 ISSN: 2321-8169
2142 - 2145

Analysis of COMPOUND TCP using Network Emulation

Mr. Harshit B. Parmar
BE Student, Department of Electronics & Communication Engineering,
C.U.Shah College of Engineering and Technology,
Wadhwan, Gujarat, India
hper009@gmail.com

Asst. Prof. Kapil S. Raviya
Assistant Professor, Department of Electronics & Communication Engineering
C. U. Shah College of Engineering and Technology,
Wadhwan, Gujarat, India
raviyakapil@gmail.com

Abstract— TCP/IP is the main and most widely used transport protocol for reliable communication. Actual use of TCP/IP protocol on wireless links in the Internet has found some serious performance issues. Different versions of the TCP have been planned to improve the performance of data transmission. Due to its widespread need, researchers have been proposing and studying new various TCP variants trying to improve its behavior. Different variants of TCP named TAHOE, RENO, NEW RENO, COMPOUND TCP, TCP VEGAS, HS (High Speed) TCP etc. Network Simulation and Emulation are widely used to Develop, Test and Debug new protocols, to explore and study a specific Network related research issue, or to evaluate the network performance of an existing protocol. We will compare the performance of various TCP variants in terms of different parameters and the final results can provide further insight into the advantages and drawbacks of TCP variants.

Keywords- *Congestion window, Corruption, Throughput, Fast retransmit, Fast recovery, Slow Start Threshold, RTT, DUPACK*

*****

I. INTRODUCTION

Internet traffic is basically made of up of small data bursts called packets. These packets contain information about the origin and destination of the data. The packets are created and reassembled by the TCP protocol and sent over the Internet by the Internet protocol.

Originally, TCP was developed for wired links as wired links have low chance of high delay and data corruption due to external parameters. Congestion is the main reason of packet loss on wired links. So, TCP was designed by keeping in mind the above parameters. Wireless links have some problem of variable and high delay with high Bit Error Rate (BER). So initially, unmodified old TCP started to perform badly on wireless links. To deal with the problems of wireless links, a research started in the field of TCP and modifications were done as per the requirements to improve the overall performance. Different variants called Tahoe, Reno, New Reno and SACK and many more came into existence. The Transmission Control Protocol is a reliable connection oriented end-to-end protocol. TCP ensures reliability by starting a timer whenever it send a segment then If it does not receive an acknowledgement from the receiver within the "time-out" interval then it retransmits the segment. We shall take brief look at each of the congestion avoidance algorithms and see how they differ from one another.

II. BASICS OF TCP VARIANTS

A. TCP Tahoe

TCP Tahoe is one of the variant of TCP congestion control algorithm that is suggested by Van Jacobson. He had added some new improvement on the TCP completion in the early stage, that enhance consists congestion avoidance, slow start and fast retransmission.

Tahoe TCP detects packet loss by simply timeouts and then retransmits the lost packets. Packet loss is taken as a sign of congestion and Tahoe TCP saves the half of the current window as ssthresh value. Then it set cwnd to one and starts slow start until it reaches the threshold value. Then it increment linearly until it encounters a packet loss. So it increase it window slowly as it approaches the bandwidth capacity.

Table no. 1 Evolution of TCP Variants

TCP Variant	Year
TCP	1974
TCP Tahoe	1988
TCP Reno	1990
TCP Vegas	1994
TCP New Reno	1995
Sack TCP	1996
High-speed TCP	2003
Compound TCP	2006

Limitations of Tahoe:

The problem with Tahoe is that it takes a complete timeout interval to detect a packet loss. Actually, in most implementations it takes even longer. As it doesn't send immediate ACK's, it sends cumulative acknowledgements; so it follows a 'go back n' approach. Hence when every time a packet is lost, it waits for a timeout and the pipeline is emptied. It offers major cost in high band-width delay product links. Tahoe TCP does not deal properly with multiple packet drops within a single window of data.

B. TCP Reno

Reno TCP is defined as a TCP which contains the slow start, fast retransmit, fast recovery and congestion avoidance

ISSN: 2321-8169
2142 - 2145

algorithms. Reno TCP retains the basic principle of Tahoe, which is slow starts and the coarse grain re-transmit timer. Though it adds some intelligence over it, lost packets are detected earlier and the communication path (pipeline) is not emptied every time a packet is lost. Reno TCP requires that we receive immediate acknowledgement whenever a segment is received. Actually, basic logic behind this is that whenever we receive a duplicate ACK, then his duplicate ACK could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost.

If we had received number of duplicate acknowledgements then that means that sufficient time have passed and even if the segment had taken a longer path, it must have gotten to the receiver by now. So, there is very high probability that it was lost. Hence Reno TCP suggests an algorithm known as Fast Re-Transmit.

Problems:

Reno performs very well over TCP for the small packet losses. But during multiple packet losses in one window, RENO doesn't perform well and its performance is almost the same as Tahoe during high packet loss because it can only detect a single packet loss. If multiple packets are dropped then the first information about the packet loss comes when we receive the duplicate Acknowledgements. But the information about the second packet which was lost will come only after the ACK for the retransmitted first segment reaches the sender after one RTT.

C. Compound TCP

Compound TCP is a TCP variant protocol offering congestion control solution for high-speed and long distance networks. The key idea of CTCP is to add a scalable delay-based component to standard TCP. This delay-based component has a scalable window increasing rule that not only can efficiently use the link capacity, but can also react early to congestion by sensing the changes in RTT. If a bottleneck queue is sensed, the delay based component gracefully reduces the sending rate. This way, CTCP achieves good RTT fairness and TCP fairness.

Compound TCP(C-TCP) is widely deployed as it is the default transport layer protocol in the Windows operating system. The Compound protocol aims to use both queuing delay and packet loss as feedback to regulate its flow and congestion control algorithms. Compound maintains both cwnd (the loss window) and dwnd (the delay window). The loss window is the same as in the standard TCP Reno algorithm, which aims to control the loss based component. CTCP can efficiently use the network resource and achieve high link utilization. In theory, CTCP can be very fast to obtain free network bandwidth, by adopting a rapid increase rule in the delay-based component, e.g. multiplicative increase. CTCP has similar or even improved RTT fairness compared to regular TCP. This is due to the delay-based component employed in the CTCP congestion avoidance algorithm. It is known that delay-based flow, e.g. Vegas, has better RTT fairness than the standard TCP CTCP has good TCP-fairness. By employing the delay based component, CTCP can gracefully reduce the sending rate when the link is fully utilized. In this way, a CTCP flow will not cause more self-induced packet losses than a standard TCP flow, and therefore maintains fairness to other competing regular TCP flows.

III. FUNDAMENTALS OF EMULATION

Simulation is a technique where components reproduce a timing behavior similar or equal to the timing behavior of the simulated targets (simulated entities). During the development, it interacts with the simulated environment in the same way it would interact with a real one. This allows testing it in different environments with relatively little effort.

Network emulation and simulation are widely used to develop, test, and debug new protocols, to explore and study a specific network-related research issue, or to evaluate the performance of an existing protocol or a scheme. Network emulation is the execution of real network protocol implementation code in a controllable and reproducible laboratory network environment. Unlike network simulation, the protocols and applications as well as the interaction between protocols are "real". Network traffic physically traverses the emulation environment, in which underlying protocols are tested and evaluated against user defined network conditions and traffic dynamics, such as packet latency, link bandwidth, packet drop rate, Bit Error Rate (BER), and link failure Network emulators are important tools for doing research and development related to network protocols and applications. With network emulation it is possible to perform tests of realistic network scenarios in a controlled manner, which is not possible by only using real network devices without emulation capabilities.

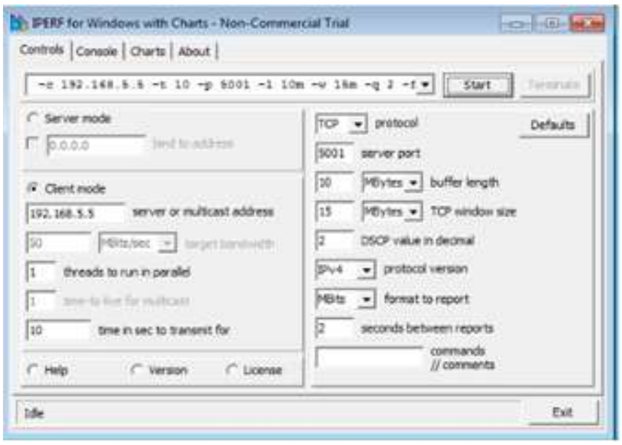

Fig. 1 IPERF (GUI Based Version)

Iperf is a frequently used for network testing and that can create Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) data streams and can measure the

 ISSN: 2321-8169
2142 - 2145

throughput of a network that is carrying them. Iperf tool is used to measure the quality of a network link. Iperf is network performance measurement tool which is written in C. Various parameters can be set that can be used for testing a network, or alternatively for optimizing or tuning a network. The network link is delimited by two hosts running Iperf. The quality of a link can be tested as follows:

- Ping command can be used to measure the Latency (Response time or RTT).
- Iperf UDP test can be used to measure jitter (latency variation) and Datagram loss.
- The bandwidth is measured through TCP tests.

Iperf tool has a server and client functionality, and that can measure the throughput between the two systems, might be unidirectional or bi-directionally. One host must be set as client, the other one as server. Iperf tool is open-source software that runs on various platforms including Linux, Unix and Windows. Basically, IPERF is a command based tool, but I am using GUI based version, that can be run at client side or server side.

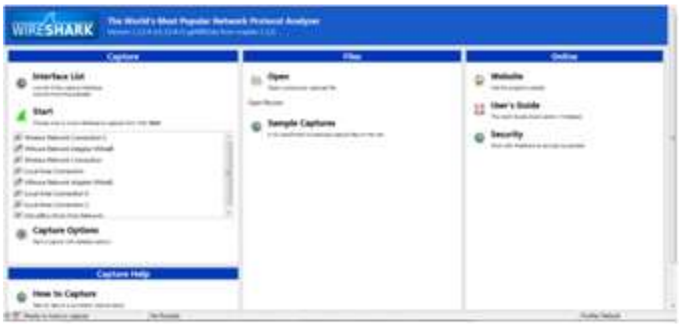

Figure 2 Wireshark Packet Analyzer

Wireshark is world's very popular network protocol analyzer. It is very powerful tool which provides network and upper layer protocols information about data captured in a network. It has a powerful rich feature set and runs on main computing platforms including Windows, Linux, and UNIX. Network professionals, security experts, developers, and educators around the world use it regularly. It is open source, and is released under the GNU (General Public License) version 2. It is developed and maintained by a worldwide team of protocol experts. Wireshark formerly used to be known as Ethereal. The Wireshark strength comes from:

- It is very easy to install.
- It is simple to use due to GUI interface.
- It has very high number of functionality available.

Wireshark is a free packet sniffer computer application. It is used for network troubleshooting, analysis, software and communications protocol development, and education. It works in Promiscuous and Non-promiscuous mode. In Promiscuous mode, NIC can see conversation to and from all of its neighbors. Wireshark displays capture information in three main panes. Its default fields include: Packet number, Time, Source address, Destination address, Name and information about protocols.

IV. EMULATION RESULTS

We have connected two systems, a laptop and a pc using LAN cable. Both the systems have windows 7 installed on it. We have used the IPERF tool to measure the throughput between these two systems, and the packets are captured by the Wireshark packet analyzer. The throughput graphs are as follows:

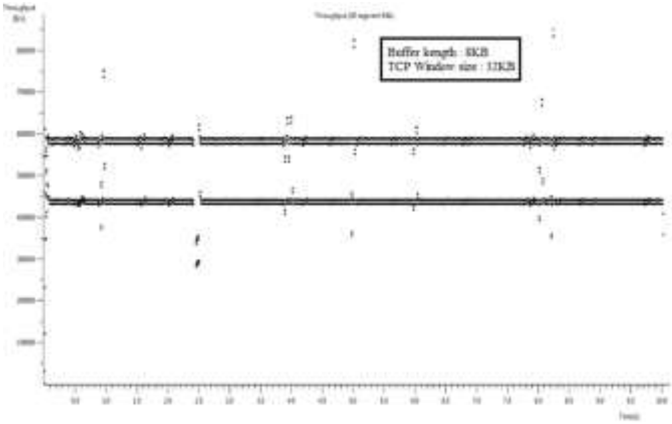

Figure 3 (Buffer Length: 8KB, TCP Window Size: 32KB

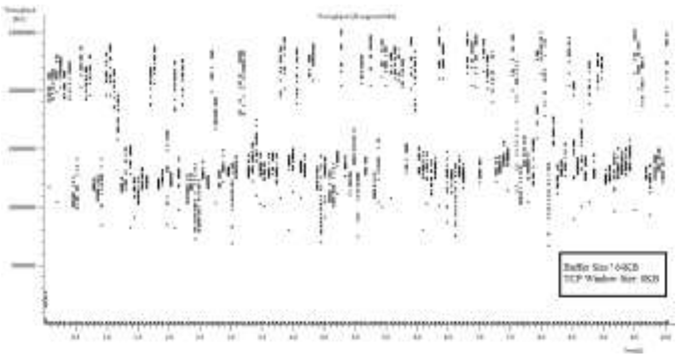

Figure 4 (Buffer Length: 64KB, TCP Window Size: 8KB)

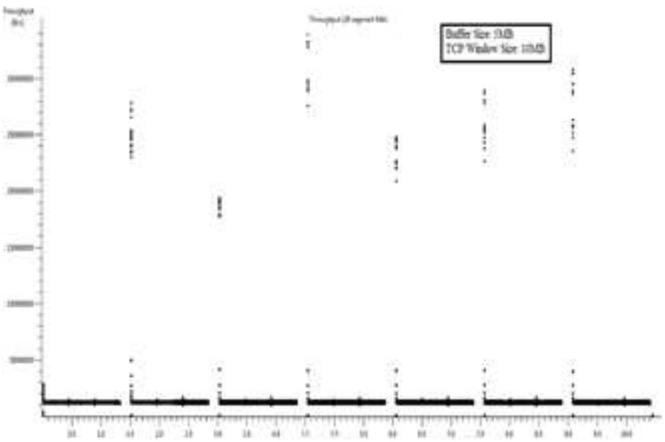

Figure 5 (Buffer Length: 5MB, TCP Window Size: 10MB)

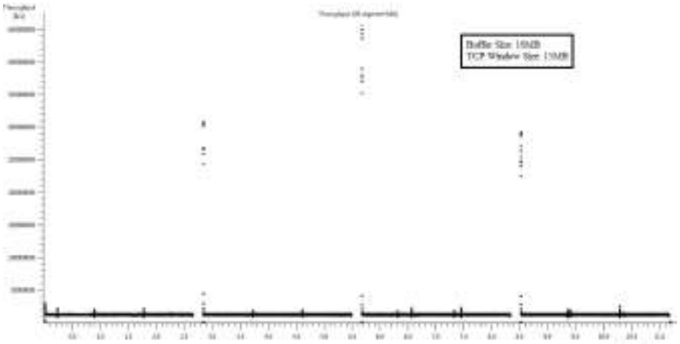

Figure 6 (Buffer Length: 10MB, TCP Window Size: 15MB)

V. CONCLUSION

We analyze the performance of a single, long-lived, Compound TCP (CTCP) connection in the presence of random packet losses. We notice that CTCP gives always a throughput equal or greater than Reno, while relative performance in terms of jitter depends on the specific network scenario. CTCP can efficiently use the network resource and achieve high link utilization. Due to the delay-based component employed in the CTCP congestion avoidance algorithm, CTCP has improved RTT fairness compared to regular TCP.CTCP generally improves the throughput by 28% to 52% compared to regular TCP.

ACKNOWLEDGMENT

We would like to express our sincere gratitude and gratefulness to our professor Mr. Kapil S. Raviya for providing us with all sorts of basic ideas and techniques essential for carrying out this project work from the very beginning to the end and enabled us to present this dissertation in this form. The teaching staffs also deserve our sincere thanks for sharing their discussion and exchange of ideas.

Lastly, we would like to extend our sincere thanks to the known and unknown writers of the books and references that has been taken during the preparation of this project work.

REFERENCES

[1] Hardik V. Miyani , Vishv B. Kukadiya, Mr. Kapil S. Raviya, Mr.Dhrumil Sheth , "Performance Based Comparision Of TCP Variants „TAHOE, RENO, NEWRENO, SACK" In NS2 Using LINUX PLATFORM"J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Yuan-Chen Lai & Chan -Li Yao , "The Performance Comparison between TCP Reno and TCP Vegas"K. Elissa, "Title of paper if known," unpublished.

[3] I. Abdeljaouad, H. Rachidi, S. Fernandes, A. Karmouch , "Performance Analysis of Modern TCP Variants: A Comparison of Cubic, Compound and New Reno.

[4] R. D. Mehta, Dr. C. H. Vithalani, Dr. N. N. Jani , "Evaluation Of TCP Variants – „RENO" and „SACK" On „REAL TEST BED" Using EMULATOR – „NISTNET""

[5] Habibullah Jamal, Kiran Sultan , "Performance Analysis of TCP Congestion Control Algorithms"

[6] Madhvi A. Bera, Bhumika S. Zalavadia, Rashmi Agrawal , "Effectuation Of TCP Agents And Equivalence Of Outcome With Different TCP Variants"

[7] Mr Devang G. Chavda, Prof. Ridhdhi I. Satoniya , "Performance Evaluation of TCP in the Presence of UDP in Heterogeneous Networks by using Network Simulator 2"

[8] Usha Banerjee, Ashutosh Vashishtha, Mukul Saxena, "Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection", International Journal of Computer Applications (0975 – 8887) Volume 6– No.7, September 2010

[9] Wolf-Bastian P¨ottner, and Lars Wolf, "IEEE 802.15.4 packet analysis with Wireshark and off-the-shelf hardware"

[10 Russell Bradford, Rob Simmonds and Brian Unger, "Packet Reading for Network Emulation"

[11] Alberto Blanc, Konstantin Avrachenkov, "Compound TCP with Random Losses"

[12] Kun Tan Jingmin Song, Qian Zhang and Murari Sridharan, "Compound TCP: A Scalable and TCP-Friendly Congestion Control for High-speed Networks"

URL

[13] http://www.isi.edu/nsnam/ns/

[14] http://www.isi.edu/nsnam/ns/ns-emulation.html

[15] https://www.wireshark.org/

[16] http://openmaniak.com/wireshark.php

[17] http://openmaniak.com/iperf.php

[18] https://iperf.fr/