MULTIPATH TCP CONGESTION CONTROL ALGORITHMS ANALYSIS

Sachin P C*, Sailesh Ravipathi†
Computer Science & Information Systems, Birla Insitute of Technology & Science, Pilani, IND.
Email: *h20180140@pilani.bits-pilani.ac.in, †h20180145@pilani.bits-pilani.ac.in

***Abstract*—Multipath TCP is an advanced implementation of TCP, using which multiple interfaces can be used to send data across the network.Multipath Implementation can be used for multi homed systems and also in data centers.Many congestion control algorithms have been proposed to implement MPTCP such that it will acheive all the goals. In this paper, we will analyse two main congestion control algorithms Linked Increase Algorithms and Oppurtunistic Likned Increase Algorithm and how they try to acheive the goals of MPTCP.**

I. INTRODUCTION:

TCP was designed when the internet strictly obeyed the end-to-end principles and each host had a single IP address. TCP connection is made between two IP address and these addresses must remain fixed over the whole connection life time. A change of connection breaks the connection and the data transfer is stopped. Hence, the data to be sent between the client and the server follows one single path. When there is high traffic in the path, it leads to congestion. Nowadays, each device of host have multiple IP addresses/Interfaces through which data can be sent, i.e. the endpoints are often connected by multiple paths but the communication occurs only through one single path. This led to Multipath TCP which uses multipath subflows to send the data from source to client. This helps in better congestion control as the data is sent more efficiently using the bandwidth of the multiple subflows and also data is sent depending on the network traffic in the subflows. The main aim of the Multipath TCP is to achieve performance advantage over TCP by moving traffic from more congested path onto the ones which are less congested.

II. DESCRIPTION AND CRITICAL ANALYSIS OF MPTCP:

Multipath TCP is an extension to regular TCP. Multipath TCP allows one TCP connection to spread across multiple paths. It helps in sending of data using multiple subflows which uses different paths. The goals of Multipath TCP are that:

1) Goal 1 (Improve Throughput) A multipath flow should perform at least as well as a single path flow would on the best of the paths available to it.
2) Goal 2 (Do no harm) A multipath flow should not take up more capacity from any of the resources shared by its different paths than if it were a single flow using only one of these paths. This guarantees it will not unduly harm other flows.
3) Goal 3 (Balance congestion) A multipath flow should move as much traffic as possible off its most congested paths, subject to meeting the first two goals.

If each multipath flow sends more data through its least congested path, the traffic will move away from the congested areas. This improves robustness and overall throughput. New congestion control algorithms are needed for Multipath TCP as the current algorithms used for congestion control algorithms in TCP have series of issues in Multipath context. One of the prominent problems is that running existing algorithms such as standard TCP algorithms independently on each path would give the multipath flow more than its fair share at bottleneck link traversed by more than one of its subflows. There are several congestion control algorithms used to control the data sent in each subflow depending on the traffic in each of them. The main aim of the congestion control algorithms is to move traffic away from the more congested path so that the loss rates across the whole network will tend to be balanced. Multipath TCP congestion control algorithm, Linked Increase algorithm, Opportunistic Alias Linked Increase algorithm,, Balanced Linked Adaptation algorithm are some of the algorithms used for congestion control in Multipath TCP.

Linked Increase algorithm (LIA) couples the congestion control algorithms running on different subflows by linking their increase functions and dynamically controls the overall aggressiveness of the multipath flow. The result is a practical algorithm that is fair to TCP at bottlenecks while moving traffic away from congested links. This algorithm relies on traditional TCP mechanisms to detect drops, to retransmit data etc.

The additive increase and subtractive decrease behaviors can be described as follows: For each ACK received on sub-flow i, the congestion window cwndi is increased by:

$$min\left\{\frac{\alpha.B_ack.Mss_i}{\Sigma_{i=0}^{n}cwnd_i}, \frac{B_ack.Mss_i}{cwnd_i}\right\}$$

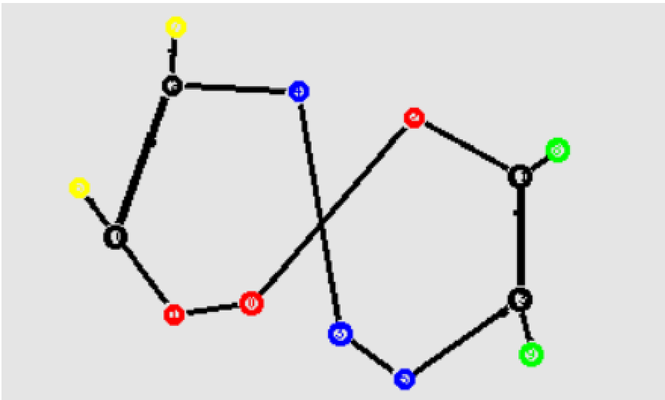

Fig. 1. This diagram shows the Network used for simulation

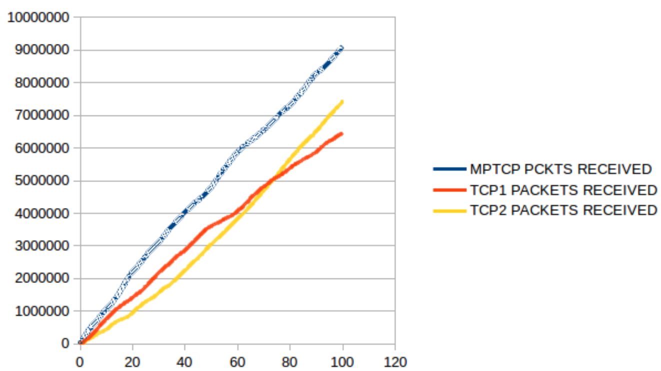

Fig. 2. This diagram shows the comparison between total number of packets sent by MPTCP using OLIA congestion control algorithm and other two TCP flows

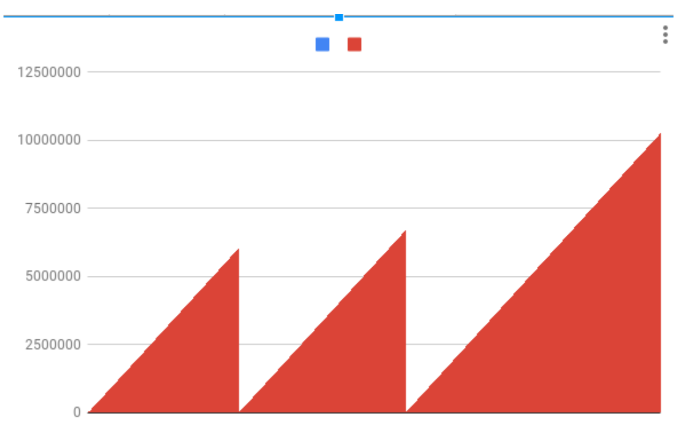

Fig. 3. This diagram shows the comparison between total number of packets sent by MPTCP using LIA congestion control algorithm and other two TCP flows

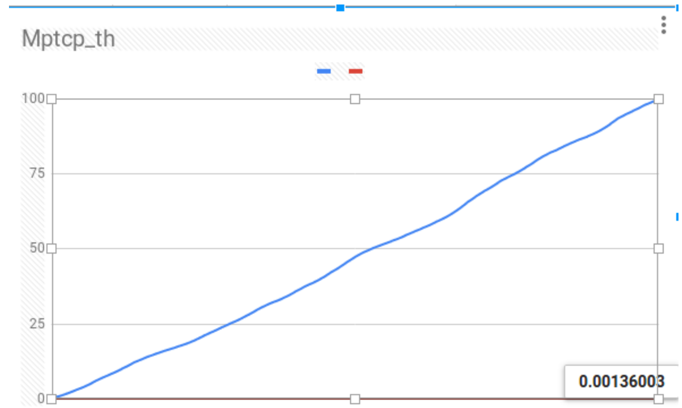

Fig. 4. This diagram shows the thrpughput given by the OLIA congestion control algorithm

α: A parameter that describes the aggressiveness of the multipath flow.
B_{ack} : Number of acknowledged bytes.
Mss_i: Maximum segment size on sub-flow i.
n:Total number of sub-flows.

- For each loss on sub-flow i, decrease cwndi by cwndi/2.

Taking the minimum of these values ensures that any multipath sub-flow will not take capacity more than single path TCP, and hence ensures goal 2. Alpha is a parameter used to adjust the aggressiveness of the multipath flow, its value is chosen such that the total throughput of the multipath flow is equal to the throughput which TCP would get on the best path and this meets the first design goal. In order to meet the first design goal, it is impossible to choose a single value of alpha that achieves the desired throughput at each time. Hence, alpha is computed based on the observed behaviors of all paths as shown below.

$$\Sigma_{i=0}^{n} cwnd_i \frac{Max\left\{\frac{cwnd_i}{RTT_i^2}\right\}}{\left\{\Sigma_{i=0}^{n}\frac{cwnd_i}{RTT_i^2}\right\}^2}$$

Opportunistic Alias Linked Increase algorithm (OLIA) was introduced as an alternative to Linked Increase algorithm (LIA). It couples the increase of congestion windows and uses the same behavior in regular TCP in case of loss. This algorithm is only applied to the increase part of the congestion avoidance phase, slow start algorithm is the same as the one used in regular TCP with small modification in case of multiple paths are established.

The additive increase behavior can be described as follows: For each ACK received on path i, increase congestion window cwndi by:

$$\frac{\frac{cwnd_i}{RTT^2}}{\left(\left(\Sigma_{i=0}^{n} cwnd_i\right).\left(\frac{cwnd_p}{RTT_p}\right)\right)^2} + \frac{\alpha_i}{cwnd_i}$$

where:
cwndp: Window size of a path p with largest congestion window.
RTTp: Round trip time of a path p with largest congestion window.
α_i: Adjust parameter for a path i.
n: is the total number of sub-flows.

- For each loss on sub-flow i, decrease cwndi by cwndi/2.

The 1st formula provides the optimal resource pooling, compensates for different RTT values. The second term with alpha guarantees the responsiveness and non-flappiness of the algorithm.
Linked Increase algorithm implementation forces a tradeoff between optimal resource pooling and the representativeness, both goals cant be achieved at the same time, this leads to the fairness problem to TCP users. Linked Increase algorithm and Opportunistic Alias Linked Increase algorithm suffer from either unfairness to the single path TCP or unresponsiveness to network changes under certain conditions especially when all paths used for MPTCP have the same round trip time Linked Increase algorithm, Opportunistic Alias Linked Increase algorithm suffer from either unfairness to the single path TCP or unresponsiveness to network when all the paths used for MPTCP have the same round trip time. In current congestion control algorithms, some algorithms tend to allocate all traffic to a single subflow and hence doesnt use the subflows efficiently even though not much traffic is there on other subflows. This leads to less throughput compared to using best single path TCP (which is one of the main goals of Multipath TCP).

III. Analysis

In this paper, we have analysed the two algorithms Linked Increase Congestion Control algorithm and Oppurtunistic Linked Increase COngestion control algorithm using NS-2 Simulations. From our analysis for a given network, Linked Increase Congestion control algorithm gives better throughput compared to Oppurtunistic Linked Increase Algorithm.

We can observe that the number of number of packets sent by MPTCP using LIA is more compared to the number of packets sent by MPTCP using OLIA for the same network. Hence, for our simulation, LIA performs better than LIA. The throughput given by the OLIA algorithm is 1.36Kbps.

IV. Conclusion or Future Work:

From the simulations we did, we got better results for LIA implementation than OLIA implementation. However, the simulation was done for a small network and the performance may vary as the network increases. We can also perform simulations for packets with short flow and long flows and can compare the analysis between these two algorithms also analyzing the fairness of each subflow w.r.t utilization of bandwidth in each subflow. Also, we have come up with a theoritical algorithm for congestion control in MPTCP which can be implemented and compared with other algorithms. This algorithm is only applied to the increase part of the congestion avoidance phase, slow start algorithm is the same as the one used in regular TCP with small modification in case of multiple paths are established.

q=$\alpha * \rho + \beta * RTT_i$
p=1-q

- For each ACK received on sub-flow i, the congestion window $cwnd_i$ is increased by: p/cwnd
  i.e. $cwnd_i$ = $cwnd_i$ +(p/cwnd)

alpha and Beta are taken such that 0¡= q ¡=1.

Also, throughput of all flows sharing a particular bottleneck link should be less than the throughput provided by the regular tcp. If it exceeds the regular tcp throughput, then the new cwndi calculated is adjusted such that the throughput shared by the flows in a bottleneck link is less than the throughput given by regular tcp.

References

[1] Design, implementation and evaluation of congestion control for multipath TCP by Damon Wischik, Costin Raiciu, Adam Greenhalgh, Mark Handley, University College London.
[2] Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP by Ramin Khalili,T-Labs/TUBerlin, Nicolas Gast Miroslav Popovic Jean-Yves Le Boudec EPFL-LCA2 October 21, 2013
[3] Coupled Congestion Control for Multipath Transport Protocols: C. Raiciu Univ. Politehnica of Bucharest M. Handly ,D. Wischik Univ. College London October 2011
[4] How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP by Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure and Mark Handley.
[5] Performance analysis of Multi-path TCP Network by Hamzah M A Hijawi and Mohammad M. N. Hamarsheh.
[6] Decoupled from IP, TCP is at last able to support multihomed hosts by Christoph Paasch and Olivier Bonaventure, UCL.