

# PERFORMANCE EVALUATION OF SELECTED E2E TCP CONGESTION CONTROL MECHANISM OVER 4G NETWORKS

Mohanad Al-Hasanat<sup>1</sup>, Abdullallah Al-Hasanat<sup>1</sup>, Saud Althunibat<sup>2</sup>, Ali AlSbou<sup>3</sup>

<sup>1</sup>Department of Computer Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan

<sup>2</sup>Department of Communications Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan

<sup>3</sup>Department of Computer Information System Al-Hussein Bin Talal University, Ma'an, Jordan

## ABSTRACT

*TCP is one of the main protocols that govern the Internet traffic nowadays. However, it suffers significant performance degradation over wireless links. Since wireless networks are leading the communication technologies recently, it is imperative to introduce effective solutions for the TCP congestion control mechanisms over such networks. In this research four End-to-End TCP implementations are discussed, they are TCP Westwood, Hybla, Highspeed, and NewReno. The performance of these variants is compared using LTE emulated environment in terms of throughput, delay, and fairness. Ns-3 simulator is used to simulate the LTE networks environment. The simulation results showed that TCP Highspeed achieves the best throughput results. Although TCP Westwood recorded the lowest latency values comparing to others, it behaved unfairly among different traffic flows. Moreover, TCP Hybla demonstrated the best fairness behaviour among other TCP variants.*

## KEYWORDS

*TCP congestion control; NewReno; 4G; HighSpeed TCP; TCP Westwood; Hybla; ns-3; LTE.*

## 1. INTRODUCTION

Nowadays, the Internet has reached many parts of the world, even most of the remotest places on the planet. Consequently, people are connected through their smart devices most of the times. The extensive development of new communication technologies and the needs of these technologies is the motivation for a large spectrum of networking researchers who seek to enhance communication system's hardware and software infrastructure [12]. Rapidly over the past two decades, spacious communications applications and services have been developed, especially in wireless communications. Long Term Evolution (LTE) mobile network, standardized by the 3GPP is one of the most recently used technology in current mobile communication systems. LTE mobile network aims to provide high spectral efficiency, high peak data rates, low latency, better coverage, and low operating cost [11].

One main fact is that most of the current Internet traffic is managed and controlled by the Transmission Control Protocol (TCP). The TCP traffic uses a window-based congestion control mechanism. This mechanism endeavour to equalize each TCP flow's sending rate with the link's fair share, also responding to the congestion link by reducing the flow sending rate. The TCP

congestion control mechanisms are well designed and optimized to perform efficiently over wired links. However, these mechanisms experience severe performance degradation over wireless links [7, 8].

To this end, several TCP implementations were introduced to enhance the TCP congestion control mechanism over wireless links. Mainly, these enhancements could be classified into three main categories; Link-Level solutions (e.g. I-TCP, M-TCP, etc.), split connection solutions (e.g. Automatic Repeat Request (ARQ), Hybrid ARQ, etc.), and End-to-End solutions (E2E) including TCP Westwood [1], TCP Hybla [2], TCP Highspeed [3], TCP Vegas, TCP VenO, and many other TCP implementations.

E2E solutions attempt to control the TCP traffic based on the Transport Layer knowledge on both the connection's ends (source and destination). Thus, these schemes don't require the cooperation of intermediate nodes. E2E solutions handle the network as a black-box; therefore, these solutions maintain the end-to-end semantics of the original TCP. In E2E solutions, either the sender or the receiver controls the traffic's flow based on the other end's notifications. Thus, these solutions try to accurately probe the available link bandwidth to achieve better performance. Most of the proposed E2E solutions try to predict the network's bandwidth using the receiver feedback, however, calculating the link's bandwidth is still a great challenge.

In this paper we compare four E2E TCP congestion control mechanisms, they are NewReno, Westwood, Highspeed, and TCP Hybla. Their performance is evaluated over one of the leading technologies of the 4G networks, the LTE, using ns-3 simulator. Three different performance metrics are used to evaluate the tested TCP implementations, these metrics are total throughput, latency, and fairness.

In the next section, four different E2E TCP congestion control mechanisms will be discussed, including Westwood, NewReno, Hybla and TCP Highspeed. The main metrics to compare the performance of these TCP variants are given in section 3. Section 4 introduced the simulation model and showed the simulation results. Finally, a conclusion is drawn in section 5.

## **2. E2E TCP VARIANTS**

### **2.1. TCP NewReno**

Mainly, most of the proposed E2E TCP congestion control mechanisms are based on controlling the source sending rate to match the network capacity (i.e. the capacity of both, the destination and the intermediate nodes). However, they differ in how the available link bandwidth is predicted. Many solutions were introduced with different mechanisms to probe the network status. The first was TCP Tahoe. A very simple mechanism to predict a congestion condition whenever a packet loss event occurs. Later, a new TCP variant was introduced, TCP Reno. Reno uses four distinct congestion control mechanisms they are Slow Start, Congestion Avoidance, Fast Retransition and Fast Recovery.

Later, a third improvement was introduced in [6], NewReno. The new variant improves Retransmission Timeout (RTO) calculation and the back-off procedure whenever a packet dropped. Moreover, NewReno uses Additive Increase Multiplicative Decrease (AIMD) congestion avoidance with Slow Start. In addition, NewReno modified the Fast Recovery mechanism to solve the problem of multiple packet loss within a single window of a data stream. In spite of these modifications, the only way that the three variants can detect congestion is by creating a congestion. As the three variants carefully probe for congestion by slowly increasing their sending rate to create congestion, then they cut down their sending rate. The next subsection

introduces another TCP variant that predicts the available link's bandwidth without creating congestion.

## 2.2. TCP Westwood

TCP Westwood introduced a new revolution in TCP congestion control mechanisms. It has introduced the first E2E bandwidth estimation mechanism using the following formula:

$$BWE = \frac{acked * MSS}{t_k - t_{k-1}} \quad (1)$$

Where  $BWE$  is the estimated bandwidth,  $acked$  is the number of the acknowledged packets between  $t_k$  and  $t_{k-1}$  period, and  $MSS$  is the maximum segment size.

Westwood uses the calculated bandwidth to properly set both, the congestion window ( $cwnd$ ), and the k8slow start threshold ( $ssthresh$ ) values. Whenever a packet drop event or three duplicate Acknowledgments ( $3dupACK$ ) arrive, Westwood set the  $ssthresh$  value to the calculated  $BWE$  value. Therefore, Westwood never triggers congestion mechanism unless its  $cwnd$  value is greater than the  $BWE$  value. The following pseudo-code shows how it works:

```

if (n DUPACKs are received OR RTO expires)
    ssthresh = (BWE*RTTmin)/seg_size;
if (RTO expire)
    cwnd=1;
else
    if (cwin>ssthresh)
        cwnd = ssthresh;
    endif
endif
endif

```

Westwood's bandwidth estimation mechanism is particularly effective over wireless networks. As sporadic packet losses due to radio channels are misinterpreted as a congestion symptom in other TCP variants. A throughput improvement of up to 550% rather than NewReno has been shown in [1].

## 2.3. TCP Hybla

TCP Hybla presented a promising solution to the performance disparity in heterogeneous networks due to their different Round Trip Times ( $RTT$ ). Unlike other TCP variants, Hybla is based on the  $cwnd$  evolutions as a function of connection delays. Thus, it attempts to obtain the same instantaneous transmission rate of a fast reference TCP connection to use it for long  $RTT$  connections.

$$W_{i+1}^H = \left\{ \begin{array}{l} W_i^H + 2^p - 1, SS \\ W_i^H + \rho^2 / W_i^H, CA \end{array} \right\} \quad (2)$$

Where  $\rho = RTT / RTT_0$ , and  $RTT_0$  is the reference TCP connection [2].

Based on equation (2), Hybla increments its  $cwnd$  by  $2^p - 1$  during the slow start phase, and  $\rho^2 / cwnd$  during congestion avoidance phase. Thus, this aggressive incrementing of the  $cwnd$  in the slow start phase could result in bursts of losses. Therefore, TCP Hybla recommends using SACK option at the receiver end to support multiple packets losses.

Hybla shows significant improvement in TCP performance over long *RTT* connections (e.g. satellite connection). Moreover, it shows fair sharing of the available link's bandwidth and friendly behaviour with other TCP variants as reported in [2].

## 2.4. Highspeed TCP

Another modification to the standard TCP congestion control mechanism for use with large bandwidth channels was introduced in Highspeed TCP [3]. The proposed variant addressed the TCP limitation of reaching the steady-state in very long time. Highspeed TCP introduced a fundamental change to the standard TCP response function at the sender side only. Actually, the proposed modification suggested to enable TCP connections to start with large *cwnd* value using the following formula:

$$W = (p / low\_p)^{S * low\_Window} \quad (3)$$

According to [3],  $p$  is the error rate,  $low\_p$  is set to  $10^{-3}$ ,  $low\_window$  is set to 38,  $High\_Window$  is set to 83000,  $High\_p$  is set to  $10^{-7}$ , and  $S$  is equal to:

$$S = \frac{(\log High\_Window - \log Low\_Window)}{(\log High\_p - \log Low\_p)} \quad (4)$$

Thus, over high error rate connections, Highspeed TCP's response function will be quite close to that of the standard TCP response function. However, over low loss ratio connections, Highspeed TCP uses larger *cwnd* size comparing to the standard TCP *cwnd* size.

## 3. METHODOLOGY

To evaluate the performance of the selected TCP variants, this study suggests using ns-3 simulator to capture and analyse specific network performance metrics including throughput, latency (average delay), and fairness. Thus, two main topologies were used to emulate a standard LTE network environment. The two topologies are based on LENA LTE model [4]. The first topology uses a simple LTE-EPC network architecture with one user equipment (UE) and one Evolved-Node Base station (eNB). Different parameters were used to test the TCP congestion control mechanisms behaviour in different scenarios. In the second topology, we have extended the previous topology to include more UEs to test the TCP fairness among 10 different TCP connections flows. The two topologies and their parameters are discussed in more details in the experiment section. Next, we will discuss the selected network performance metrics.

### 3.1. Throughput

A network throughput is the aggregate data that is successfully received to all the network's terminals. Throughput is an important performance indicator for most transmission protocols [10]. One main goal to achieve is to maximize the network throughput. To do so, both the connection ends should perfectly utilize the link capacity.

Mainly, there are two ways to calculate the network throughput, the indiscriminate throughput and the goodput. The indiscriminate throughput measures the total bits received at a destination within the connection period. On the other hand, goodput excludes the retransmitted bits, downstream dropped packets, and the packets fragments. In this research, we will calculate the indiscriminate throughput as in the following equation:

$$\text{Throughput} = \frac{\sum \text{received\_bits}}{\text{connection\_period}} \quad (5)$$

### 3.2. Delay

Delay can be measured in term of per-packet transfer latency in millisecond (ms). This time includes nodal processing time and transmission delay. In this research we will calculate the average delay as the summation of all the latencies for all the received packets divided by the total number of the received packets using the following equation:

$$\text{Delay} = \frac{\sum_{i=1}^n \text{delay\_of\_packet}}{n} \quad (6)$$

### 3.3. Fairness

Fairness is a very important metric used to evaluate the performance of many systems. Fairness determines whether users are receiving a fair share of the system resources or not. TCP Fairness requires that all the comparable TCP flows fairly share the link's bandwidth. Mostly, TCP unfairness problem results when different TCP flows with different RTTs share the same bottleneck.

One method of testing TCP Fairness is using Jain's Fairness Index [5]. This index is bounded between 0 and 1. For any set of flows' throughputs ( $T_1, T_2, \dots, T_n$ ) Jain's index is given as in the following formula:

$$\text{Fairness}(\text{throughput}) = \frac{\left( \sum_{i=1}^n T_i \right)^2}{n \left( \sum_{i=1}^n T_i^2 \right)} \quad (7)$$

If all users received an equal number of bits, then Jain's index will equal to 1. Where 0.90 Jain's index means 90% of the bottleneck bandwidth is shared fairly among the TCP connections.

## 4. SIMULATION AND RESULTS

As discussed before, ns-3 simulator will be used to compare the performance of the selected TCP variants over an emulated LTE environment. One stable LTE model is presented in LENA project [4]. The model provides an open source LTE/EPC network simulator based on the popular ns-3 simulator. Fig. 1 shows a simple LTE/EPC network consists of one UE, an eNB, one SGW, one MME, and a remote host.

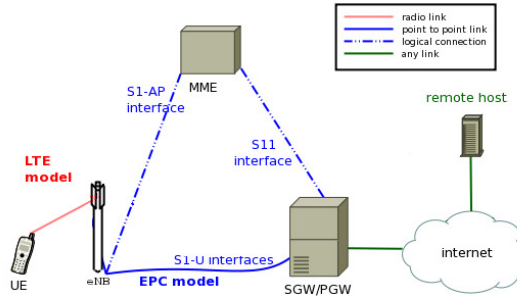


Figure 1: Long Term Evolution architecture based on LEAN model [4]

For the first experiment, we have used the same simple model shown in Fig. 1. Table 1 lists the first experiment scenario.

Table 1. The first experiment scenario

Parameter	Value
Number of UEs	1
Number of eNBs	1
Packet Size	1500 bytes
Queue scheme	Drop-Tail
Simulation period	100 seconds
Packet Error Rate (PER)	0.001, 0.005, 0.01, 0.05
TCP Protocols	NewReno, TCP Westwood, Hybla, and Highspeed TCP
Mobility	Fixed position
Simulation period	100 seconds

A **BulkSend** built-in class is used to generate a single flow of traffic starting at the remote host and destined for the UE along the simulation period. To simulate the wireless link over the UE-eNB link, the built-in ns-3's **RateErrorModel** built-in class is used to generate random transmission errors during the simulation period. Table 2 summarizes the first experiment links parameters.

Table 2. The first experiment links specifications. These values are based on the LTE standard specifications of the 3GPP [9].

Link	Transmission rate	Propagation Delay
Remote host-SGW/PGW	1 Gbps	0.03 ms
eNB-SGW/PGW & MME	1 Gbps	3 ms
UE-eNB-Download Link	25 Mbps	5 ms

The same parameters listed in the Table2 are used to calculate the throughput values for the four TCP variants. Moreover, the throughput is calculated again using different PER values as listed in Table 1. The throughput results are shown in Figure2.

Figure2 shows that TCP Highspeed achieved the largest throughput values comparing to other implementations. Whereas, TCP NewReno recorded the lowest throughput values. In addition, it is clear that all TCP variants are affected by the changing of the PER values. More specifically, NewReno performance is seriously degraded over high PER channels.

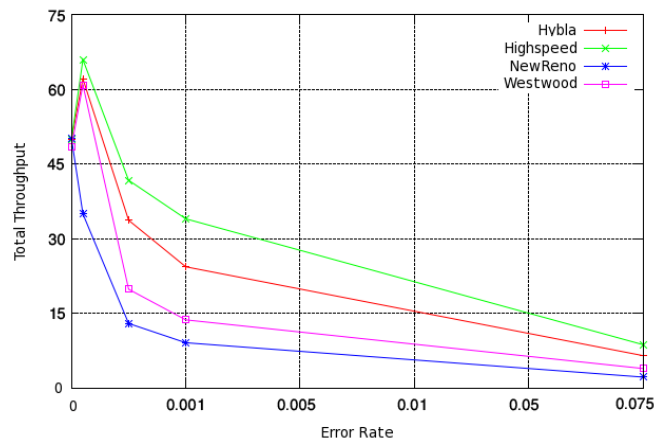


Figure 2. The throughput values for TCP Westwood, TCP Hybla, TCP Highspeed, and TCP NewReno in different PER values.

On the other hand, the average delay values are recorded and plotted in Figure 3. The results showed that TCP Westwood reported the lowest delay values except when PER is equal to 0.001. This result could be explained as NewReno works perfectly over wired-like links where a very low transmission error is expected. However, NewReno recorded the longest delay values for other PER values, and Hybla recorded the highest latency among other TCP variants when PER is equal to 0.001.

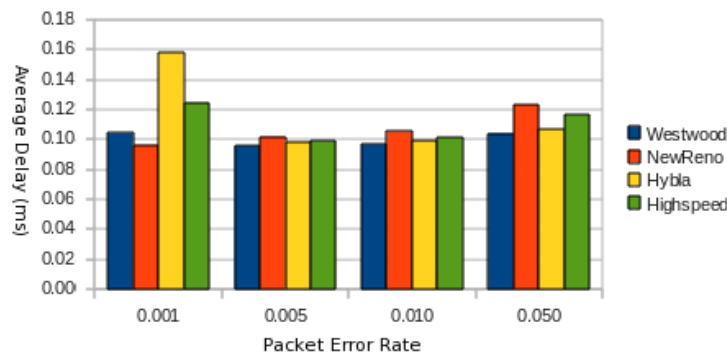


Figure 3. This figure shows the average delay in milliseconds for NewReno, Hybla, Highspeed, and Westwood TCP variants using the same bandwidth with different PER values.

A different topology is used to test the selected TCP variants fairness. The new topology uses the same specifications listed in the previous experiment but extending the UE number to 10 users. Thus, the throughput values for every flow are recorded, and then Jain's index for every TCP

variant is calculated based on equation (7). Figure. 4 shows a graphical representation of every flow throughput.

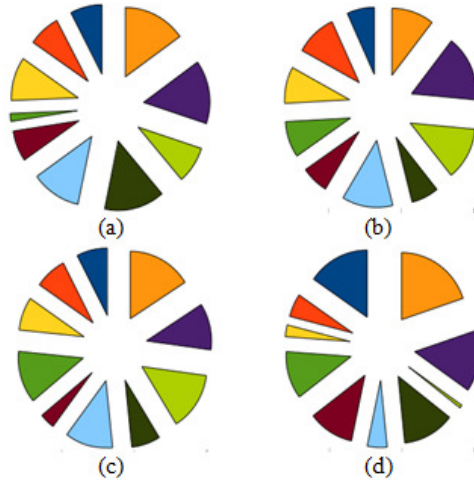


Figure 4. The throughput of the 10<sup>th</sup> flows using four different TCP variants; NewReno in (a), Hybla in (b), Highspeed TCP in (c), and Westwood in (d).

The above figure clearly shows that TCP Hybla Figure. 3 (b) almost achieves fair share among all the 10<sup>th</sup> flows. TCP Highspeed comes in the second place. However, TCP Westwood shows the worst fairness behaviour among others. Table 3 shows Jain’s fairness index for the four TCP variants.

Table 3. Jain's Fairness index for the selected four TCP variants

TCP Variant	Jain's index
TCP Westwood	0.744
TCP Hybla	0.92
TCP Highspeed	0.88
TCP NewReno	0.86

## 5. CONCLUSIONS AND FUTURE WORKS

In this study, the performance of four E2E TCP congestion control mechanisms over LTE network was discussed. The selected TCP variants are TCP Westwood, TCP Hybla, TCP Highspeed, and TCP NewReno. Three main network performance metrics were selected to compare the four TCP implementations. These metrics are throughput, delay, and fairness. Ns-3 simulator was used to collect and monitor the selected TCP behaviours over LTE emulated environment. LENA LTE/EPC model was adopted to simulate the LTE network environment.

The simulation results showed that TCP Highspeed achieved the best throughput values. In addition, TCP Westwood attained the lowest latency values comparing to other TCP implementations. On the other hand, the best fairness behaviour is demonstrated by the TCP Hybla.



For future works, other TCP variants could be selected to test their performance. In addition, more experiment scenarios could be used such as, including the mobility feature for UE. Other performance metrics could be also useful such as friendliness, and loss ratio.

## REFERENCES

- [1] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. Y., & Wang, R., "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," ACM SIGMOBILE, pp. 287-297, 2001.
- [2] Carlo Caini and Rosario Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," International Journal of Satellite Communications and Networking, vol 22, pp. 547-566, 2004.
- [3] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC:3649, 2003.
- [4] LENA documentation, 2011. Available from: <http://lena.cttc.es/manual>, retrieved on 27th August 2017.
- [5] Jain, R.K., Chiu, D.W., Hawe, W.R. "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," ACM Transaction on Computer Systems, 1989.
- [6] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 3782, 2004.
- [7] Al-Hasanat, M., Seman, K., Saadan, K., "Enhanced TCPW Slow Start Phase," Transactions on Networks and Communications, vol. 2 No 5, ISSN:2054-7420, pp. 194-200. Oct 2014.
- [8] Al-Hasanat, M., Seman, K., Saadan, K., "Enhanced TCPW's Fast Retransmission and Fast Recovery Mechanism over High Bit Errors Networks," 2015 International Conference on Computer, Communications, and Control Technology. 21-23 April 2015. Malaysia, Kuching.
- [9] 3GPP, "Technical Specifications and Technical Reports for a UTRAN-based 3GPP system (Release 8)," TS21.10, v8.0.0, 2009.
- [10] F. Beritelli, C. Rametta, A. Raspanti, M. Russo, F. Scaglione, G. Spallina, "An Advanced OoS Analysis and Evaluation Method for Mobile Internet Access," International Journal of Wireless & Mobile Networks, vol. 8 No. 3, pp. 55-70. June 2016.
- [11] F. Afroz, R. Heidery, M. Shehab, K. Sandrasegran, S. Shompa, "Comparative Analysis of Downlink Packet Scheduling Algorithms in 3GPP LTE Networks," International Journal of Wireless & Mobile Networks, vol. 7 No. 5, pp. 1-21. October 2015.
- [12] Abdullah Alhasanat, K. Abdulkader and M. Ahmed, "Range-Based Data Collection with a Mobile Sink in Wireless Sensor Network," International Journal of Wireless and Mobile Networks IJWMN Volume7, Issue 6, December 2015.