Multipath TCP: A Comparative Analysis for Linux kernel implementation

UCSDCSE
Computer Science and Engineering

Agneev Ghosh, Jayant Malani and Heli Utpal Modi

Goal: Analysis & Justify usability

Aim: Using existing MPTCP principles for Linux kernel implementation, develop a framework incorporating a comparative study of MPTCP protocol Design Goals and its performance w.r.t. TCP and UDP protocols.

Experimental Setup

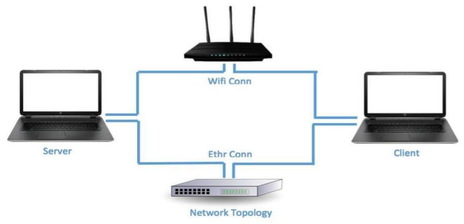


Figure A

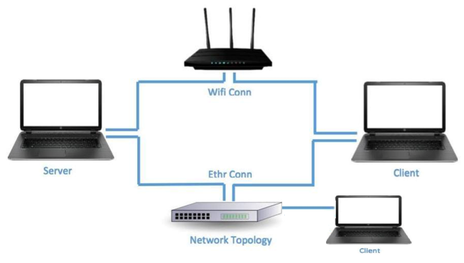
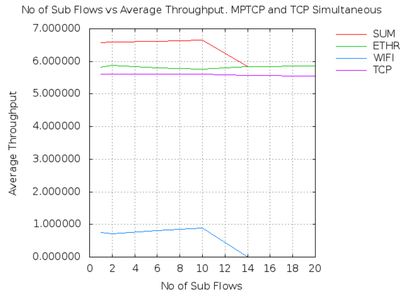


Figure B

- *Figure A* depicts the primary experimental setup .
- It comprises of a MPTCP enabled client and server having multiple communication paths viz. Ethernet (5-port 12.5MBps Unmanaged Gigabit Switch) and Wi-Fi (1MBps).
- *Figure B* lays out the secondary architectural setup which has an additional (Non MPTCP) client (connected via Ethernet) introduced to compete with the MPTCP client , to emulate pragmatic scenarios.

MPTCP Design Goals Validation

#1 Fair share with TCP and
#2 Perform at least as well as TCP



Graph 1

As observed in the above graph,

•MPTCP follows the fair share property while increasing no of sub flows

•It performs as well as TCP when there is competing traffic

#3 MPTCP should use efficient paths

In the experimental setup, MPTCP always preferred Ethernet in spite of increasing RTT delay or packet loss on it.

#4 Efficient Resource Utilization

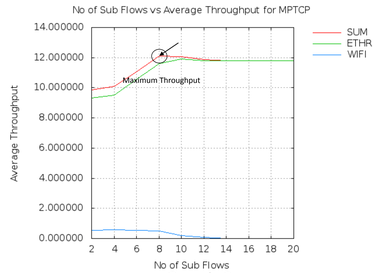
Graph 2

Figure 3

MPTCP utilized full capacity of the resources (i.e. throughput) on available paths as compared to TCP connection.

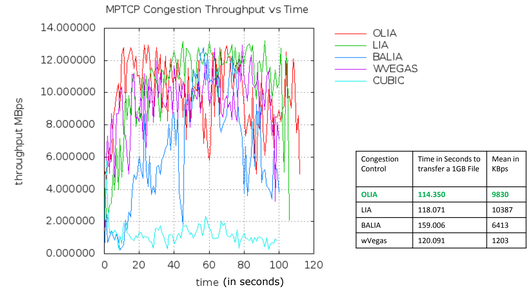
Further Simulation & Evaluations

Increasing Sub-Flow Analysis.



Graph 3

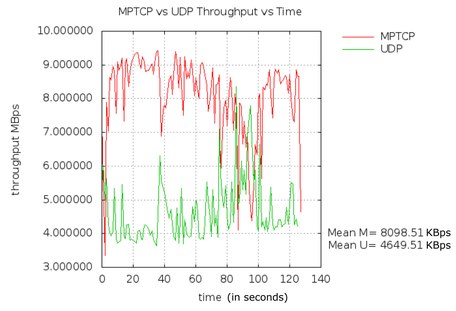
MPTCP Congestion Control Evaluation



Congestion Control	Time in Seconds to transfer a 1GB File	Mean in KBps
OLIA	114.350	9830
LIA	118.071	10387
BALIA	159.006	6413
wVegas	120.091	1203

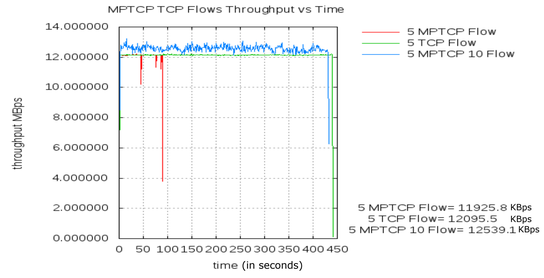
Graph 4

Behavior with UDP Congestion



Graph 5

*N sub-flows MPTCP v/s *N number of TCP on Single Path



Graph 6

Conclusive Remarks-Key Findings and Future Work

The Key Findings of the experiment are as follows:

1. As collectively observed in Graphs 3 & 5, we found that the optimal number of sub-flows count(*N) is 10, for which MPTCP outperforms TCP and UDP, always maintaining the property of fair share in both the cases.
2. Graph 4 provides a comparative analysis among the MPTCP congestion control protocols, and it can be concluded that OLIA performs the best by consuming least time for transmission.
3. Graph 6 provides a comparative analysis of 5 independent MPTCP/TCP connections (A), 5 MPTCP connections having *N number of sub-flows (B) and 1 MPTCP connection having 5 sub-flows (C). As observed, A outperforms B which outperforms C, thus justifying the need to use MPTCP connections over TCP connections.

•From the extensive analysis of various parameters such as RTT delay, packet drop rate, # of sub-flows, competing TCP and UDP connections and use of different congestion control protocols- we can conclude that MPTCP functions as good as TCP and UDP along with withholding its basic design properties i.e. full utilization of resources

•For future work, evaluating the scalability w.r.t. deploying it over the internet and implementing better window management schemes are some avenues of interest.