Design and Development of Efficient Multipath TCP for Data Center in Public Cloud

Shashikala S V, Ravikumar G K

***Abstract**: In recent days, variety of application development making use of cloud data centers effectively for analyzing data, processing related applications such as interactive and batch processing. Some of the factors influence on network performance by maintaining low delay, high-speed data transfer and more reliability. Maintaining and satisfying these factors becomes a challenging issue due to dynamic variations on network. TCP (Transmission control Protocol) proposed recently a new concept of multi-path TCP (MPTCP) for maximum utilization of paths over network flow. In spite of its added advantages, some sort of work on MPTCP to be carried out on cloud environment and further, efficient way of using MPTCP on real-world cloud application still looks like unclear problem. Our proposed work concerned on MPTCP usage in most effective and feasible way for cloud and data center environments over various conditions on network. The experiment is conducted by clustering the public cloud data using k-means algorithm and communicated over a network using MPTCP. The results shows that the proposed method yields high-speed data transfer and low communication delay when compare to traditional TCP technique.*

***Keywords** : Data centres , MPTCP, TCP*

I. INTRODUCTION

Cloud data centers deploys a complex variety of tasks having enhanced and varying conditions. The nature of almost all task are distributed and communication happens over the network established in data centres. At the same time, data centres network provides support for various requirements. The load balancing is done properly from small client-server application (E.g.: Web application) to data processing applications on distributed approach (E.g.: Live migration on virtual machine).

The various combinations of utilization's, task burstiness, and dynamic arrangement of usage segments on data center servers brings increases its dynamicity and burst system [5] traffic. This high changeability in the traffic of a system can leads to blockage on specific region of the cloud data server network which can leads to inequality to organize. This irregularity and blockage in turn gives problems both system performance and delay, and can seriously troubles tasks execution. In a few latency-oriented applications, like, internet-search, influences rightness [3]. Given the adverse environment of network inequality, system load assessment has inward major notice by both business and academic world [2, 3]. Conventionally, one of two approaches have used to speak on dynamic load balancing inside data centres over networks. The primary move towards it uses network structures say for example fat-trees [1, 9] to give several paths among servers as well as arrangement switches, in that way provide collective network bandwidth. Later method finds out overloaded system essentials and makes appropriate switching on packets and forward decision to ease hot-spots on network. These advances push network management behind the switches over networks, and put into practice by switch-based methods like [11] Software Defined Networks (SDNs)

A less investigated option for taking care of system elements is the utilization of multiple-path networking where information to be sent and receive from an application which traverses various paths simultaneously. Multiple-path networking administration has numerous potential advantages. As a part of expanded dependability, multi-pathing empowers arrange resources from various ways to be accumulated—conceivably giving expanded data transmission. Multi-path networking can also minimizes data blockage inside the data centres or problems in network handler [21] by exploring the repeated paths. The feasibility of multiple path networking is important in the present data centres because (i) basic cloud data centres uses network topological algorithms, for example, clos and fat-trees give numerous paths to every cloud server, and further (ii) data servers themselves have multiple network interfaces (NICs), empowering them to develop rep routes to different servers.

Multi path TCP is an IETF expansion to TCP and it is a protocol which traverses end-to-end in a communication network and designed to deliver multi path networking by reducing the complex design in Hardware network Architecture [8]. Multi path TCP makes a traditional TCP stream into numerous sub-streams that pass through multiple pathways between cloud servers employ the obtainable multiple network interfaces [17] in a servers. It acts as added substitute for TCP which can hypothetically improve data center tasks efficiency by aggregating the bandwidth over multiple routes, improved reliability and resilience against network breakdown, and enhanced performance [17] with jamming. Further, with respect to data centre systems, the performance of Multi path TCP has been assessed using simulation and easy applications for measurement of the network like iperf, but not considering real-time data center and cloud balancing [15].

Manuscript published on November 30, 2019.
***** Correspondence Author
Shashikala S V*, Dept., of CSE, BGSIT, Adichunchanagiri University, BG Nagar, Mandya, INDIA.
Dr. Ravikumar G.K, Dept., of CSE, BGSIT, Adichunchanagiri University, BG Nagar, Mandya, INDIA.

Likewise, MPTCP's consistency profit has been considered for mobile circumstances such as cell phones with multi interfaces [6], a dissimilar situation by data center system.

II. BACKGROUND SURVEY

From the past decade, data center applications are developed drastically from legacy tasks to today's Cloud side complex tasks. Today cloud workloads ranges from micro service having latency of micro seconds to batch processing tasks, distributed information processing tasks [18] and are executing for hours to days. Sequentially to the evolution of the tasks, a parallel evolution is observed in data center networks to produce high network performance.

Application dependence on performance of the network may changes significantly, since execution of the tasks may be congested by the other resources like Central processing Units and Input output systems. For e.g., distributed data processing workloads the significance of high-speed networks like Hadoop and Spark are questioned at a recent time [13, 19]. Currently, the importance of MPTCP has been concentrated on its performance in a network, and it is also significance to identify the advantages of multiple path TCP for the task which shows performance form end-to-end during network communication.

Multi-path TCP holds set of extended protocols to the traditional TCP, which empowers TCP to use various sub-flows by effectively dividing into multiple paths which is shown Fig 1. By the use of multiple NIC's, Sub-flows take advantage which are obtained on each data server to travel [17] several paths between the servers. The data centres can deploy MPTCP to produce aggregation of bandwidth, higher performance and resiliency if some of the network has congested paths.

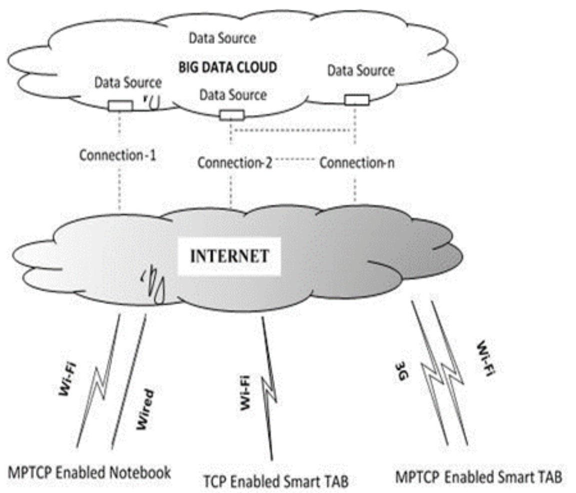


Figure 1. Use of MPTCP for big data Cloud

The design of MPTCP is robust against middle-boxes like NATs. Using the available BSD socket users can easily create MPTCP connections. MPTCP is deployed on top of TCP stack kernel as a layer. Applications needs no modifications in an MPTCP enabled kernel, since MPTCP layers considers TCP sockets as MPTCP sockets. It has more number of preferable properties, which can offer many advantages in data centres. These consists of the following:

A. Aggregating the Bandwidth via Resource Pooling

"Providing resource pool at the network layer" this one of the design principles of MPTCP. MPTCP displays multiple network connections as a single abstraction connection. Every sub-flow forward information on one network path, and the resources which are present on these paths were pooled and aggregated. For example, A two 10 Gbps interface is established between two servers (without MPTCP), can yield a throughput of up to 20 Gbps when interfaced by an MPTCP.

B. Improved Resiliency

An obvious reliability advantages can be obtained by using multiple paths. MPTCP improves network performance against failures by maintaining the connection and sends the information data using the other sub-flows or paths provided if there is failure in one path.

C. Readiness in Operations

Multiple Path TCP performs its operations at the host level, which does not wait for the support from any network resources, when compare to other networking approaches which rely on switch-level management for data centre networking [2, 46]. MPTCP provides a standard TCP socket abstraction to each task since it is acting both network and application transparent. MPTCP is accessible in operating systems (OS) such as UNIX, Linux and in iOS. Therefore, data centres and cloud users makes use of MPTCP without modifying any parameters in the network/applications.

D. Implementation of MPTCP

In Linux and other Operating systems, the implementation of MPTCP can be easily learned from the outlook of congestion control, control planes and the data.

E. Control plane

To send data on multiple paths MPTCP uses TCP connections. It maintains and enhances TCP's four-tuple model for establishing the connection. To the tasks/application, MPTCP extends the abstraction of a TCP connection. Underneath, for a single connection it sets up multiple sub-flows as shown Fig 2. Every sub flow specifies a distinct four-tuple, and using TCP the data is sent through each sub-flow. As a division of establishing a connection using MPTCP, it displays and negotiates the sub-flows depending on the presence of IP addresses on each host.

Sub flow represents a distinct path between the sender and the receiver. For eg, if every host contains 2 network interfaces (that is 2 IP addresses), four sub flows established by MPTCP. Primary sub-flow that corresponds to TCP 4-tuple connection established by MPTCP. The primary sub-flow is first established and secondary sub-flow on the other paths. Once the establishment of TCP connection, Sub flows are said to have been established and ready to send or receive data. Corresponding sub flow is removed/added by MPTCP if a path becomes (in) accessible.

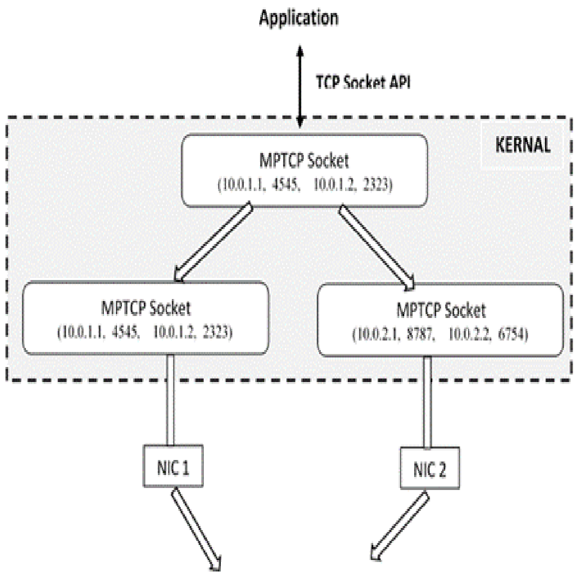


Figure 2. Multiple TCP created by MPTCP sub flows for a single TCP Connection

F. Information plane

For the underlying sub flows, MPTCP uses TCP and its information sending mechanisms depend on sequence numbers provides by TCP. The presence of multiple routes needs that the problem of scheduling packets on to sub-flow tackled by MPTCP layer, and regrouping packets [4,14] from various sub-flows at the destination. MPTCP empowers different scheduling schemes including, round robin. In standard, high efficient scheduling schemes are compared based on the available window sizes and round trip times of the sub-flows.

G. Congestion control

Multi Path TCP makes use of congestion control scheme [16] which works across all sub-flows. One of the design objective is to modify the TCP network connections to not to cause other connections to starve. Appropriately, a single congestion window used by congestion control scheme which is shared by all sub-flows and modifies the global congestion window depending congestion on each sub-flow. Along with coupled policy, the literature [10, 12 and 43] also reveals other congestion control policies which includes uncoupled polices, that considers each sub-flow alone and which leads to a problem for other TCP network connections [7].

III. PROPOSED METHOD

The experiment comes up with an assumption of different environment such as production and cluster environment in public cloud respectively. The cluster environment is created by clustering the data using K-means algorithm. The experiment is conducted in a cluster environment with public cloud consisting of various loads over network.

A. Clustering Cloud Data

In this section, we discuss K-means clustering that we use to cluster the big data which is loaded in cloud network. K-means algorithm belongs unsupervised learning. The objective of this algorithm is to segregate the unlabeled cloud data into K number of groups. Specifically, it minimizes an objective function Ψ;

$$\Psi = \sum_{i=1}^{K} \sum_{x \in X_i} |x - \bar{x}_i|^2 \quad (1)$$

Here, $\{x\}$ is a set of data X_i is a cluster and $\bar{x_i}$ is the cluster center of X_i. The brief summary of the algorithm can be given as

1. Randomly assign data into K number of clusters.
2. Calculate the cluster centre using the below equation.

$$x_i = \frac{1}{|X_i|} \sum_{x \in X_i} x \quad (2)$$

3. Regroup all data x into a new cluster based on the Euclidean distance values $|x - \bar{x_i}|$ and which is minimum for $\bar{x_i}$.
4. Repeat the steps 2 and 3 until there is no changes in cluster.

B. Public Cloud setup

Our experiment is conducted with two distinct setup environments to obtain comparative study among TCP and MPTCP. It assumes cluster environment on the public cloud systems. We experimented using AWS (Amazon Web Server) cloud infrastructure with two VMs having capacity of 4GB RAM, 2VCPU, 100GB drive space and two interfaces connected to different subnets.

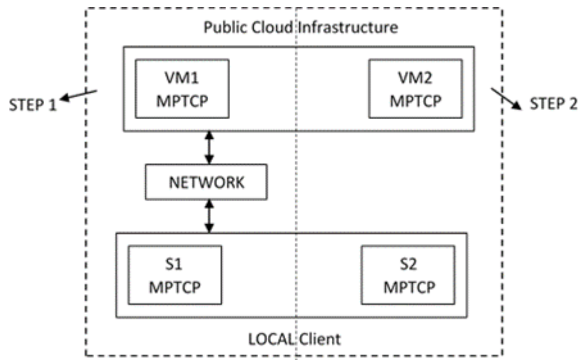


Figure 3. Shows the proposed experimental setup

Two VMs are named as VM1 and VM2 as shown in Fig 3 where VM1 was running Ubuntu 18.04, kernel version 4.10 with regular TCP setup, whereas VM2 was running with Ubuntu 18.04 kernel version 4.10 with MPTCP V0.93.

Similarly, two dedicated system are used for downloading cluster data where system S1 was equipped with 2GB RAM, i3 processor, wifi and Ethernet interfaces, Ubuntu 18.04 with normal TCP setup whereas system S2 has same specification as S1 along with MPTCP V0.93 setup.

C. MPTCP Set-Up

MPTCP configured on Linux machine version 4.10, which is new addition to linux kernel. The different configurations of MPTCP kernel and congestion control methods used to carry on our experiment. Until and unless it's not been specified means, will go with kernel default version and LIA congestion control method. The MPTCP header checksum to be turned-off to reduce unnecessary overhead on CPU.

Retrieval Number: D7803118419/2019©BEIESP
DOI:10.35940/ijrte.D7803.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The buffer size is suggested based on multiplying maximum round-trip duration for all paths by total available bandwidth. RFC6182 is used to set receive buffers on all experiments. The result on receive buffers are comparatively larger (256 KB) than default (8 KB) size.

D. Application Workloads

The memory and disk intensive workloads are two widely used classes of cloud workloads in our experiment. The various characteristics are generated on network traffic for both such as bandwidth, size of packets on network.

The illustration on memory-intensive workloads follows the same as mentioned below:

- Iperf supports for large block data transfer micro benchmarks
- Live VM migration from network- intensive blocks to block data transfers
- Generate request by using key-value store called Redis, with Yahoo Cloud Serve Benchmark (YCSB)

The disk-intensive workload uses following for experiments:

- Micro benchmarks for large disk information transfer using FTP and rsync
- Spark used for distributed information processing

E. Network Conditions

The initial experiment where application uses uncongested network with available bandwidth of 10GigE interface for both disk and memory related workloads. The switches do not face traffic at its background. The experiment assumes, there is a multiple application hosted by server and repeatedly performs operation on the same environment. The bandwidth allocation values are restricted to smaller one by following rate limit and it belongs to same range in public environments.

IV. SYSTEM PERFORMANCE

The Cloud service providers, such as Amazon, IBM soft layer and Microsoft Azure, all provide a virtual information center and provides Virtual machines (VMs) to host end users. Multiple Network Interface Controllers (NICs) can be easily configured with these VMs.

Our experiment is conducted with two distinct setup environments to obtain comparative study among TCP and MPTCP. It assumes cluster environment on the public cloud systems.

We experimented using AWS (Amazon Web Server) cloud infrastructure with two VMs having capacity of 4GB RAM, 2VCPU, 100GB drive space and two interfaces connected to different subnets. As mentioned earlier our experimentation is done in AWS public cloud system, therefore we observed variations in the bandwidth on both TCP and MPTCP setup. This bandwidth test was carried out using ipref tool.

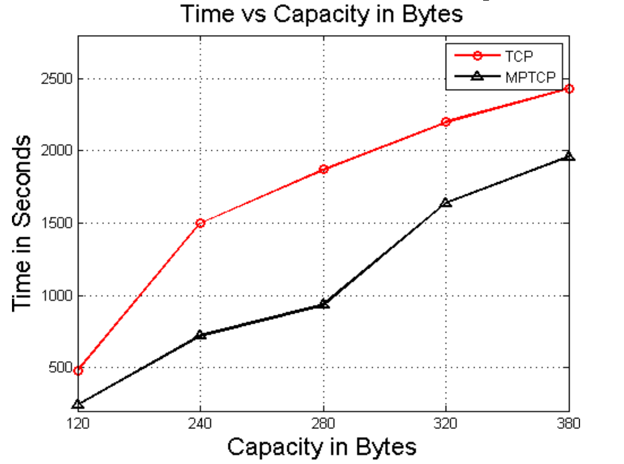
Our analysis was broadly classified into two categories

- Link Handover
- Throughput

Link Handover: We used ifstat tool to monitor the traffic flow in all interfaces. Started downloading scripts in both TCP and MPTCP setups. One of the network interface was removed while data is being downloaded. The observation was TCP setup failed to switch the traffic flow to the working interface, whereas the MPTCP smoothly handled over working interface.

Figure 4. Represents the performance of MPTCP over TCP

Throughput: Since MPTCP started downloading from all the available network interfaces it increases the speed than the



TCP. The Table 1 shows the communication delay happened for both TCP and MPTCP to access the files of different capacity.

Table 1. Comparison of network delay obtained while transferring the data between MPTCP and TCP

File Size	TCP	MPTCP
	Time in Seconds	Time in Seconds
120 MB	480	240
240 MB	1500	720
280 MB	1870	934
320 MB	2205	1639
380 MB	2434	1958

V. CONCLUSION

This work reveals the experimental studies the performance of Multi path TCP's with multiple network conditions. The work concerned on MPTCP usage in most effective and feasible way for cloud and data centre environments over various conditions on network. The experiment is conducted by clustering the public cloud data using k-means algorithm and communicated over a network using MPTCP. The results show that the proposed method yields high-speed data transfer and low communication delay when compare to traditional TCP technique.

REFERENCES

1. A. Vahdat. M. Al-Fares and A. Loukissas, A scalable, commodity data center network architecture. In ACM SIGCOMM Computer Communication Review, volume 38, ACM, 2008, pages 63–74.
2. T. Edsall, M. Alizadeh, R. Vaidyanathan, S. Dharmapurikar, et al. Conga: Distributed congestion-aware load balancing for datacenters. In ACM SIGCOMM Computer Communication Review, volume 44, 2014, pages 503–514.
3. J. Padhye, M. Alizadeh, D. A. Maltz, Patel, et al. Data center tcp (dctcp). In ACM SIGCOMM computer communication review, volume 40, ACM, 2010, pages 63–74.

4. C. Paasch, O. Bonaventure and S. Barre. Multipath tcp: from theory to practice. NETWORKING 2011, pages 444–457.
5. D. A. Maltz, A. Akella, and T. Benson. Network traffic characteristics of data centers in the wild. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pages 267–280.
6. R. Netravali, S. Deng, H. et al , Wifi, lte, or both?: Measuring multi-homed wireless internet performance. In ACM IMC, 2014, pages 181–194.
7. D. A. Hayes, S. Ferlin, O. Alay, and et al. Revisiting congestion control for multipath tcp with shared bottleneck detection. In IEEE INFOCOM, 2016, pages 1–9.
8. C. Raiciu, A. Ford, M. Handley, TCP extensions for multipath operation with multiple addresses. RFC 6824, 2013.
9. J. R. Hamilton, A. Greenberg, D. A. Maltz, P. Patel, et al.Vl2: a scalable and flexible data center network. In ACM SIGCOMM computer communication review, volume 39. ACM, 2009 , pages 51–62.
10. J. Iyengar, S. Hassayoun , and D. Ros. Dynamic window coupling for multipath congestion control. In IEEE ICNP, 2011, pages 341–352.
11. D. Kreutz, F. M. Ramos, S. Azodolmolky, et al. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 2015,pages 14–76.
12. M. A. Razzaque, T. A. Le, S. Lee, and H. Jung. ecmtcp: an energy aware congestion control algorithm for multipath tcp. IEEE communications letters, 16(2), 2012, 275–277.
13. R Rasti, K. Ousterhout, et al. V. ICSI. Making sense of performance in data analytics frameworks. In NSDI, volume 15, 2015, pages 293–307.
14. S. Ferlin, C. Paasch, et al.. Experimental evaluation of multipath tcp schedulers. In Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop. ACM, 2014, pages 27–32.
15. M Handley, C. Pluntke, et al. Improving datacenter performance and robustness with multipath tcp. In ACM SIGCOMM Computer Communication Review, volume 41, ACM, 2011, pages 266–277.
16. M. Handly, D. Wischik and C. Raiciu. Coupled congestion control for multipath transport protocols. RFC 6356, Oct 2011.
17. C. Paasch, C. Raiciu, et al. How hard can it be? Designing and implementing a deployable multipath TCP. 2012.
18. A. Tumanov and C. Reiss. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In Proceedings of the Third ACM Symposium on Cloud Computing, SoCC '12, New York, NY, USA, 2012. ACM,pages pages 7:1–7:13.
19. P. Stuedi and A. Trivedi, On the [ir] relevance of network performance for data processing. In 8th USENIX Workshop on Hot Topics in Cloud Computing (Hot Cloud 16), Denver, CO, 2016. USENIX Association.
20. D. Wischik, M. B. Braun and M. Handley, The resource pooling principle. ACM SIGCOMM Computer Communication Review, 38(5), 2008, pages 47–52.
21. C. Raiciu, D. Wischik, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In NSDI, 2011.
22. R. Miao, K. Zarifis, M. Calder, et al. Dibs: Just-in-time congestion mitigation for data centers. In Proceedings of the Ninth European Conference on Computer Systems. ACM, 2014, page 6.
23. J. Liu, J. Zhao, H. Wang, and C. Xu. Multipath tcp for datacenters: From energy efficiency perspective. In INFOCOM 2017-IEEE Conference on Computer Communications, IEEE IEEE, 2017, pages 1–9.

AUTHORS PROFILE

Shashikala S V, Research Scholar, Department of Computer Science & Engg., BGSIT, with 20+ years of teaching. She received M Tech degree from Visveswaraya Technological University(VTU) and currently pursuing Ph D in VTU under guidance of Dr. Ravikumar G K . She has published 20+ technical papers in national and international journals and her area of interest are Programming, Big data, Cloud computing, Machine learning and Networking. She has presented papers in national and international conferences. She is a Life member of ISTE , Institute of Engineers and member of CSI.

Dr. Ravikumar G.K , Professor & Head of Research Center, Department of CS&E, ACU-BGSIT, having 20+years of experience with 10+ years of experience in Industry and 10+ years of experience in academics. Presently he is guiding Six Research Scholars under Visveswaraya Technological University(VTU) and his area of interest are Big Data, Data Mining, Cloud Computing, Data Masking, Test Data Management, Data Migration. He has published 50+ technical papers in National and International Journals and he has filed for 10 patents. He is a Life member of ISTE, CSI and IAENG(International Association of Engineers).