Implementation and Optimization In Rudimentary TCP Using Multiple TCP Sub-flow Technique

Mr. LalSingh Chouhan, Mr. Kamal Borana
M.E Computer Science Engineering, Assistant Professor
Department of Computer Science
S.D. Bansal College of Engineering Indore MP

Abstract - **Many computers and devices such as smart phones, laptops and tablet devices are now equipped with multiple network interfaces, enabling them to use multiple paths to access content over the network. If the resources could be used concurrently, end user experience can be greatly improved. The recent studies in MPTCP suggest that improved reliability, load balancing and mobility are feasible. Multipath approach is benefited from having multiple connections between end hosts. However, it is desired to keep the connection set minimal as increasing number of paths may not always provide significant increase in the performance. Moreover, higher number of paths unnecessarily increase computational requirement. Ideally, we should suppress paths with low throughputs and avoid paths with shared bottle-necks. In case of MPTCP, there is no efficient way to detect a common bottleneck between subflows. MPTCP applies a constraint of best single-path TCP throughput, to ensure fair share at a common bottleneck link. The best path throughput constraint along with traffic shift, from more congested to less congested paths, provide better opportunity for the competing flows to achieve higher throughput. However, the disadvantage is that even if there are no shared links, the same constraint would decrease the overall achievable throughput of a multipath subflow. We are presents in this research a new multipath delay based algorithm, EIA (Estimated Iterative Algorithm), which helps to providing higher throughput and efficient load balancing.**

I. INTRODUCTION

In 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 dissertation presents some other approaches to the design better optimize protocols and enhancement proxies for TCP congestion control mechanism

In this paper we enhance 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.

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 situation.

The traffic volumes transmitted over the Internet are growing, raising concerns about the sufficiency of network capacity. If the demand for Internet services continues to grow as expected, two approaches can be taken to secure sufficient capacity supply: 1) building new capacity within network domains, or 2) benefiting from more efficient use of the existing capacity. The stakeholders in the Internet, such as Internet service providers (ISPs) and content providers (CPs), are adopting both strategies within their network domains.

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.

These features have been specified in numerous Internet protocols, e.g., SCTP and Multipath TCP (MPTCP). IETF has been working in this field and devolving these type of protocols which can be proved efficient in today's scenario. Despite the many efforts has been put in this field of having multipath protocols at transport layer but still it is in its initial age as we are not able to implement these protocol in working environment. Technical comparisons of multipath and their competitor protocols have been made, but the deployment and economic feasibility of multipath protocols remain unstudied.

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 controlTuan 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

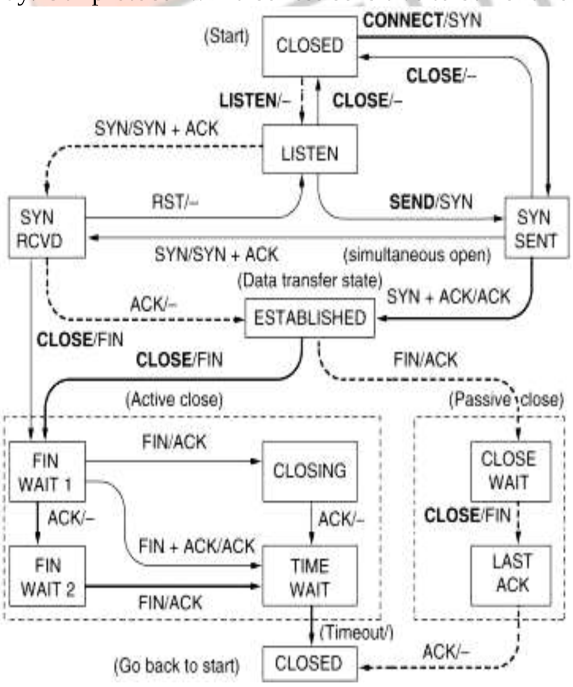


Figure 1: MPTCP State machine

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.

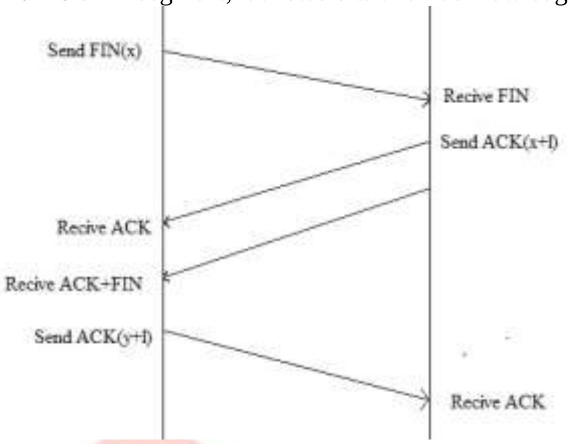


Figure 2: TCP Handshaking

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. SIMULATION MODEL

For the purpose of conducting our experiment, we implemented the our multipath transport protocol and compare it buffer based TCP protocol, round robin TCP and TCP Reno protocol. In our simulations, we use different numbers of nodes, where one node acts as sender and other act as receiver for data file. We use FTP protocol for data load. The implemented code simulates the common characteristics and limitations of real networks like dynamically changing latency of links and drop behaviour when the load exceeds the link capacity. We use dynamically changing behaviour for delay making in TCP. And make TCP socket enable to use distinct path to the destination as our protocol use distinct paths.
For the sake of simplicity, we just consider two available paths between the client and the server (i.e. path 1 and 2). We use different-different packet sizes like 512 bytes and 1024 byte and then make study of congestion control.

Simulation Environment

As test bed simulation is the most important part of the thesis, we use network simulator 2 as our experiment conducting tool. We make different wireless and wired scenario for conducting our experiment. A snapshot of our wireless topology is shown in fig.

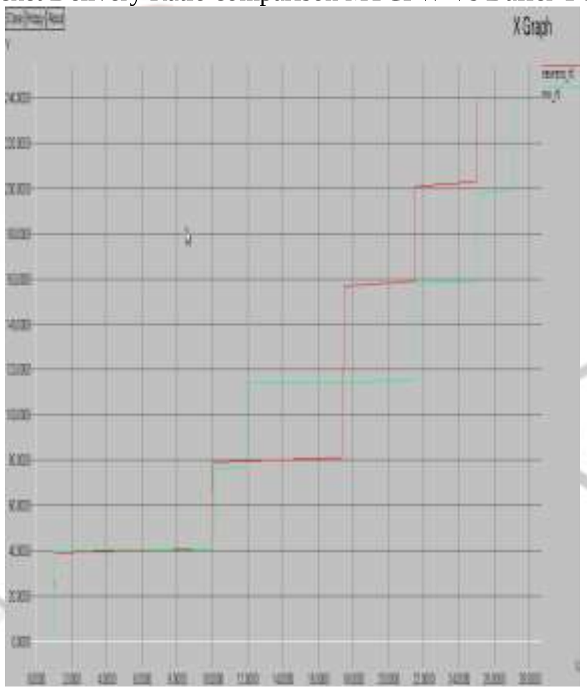
Node Configurations

Outgoing packet size	512 bytes (based on 1500-byte MTU)
Outgoing rate	As many packets as possible per second
Incoming packet size	0 bytes (off)
Incoming rate	0 (off)
Runtime	120 seconds
Receive buffer	100
Send buffer	100

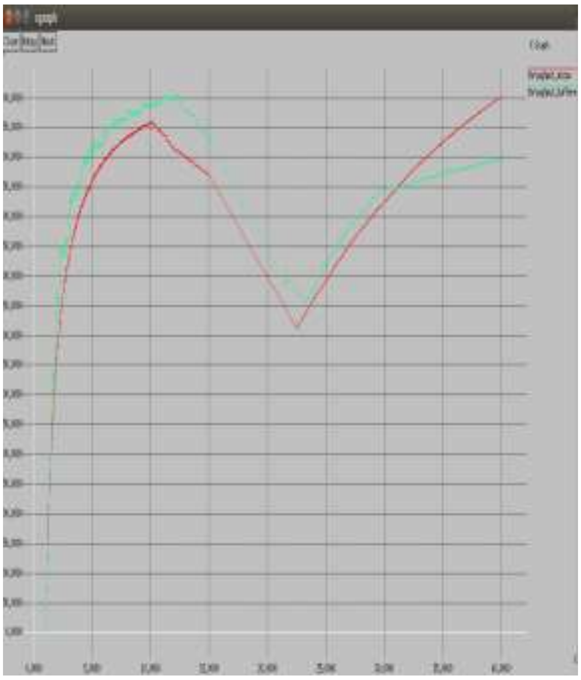
Various experiments are perform to evaluate proposed TCP which was observe for different performance parameter as packet delivery ration, throughput and performance overhead.

Experimental results are shown below:

Packet Delivery Ratio comparison MTCPW Vs Buffer TCP



Routing load for MTCPW Vs Buffer TCP


Throughput for MTCPW Vs Buffer TCP

V. CONCLUSION

Our In this part we first present the simulation model used to evaluate the performance of the proposed methods and then provide the experimental results and outputs.

VI. 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.