authors of [10] suggested the modification of the receiver window in order to control the dynamic application priority. So, basically the RWS as [10] declares, is a method of mapping user feedback about application priorities to lower layer specific information. This method is required in the case where multiple flows are allowed, thus it means we have different delay and bandwidth factors. If a user wants a specific application to take a higher priority than the others, the user simply achieves that through the RWS. This action reflects two things: first, the advertise window to change, second, the throughput of the application will be increased as well. The second mechanism, is called ATCP. ATCP uses the feedback that is generated from the network layer at the mobile host (MH) which is related the MH stats of connectivity. ATCP supposes that a MH will send an event signal to the TCP

when its connected or when its disconnected to the network. This feedback along with the RTO information are utilized to make better changes to the TCP throughput. According to [10], the previous studies that were initiated in this field focused on enhancing the TCP performance for wireless networks, where a fixed host is transmitting data to a mobile host. However, ATCP was designed to solve for both directions, which is improving the performance of TCP for a fixed host sending to a mobile host, and vice versa.

- **Serialized Timer Approach**
  Packet transmissions for wireless networks suffer greatly in comparison to the wired networks due to different factors. One of these factors is the bandwidth which plays a major role in causing the probability for acknowledgments to arrive after the retransmission timeout. When the transmission timeout occurs the sender triggers the congestion control procedures. The congestion control mechanisms cause more delay in the transmission across the networks. Moreover, congestion control mechanisms as a result of the above discussion will degrade the performance of the network and create a massive waste in the network throughput each time it invokes the congestion control mechanisms. The Serialized Timer Approach suggests that the sender should wait longer before it decides to trigger the congestion control mechanisms. That is accomplished through the use of two timers. The first timer is the typical retransmission timer (RD). The second one is a congestion response decision timer or shortly (CD). The retransmission of packets is resolved upon the first timer RD as usual. However, deciding whether to trigger the congestion control mechanisms or not is made according to the second timer CD. The Serialized Timer Approach as its described in [8] works as the following:
  - The sender transmits a packet and initializes the RD timer.
  - The sender expects the receiver to respond with an acknowledgment. If that acknowledgment is received before the expiration of the RD timer, the sender resets the timer and the transmission continues. Now, imagine that the acknowledgment arrived after the expiration of RD timer. In the situation where the Serialized Timer Approach is not employed, this is a sign of congestion which requires the immediate triggering of congestion control mechanisms, which in turn causes the decrease in the performance of the network. However, the deployment of the Serialized Timer Approach where a second timer is used, the sender needs to halt a little longer to ensure that the cause behind the late arrival of acknowledgment was really the congestion.
  - The sender resend the packet with the late acknowledgment and sets the CD timer.
  - If the acknowledgment of the resent packet arrives before the CD expires, then the sender will conclude that the first event was due reasons other than congestion, so no need for congestion control. On the other hand, if the acknowledgment arrived after the CD timeouts, then the sender triggers the mechanisms for congestion control.

- **TCP WAM**
  The authors of [8] proposed a TCP congestion control protocol (TCP-WAM). The

center of attention for this protocol is the lost acknowledgment. Both the TCP sender and receiver side need to be changed [8]. These alterations must be injected into TCP sender side. These modifications should be able to reflect the necessary capabilities that will make the TCP sender less dependent on the properties of the opposite channel, which in this case is the TCP receiver [11]. Meanwhile, the other aspect of TCP changes must enable the TCP receiver to send some sort of notification to the sender. This notification gives the sender an indication about the previously sent data. When a sender transmits a packet, an acknowledgment should be sent back from receiver to the sender [9]. If the sender doesnt get that acknowledgment, the sender transmits the following packets and mark them as well. On the receiver side, the receiver checks the packet header for a mark. If this packet is marked, the receiver will understand that the sender didnt get an acknowledgment for the preceding packets. The receiver inspects its buffer for such packet, if it's received, the receiver will send back an acknowledgment for the packets that just arrived. Thats not all, the receiver also marks this acknowledgment with the value 1. When this marked acknowledgment reaches the sender, the sender will understand from the value 1, that the previously unacknowledged packet was delivered to the receiver indeed. The sender will conclude that the reason behind all of this was the loss of sent acknowledgment not due congestion, so no need for any measures against congestion. However, if the receiver searches its buffer, and find no sign of packet that is relevant to the mark that was placed in header of the successive packets, the receiver marks the acknowledgment with 0. When this marked acknowledgment arrives at the transmitter, the sender will realize that the timeout was due data loss, which requires the appropriate measures. TCP-WAM during retransmission timeouts doesnt initiate slow start, instead it continues with transmission of new data. In TCP-WAM protocol the congestion window size retains the same and TCP-WAM reduces the RTO. Both these actions are unlike the regular TCP.

- **Adaptive Backoff Response Approach (ABRA)**
  Each time a sent packet is unacknowledged, the TCP retransmits that packet and doubles the RTO interval, that is done by multiplying the RTO value by two. Now, this action is redone until the sender receives an acknowledgment for the resent packet. This cycle involves a continuous increase in RTO interval, that means RTO value will continue to grow each time an acknowledgment for the transmitted packet is not received. Although, the connection might have returned and fixed, the sender won't be able to send any further packets during the RTO time. In [7], a novel mechanism was introduced that attempts to make use of the wasted time. The method is called the Adaptive Backoff Response Approach (ABRA). This technique involves the use of a newly calculated retransmission timeout (RTO). In this approach, the RTO interval is built upon a new value called the smoothed round trip time (srtt) according [7]. SRTT is the weighted average of measured RTOs. The authors of [12] stated that the ABRA saves three values when the

timer for retransmission expires. The values are congestion window (cwnd), slow start threshold (ssthresh) and smoothed round trip time (srtt). So, RTO interval is multiplied by a value called backoff which is calculated according to the following equations [7]:

$$Backoff_{new} = \frac{1 + (last\_srtt - min\_srtt)}{\texttt{max\_srtt} - min\_srtt}$$

$$RTO_{new} = Backoff_{new} * RTO_{current}$$

In [7], using QualNet simulator an experiment was conducted. The behavior of different versions of TCP (Reno and New Reno) was studied in comparison to deploying the ABRA scheme with TCP New Reno (ABRA New Reno). The study was carried out using the following metrics: number of received, dropped and retransmitted packets, and throughput. They reported that ABRA New Reno works better than TCP New Reno under varying conditions of different node speed, high density node and pause time. The behavior of ABRA New Reno is stable and it achieves the best throughput than the other versions of TCP.

- **Fixed RTO Method**
  This method attempts to enable the sender to distinguish between the losses that occur due to congestion and the ones that occurs because of the temporary rout failures. It suggests the following: if the timer expires consecutively, the sender should freeze the value of RTO and retransmit the unacknowledged packet. Once the retransmitted packet is acknowledged it implies that the rout is restored. That allows the RTO to start doubling again. In [5] three routing protocol (AODV, DSR and ADV) were studied using this method. They were experimented using ns-2 simulator. Their behavior was investigated before and after implementing fixed RTO method. The results indicated that the fixed RTO technique improved the performance of TCP over OADV and DSR routing protocol. However, ADV couldnt benefit from this method.

- **SNOOP Protocol**
  The idea behind this protocol [11] is to shield the transmitter from errors that result from wireless or mobile part of the network. This is accomplished by implementing an agent at the base station. This agent catches all the unacknowledged packets and retransmits them locally. This protocol [13] has achieved speeds up in throughput of up to 20 times. It was implemented with a number of TCP versions and tested using ns-2 simulator [14]. The results shows that performance of TCP improved using SNOOP protocol. Actually it shows more robust behavior than the regular TCP.

- **Explicit Link Failure Notification (ELFN) Technique**
  This method [15] involves freezing the state of TCP sender. The RTO is not doubled and congestion window is not reduced. Meanwhile, the sender begins sending a probing massages at a regular interval. These massages are sent to sense the conditions of the disconnected rout. If the sender receives an acknowledgment for such a message, that means the broken route is restored and the sender can use it to retransmit. Now the values of cw and RTO can both change. This method was experimented using ns-2 simulator. This

technique improved the TCP performance by distinguishing between congestion losses and non-congestion losses.

4 Summery

Table 1, provides a comparison between the previously listed techniques. We have compared each mechanism as follows: basic idea behind the used method. The studied metric if there is any. The changes that are made when the technique is implemented, the changes can be made to the sender, receiver and/ or the base station. Also, we have compared the methods in terms of the tested behavior. Whether the transmission was tested from fixed host (FH) to the mobile host (MH) or in both directions. In addition, which previously listed TCP factor each mechanism has solved. At the end of the table, the list of benefits provided by each solution. Finally, the cells that contain dashes implies that this type of information is not provided or does not exist.

5 Conclusion

In wireless and mobile networks, degradation in performance of TCP can be attributed to several factors [16]. Some of these factors are listed in this paper. In addition, we presented a number of the techniques that were proposed by researchers to enhance the performance of TCP. Some of these techniques are a set of changes to the already existing methods that are deployed for handling errors and losses in wireless and mobile environment. All of the above listed techniques addressed the single connection model except for fixed RTO method, it was also tested for the multiple connections. At the end of our paper we have included a brief summary of the proposed techniques and the way in which they function and behave. Table 1, shows the eight listed techniques and their comparable features. These features include the basic idea behind each suggested technique, TCP performance factor and the benefits of each suggested method.

Technique		**Basic Idea**	**Studied Metric**	**Solved TCP Factor**	**Changes Included**	**Addressed Behavior**	**Benefits**
Cross Layer Feedback	RWS	Based on the feedback (priorities) provided by the users.	Throughput	Bandwidth issues.	Minimal modifications to the mobile host, no changes to the sender or other network entities	FH- MH.	Increased throughput in terms of dynamic incorporation of user specified priorities.
	ATCP	Based on the feedback (mobile host state of connectivity) provided by the network layer.	Throughput	Bandwidth and mobility issues.	Minimal modification to the mobile host, no changes to the sender or other network entities.	Both directions. FH-MH, and MH- FH.	Enhanced throughput of up to 40% over TCP Reno. Not dependent on the prediction of disconnection.
Serialized Timer Approach	-	Involves using an additional timer (CD) for triggering congestion control mechanisms.	-	Bandwidth issues.	Sender side changes only.	-	It gives more time to decide upon triggering congestion control mechanisms.
TCP WAM	-	Marking the sent packets and acknowledgments.	-	Latency and mobility issues.	Modifications on both sender and receiver side.	-	Does not double RTO value, which solves long waits that results from temporary link failures.
ABRA	-	Does not double RTO value, instead it is calculated using srtt.	Throughput, no of packets received,no of packets retransmitted and no of packets dropped.	Mobility issues (Temporary route failures)	Sender side modifications.	Both directions.	Achieved better throughput than Reno and New Reno. More robust behavior regarding the other studied metrics.
ELFN	-	Freezing timer and cw. Sending probing messages on a fixed time interval.	Throughput.	Mobility issues.	Sender side modifications.	FH - MH.	Distinguishes between congestion losses and non-congestion losses. Improved throughput over TCP Reno.
Fixed RTO	-	Freezing Timer.	Throughput.	Mobility issues.	Modifications on sender side.	Both directions. Also multiple TCP connections.	Improved TCP performance for OADV and DSR routing protocols but ADV could not benefit from this method.
SNOOP	-	Shielding FH from MH and unreliable nature of wireless links. That is done by implementing an agent at the base station to perform local retransmission.	Throughput.	Mobility issues.	Base station changes, while sender and receiver is not changed.	Both directions.	Achieved throughput speedups of up to 20 times over regular TCP and it shows more robust behavior.

Table 1: Comparison of Proposed Techniques.

6 Future Work

The following are the list the studies that we will conduct in the future:

1. Studying other proposed techniques.
2. A simulation based study will be conducted to compare the proposed techniques.

References

[1] S. Srinivasan B. Hari, P. Venkata and K. Randy. A comparison of mechanisims for improving tcp performance over wireless links. *IEEE/ACM Transactions on networking*, Vol. 5, December 1997.

[2] B. Gokul and M. Janise. Performance evaluation of variable bucket size coding with tcp for multimedia streaming over manet routing protocols (wip). 2014.

[3] H. Florian. Performance evaluation of tcp flows. 2014.

[4] Q. Shamimul and M. Kumar. Impact of random loss on tcp performance in mobile ad-hoc networks (ieee 802.11): A simulation analysis. *International journal of computer science and information security (IJCSIS)*, Vol.7, 2010.

[5] A. Elan B. Hari, S. Srinivasan and K. Randy. Improving tcp/ ip performance over wireless networks. *1st ACM international conference on mobile computing and networking (Mobicom)*, Nomvember 1995.

[6] G. Vijay N. Binh, B.Arijit and K. Aneha. Towards understanding tcp performance on lte/ epc mobile networks acm. *International Journal of Computer Networks Communications (IJCNC)*, 2014.

[7] B. Dhananjay and S. Sanjeev. Improve performance of tcp new reno over mobile ad-hoc network using abra. *international journal of wireless and mobile networks (IJWMN)*, 3, April 2011.

[8] N. Amiya F. Breeson, N. Venkat and S. Ivan. Techniques for enhancing tcp performance in wireless networks. 2012.

[9] K. Nikhil D. Purvang and K. Dasgupta. Improving tcp performance over wireless network with frequent disconnections. *International Journal of Computer Networks Communications (IJCNC)*, Vol.3, November 2011.

[10] S. Ajay R. Vijay and I. Sridhar. Improving tcp performance over mobile wireless environments using cross layer feedback. *International Journal of Computer Networks Communications (IJCNC)*, 2003.

[11] E. Hala. Improving tcp performance over mobile networks. *ACM computing survey*, Vol. 34:357–374, September 2002.

[12] M. Hadhoud F. Ghanem, S. Elkilani. Improving tcp performance over mobile ad hoc networks using an adaptive backoff response approach. *IEEE*, pages 978–1–4244–3778, September 2009.

[13] C. Manish and K. Avichal. Performance evaluation of tcp variants in wi-fi networks using cross layer design protocol and explicit congestion notification. *International Journal of Wireless and mobile networks (IJWMN)*, Vol. 3, June 2011.

[14] D. Thomas and B. Rajendra. A comparison of tcp performance over three routing protocols for mobile ad hoc networks. *ACM Symposium on mobile ad hoc networking and computing*, October 2001.

[15] H. Gavin and V. Nitin. Analysis of tcp performance over mobile ad hoc networks. Vol. 34:275–288, 2002.

[16] T. Hellen I. A. Rai. On improving the tcp performance in asymmetric wireless mesh networks. 2011.