Theoretical Approach and Case Study For Optimization in Elementary TCP

[1]Akshay Bhavsar, [2]Lalsingh Chouhan
[1]ME Computer Science engineering, Department of computer science,
Shiv Kumar Singh Institute of Technology and Science, Indore
[2]Assistant Professor Department of computer science, SKITS engineering collage Indore

---

Abstract - The World Wide Web now revolutionized our life and the way of communication take place which lead to making it impacting our daily life. Currently it is wide information repository of news, media, and facts and figures. This advancement of network goes through various phases like telephony, wired and wireless networks, UMTS and WiMAX and currently LI-FI etc. Now the common problem is that we have immerging numbers of users and vast population which uses internet and still we have very less operating network speed. We must optimized computer networks to get maximum benefits with minimum operating cost. Most of the wired networks are designed very well to perform in many different network conditions. However, TCP applications in dynamic and wireless networks experience severe rendering degradation because packet losses causes by bit errors and handoffs initiate congestion control mechanism. This situation leads to an absolute necessity to design and optimize the TCP congestion control mechanism to effectively administration of the non-congestion related issues in dynamic infrastructures. This paper presents some other approaches to the design better optimize protocols and enhancement proxies for TCP congestion control mechanism. In this paper we discuss enhancement in TCP performance by using multiple virtual path connection to the destination. With this motivation we build multiple subflows which are used to transfer packets to the destination.

I. INTRODUCTION

In recent year devices like android phones, PDA, laptops have become more and more common. These all devices have adding benefits that they are multihomed devices, they have 3G and Wi-Fi interfaces inbuilt among themselves which gives you freedom to provide more access technologies to user like 3G, 4G, LI-FI etc. Hence, All the techniques which are proposed till now has emphasis on congestion control technique as it is major drawback at transport layer. The congestion control techniques which we are using now perform well in small networks but as network is growing fast, it cannot deal with faster data transfer. However multipath data transfer network provide solution for this type of problems.

TCP is most used protocol for transporting data and used by many of the applications. It was designed in the late 70s, and at that time devices were not equipped with multiple interfaces, so researchers didn't need to pay attention to this matter but beside this the TCP designers knew that network could be shutdown some time or path can be unreachable, so they choose to follow layered architecture and decouple transport layer from network layer so that the network could reroute packets around failures without making impact on transport layer connections.

Rerouting of data packets made possible by having dynamic routing protocols which uses dynamically path finding if one path goes down, and this work become easier because of layered architecture so this make less burden on designer so they can make more flexible protocols

In this era we are now equip with devices which can provide connectivity with more then one interface, like datacenters have many redundant paths between servers and data center station, and multihoming has become the infector for large servers and datacenters. Meanwhile, TCP is still working on its past technology at single path transfer, at the time of building connection between source and destination TCP bind to IP address of source and destination. If even a single address has been changed, for whatever reason, the whole network will fail down. Even TCP can load balance between more than one network because it need to reorder packets at destination side, and while waiting for packet to reorder TCP can be misinterpret this as congestion and can discard packet at receiver side.

Multipath protocols provide a solution for a more efficient application of the network resources allocated to and administrated by different stakeholders in the Internet. The multipath protocols are capable of transmitting the traffic of individual end users through several paths and switching – potentially seamlessly – from one path to another, which is expected to lead to an improved end-user experience of online services

II. RELATED WORKS

Esra C. Paasch et al. [1] focus on different schedulers for Multipath TCP. they first design and implement a generic modular scheduler framework that enables testing of different schedulers for Multipath TCP then use this framework to do an in-depth analysis of different schedulers by running emulated and real-world experiments on a testbed. they consider bulk data transfer as well as application limited traffic and identify metrics to quantify the scheduler's performance.

B. Hesmans and O. Bonaventure [2] gives a new extension to TCP that enables a host to transmit the packets from a given connection by using several interfaces. they propose mptcp-trace, a software that enables a detailed analysis of Multipath TCP packet traces.

Nigel Williams [3] update and expand MPTCP implementation and congestion control
Tuan Anh Le [4]proposed the scheme of energy aware transmission protocols by propose energy-aware congestion control algorithm for multipath TCP (ecMTCP), in which the rate control is based on a traffic sharing policy amongst the paths, and which is driven by their energy costs and traffic loads.
Sinh Chung Nguyen [5] tested the behaviors of MPTCP without coupled congestion control option in heterogeneous networks and found that heterogeneous environment degraded MPTCP performance because of out-of-order phenomenon
Wischik [6] in his research provide end to end mechanisms for sharing capacity, precisely to alterations to TCP's congestion control algorithm
Costin Raiciu [7] observed how the use of MPTCP could progress data center performance by accomplishment very short schedule distributed load balancing.

III. PROPOSED METHODOLOGY

Already By enabling multipath support, a multihomed device can perform load-balancing between congested paths; it can be recover lost bandwidth by use shifting traffic on secondary paths and use bandwidth of secondary paths. Still, current multi path methods accomplish only abrasive grained load balancing due to a rough evaluation of network congestion using packet losses.
In this research we will formulate a protocol to help multihomed device to load balance more than one path and can be resolve congestion control. In this research we will equalize flow in each subflow and resolve bottleneck problem with is a common problem in low bandwidth networks. Our research work and outcomes will show result enhancement in term of throughput and packet delivery ratio. Our protocol provide fairness feature at TCP congestion window progression. After validating result our research will provide enhanced version of TCP while not destroying its entire fairness feature. Outcomes of our result will be compare with various version of TCP, and to ensure that our propose protocol is based on the delivery delay of the data segments can accomplish minimal end to end delay. Our protocol will also less sensitive to buffer allocation size

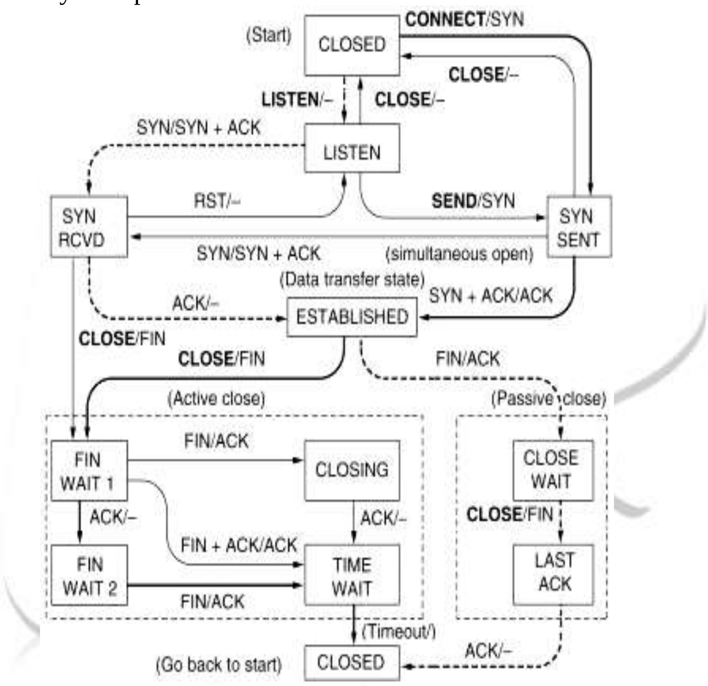

Figure 1: MPTCP State machine

A Considering the creation and closure of the connection and the transactions between states, MPTCP's behaviour is identical to TCP, they both share the same model for all the connection-oriented operations such as: opening and closing a connection, acknowledge the other end-point about sent data, etc.
Thus, MPTCP uses a three-way handshake to establish a connection, as follows:

- The server performs a passive open to be ready to accept incoming connections. This operation is usually done by creating a new socket, binding it to a network address and performing a "listen" instruction.
- The client, using a connect function, performs an active open by sending a SYN packet which includes an initial pseudo-random sequence number.
- When the server receives the client's SYN segment, it answers with an acknowledgment and its own SYN.

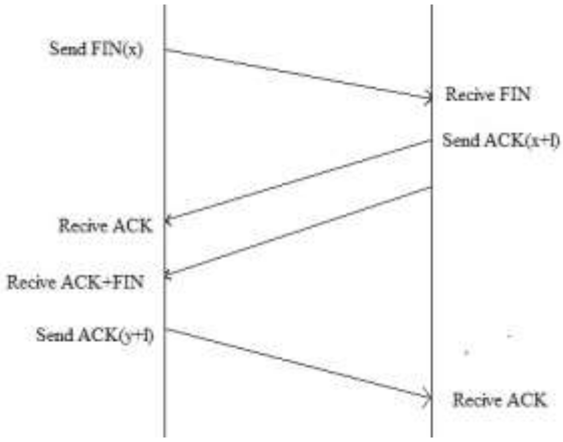

Figure 2: TCP Handshaking

While the exchange of three packets is needed to open a connection, the closing procedure requires four, as follows:

- One of the two endpoints calls the CLOSE interface (see paragraph 3.3), starting a process called active close. This involves sending a FIN segment that indicates the end of the data transmission on the current connection.
- When the other endpoint receives the FIN, it executes a passive close and acknowledges the FIN packet with an ACK. This operation is also reported to the process that opened the socket as an "end of file" after any possible packet in the sending queue has been sent.
- When the process receives this EOF message, it will call the close function to its own socket causing the transmission of another FIN segment.
- The other endpoint will receive the FIN packet and will answer with an ACK.

IV. CONCLUSION

Our new Multipath TCP extension protocol will improve the TCP performance while not destroying the whole performance by creation TCP more destructive on its superlative currents. Different congestion control alternatives for TCP (Transmission control protocol) will calculate and compare. Our proposed algorithm provides a higher priority to the equality feature at the outflow of congestion window development. These indications to a throughput consequence as exposed for the distinct development and it can straight main to a throughput lower than that of other TCP agents.

Our propose algorithm provides a higher priority to the fairness feature at the expense of congestion window progression. Performance comparison outcomes will obtain for the accessible As predictable, To present that our proposed mechanism that is based on the delivery delay of the data segments can accomplish minimal re-allocation delay and hence will also fewer sensitive to the receiver buffer size.

V. REFERENCES

[1] B. Hesmans and O. Bonaventure," Tracing Multipath TCP Connections " ,ACM SIGCOMM. 2014.

[2] C. Paasch, S. Ferlin, O. Alay and O. Bonaventure, 'Experimental Evaluation of Multipath TCP Schedulers',ACM SIGCOMM Capacity Sharing Workshop (CSWS'14). 2014.

[3] G. Detal, "Evaluating and Improving the Deployability of Multipath TCP ",ACM SIGCOMM Capacity Sharing Workshop (CSWS'14). 2014..

[4] Nigel Williams, Lawrence Stewart, Grenville Armitage," Design Overview of Multipath TCP version 0.3 for FreeBSD-10" CAIA Technical Report 130424A April 2013.

[5] Tuan Anh Le, Choong Seon Hong, Member, IEEE, Md. Abdur Razzaque, Member, IEEE, Sungwon Lee, Member, IEEE, and Heeyoung Jung, Member, IEEE," ecMTCP: An Energy-Aware Congestion Control Algorithm for Multipath TCP" IEEE COMMUNICATIONS LETTERS, VOL. 16, NO. 2, FEBRUARY 2012.

[6] Sinh Chung Nguyen and Thi Mai Trang Nguyen," Evaluation of Multipath TCP Load Sharing with Coupled Congestion Control Option in Heterogeneous Networks" 978-1-4577-1261-6/11/ -2011 IEEE

[7] Damon Wischik, Costin Raiciu, Adam Greenhalgh, Mark Handley," Design, implementation and evaluation of congestion control for multipath TCP" Proceeding

[8] Costin Raiciu, Sebastien Barre, Christopher Pluntke," Improving Datacenter Performance and Robustness with Multipath TCP" SIGCOMM '11 Proceedings of the ACM SIGCOMM 2011.