Design and Development of Efficient Multipath TCP using GMM clustering for Big Data in Public Cloud Data Center

Shashikala S V, Ravikumar G K

Abstract*: Currently new topologies have been introduced in many data centers that provide location independence and larger aggregate bandwidth by making different multiple paths in the network core. This work proposes transport of data from TCP (Transmission control Protocol) to 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. Further, the work also 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 Gaussian Mixture Model (GMM) based Expectation and Maximization (EM) 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* : TCP, MPTCP, Data center, GMM**

I. INTRODUCTION

Many Companies like Amazon, Google and Microsoft have constructed data centres of unprecedented size which can be used to run many complex cloud applications. These complex applications are spread across several machines but undertaking this, will degrade the life of network fabric present within the data center. In a point to point communication traffic, bulk quantities of data has been transmitted, example, transferring of distributed file system such as GFS. The applications such as Dryad, BigTable or MapReduce exchange large amount of information between several machines. It is important to maintain the role of each machine by preventing the hot spots creation in network fabric, in order to get maximum flexibility when running a new application.

Nowadays, Data center networking has evolved as an important attention because now data centers became an important case in their own right way, and also mainly, it simultaneously reveals the physical network topology and the routing traffic. In order to allow many operators to implement the functionality of the application in a location independence manner, new denser network topologies like VL2 [5] and FatTree [1] have been implemented. However, such a denser network results in difficulty for direction-finding. How to ensure distribution of load between multiple paths is uniform with independent of the communication traffic?

Cloud data centres 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). Multi path TCP is an IETF expansion [4] 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

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 minimises data blockage inside the data centres or problems in network handler 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 different network interfaces (NICs) to develop rep routes to different servers.

In this work we proposes a development of data center transport from TCP (Transmission control Protocol) to multi-path TCP (MPTCP) for maximum utilization of paths over network flow. Further, the work also 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 Gaussian Mixture Model (GMM) based Expectation and Maximization (EM) algorithm and communicated over a network using MPTCP.

Revised Manuscript Received on February 14, 2020.
***** Correspondence Author
Shashikala S V, Research Scholar, Dept. of CSE, BGSIT, BG Nagar, Mandya, INDIA. Email: shashisv7@gmail.com
Dr. Ravikumar G K, Professor & Head, R & D Center, Dept. of CSE, BGSIT, BG Nagar, Mandya, INDIA. Email: ravikumargk@bgsit.ac.in

The performance of MPTCP is evaluated over traditional TCP running in public data center. We determine that MPTCP performance in terms of high-speed, data transfer and low communication delay for soft clustered data is better when compared to traditional TCP. Further, we also show that multipath TCP can utilize many topologies and traffic patterns where single path TCP cannot.

II. BACKGROUND SURVEY

Multipath TCP is an extension of TCP that empowers concurrent transmission of information from one end to other end across several paths. For an example, like cellular phone where multi-path TCP allows several applications to transfer and receive information across several interfaces, like cellular and Wi-Fi, by building single TCP sub flow for every interface [14] MPTCP scheduler makes use of this sub flow for transmit and receive of the information. The design of MPTCP has several benefits like resource usage, throughput and soft reaction to bugs good performance with paths. Figure 1 shows the MPTCP architecture.

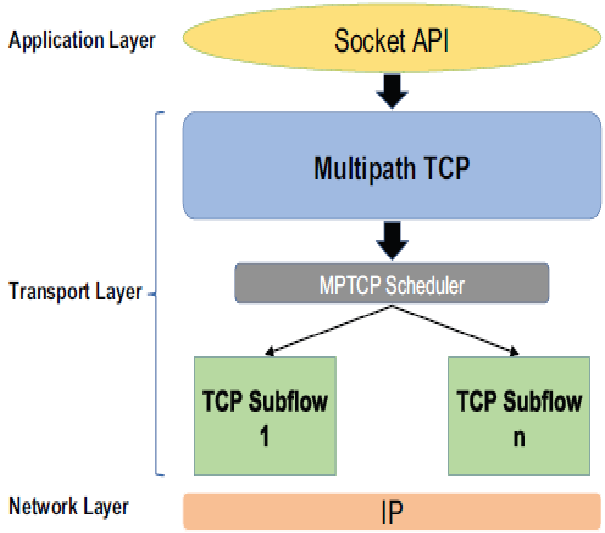

Fig. 1. Depicts the design of MPTCP.

The sub flow with shortest round trip time (RTT) is selected by a scheduler [15] in order to send the data provided, if several sub flows are available. Out of all sub flows, the sub flow whose congestion window is not yet full, the scheduler selects that path with lesser round trip time, during the transmission of the segment. The scheduler builds an inclination towards one of the interfaces if there is more than one such path, Further, it continues to send the information on the particular sub flow till the congestion window of that sub flow turns out to be full.

MPTCP has analysed in mobile context in the previous work. In [16] the authors studied the mobility effect on MPTCP, while in [17] the authors presented a different modes of MPTCP to be utilized by cell phones for Wi-fi handover. However, with respect to lossy sub flows, heterogenetic nature of path is not investigated in neither of the work. In [18] the authors presented a comparative study between single TCP and multi path TCP. In [9] the authors calculated the importance of scheduler architecture on performance by deploying various scheduler algorithms. Whereas different congestion mechanism algorithms are compared in [20][21][22][23].In [24] the authors introduced scheduler algorithms that picks sub flows dependent on an estimation of the level of traffic they can deal with before getting congested. The methodology reveals by considering limited buffering quantity by considering huge transfers in a network. In [25] authors presented another new scheduling policy was maintain a strategic distance from out-of-request segments. However, the removal of segment from TCP buffer after it is transmitted by another sub flow is not explained by the authors.

In [26] on the ns-2 simulator platform the authors presented a delay-aware packet scheduler algorithm. The proposed method evaluates the path heterogeneity both in delay and stable conditions. In [27] the authors evaluated the performance of MPTCP over mobile network. The study has been carried out by considering various sub flows and detailed statistics like round trip time (RTT) and out of order delivery, but the authors did not consider the lossy sub flows. Lim et al. presented a new scheduler algorithm that send buffers by monitoring the available bandwidth on every sub flow, however, it doesn't exploit the data loss rate on every single sub flow.

III. NETWORK DATA CENTER

From the Top level, networking architecture in Data centers are mainly divided into four components namely

- Physical networking
- Routing network
- Paths selection by routing
- Selected paths traffic congestion control.

Each component in Data center networking are inter dependent on one another. Each component we will discuss in detail and also it is worth to mention since MPTCP reveals both traffic congestion control and path selection control.

A. Network Topology

Generally, hierarchical network topologies have been used for building data centers, that is, racks of hosts interface with a top-of-rack switch, these switches associate with aggregation switches and these are associated with a center switch. If the majority of traffic flows in and out of the data center, than the use of such network topologies make sense. There will non uniform sharing of bandwidth if majority of the traffic is within datacentre (Intra).

These restrictions are overcome by recent proposals. In order to provide complete bandwidth between two hosts in a network, the topologies like Clos [3], VL2 and FatTree uses several center switches. They themselves differ, where FatTree makes use of bulk quantity data having low speed links whereas VL2 will make use of lesser faster links. Conversely, in BCube [6], the hierarchy is deserted in support a hypercube-like topology. Each of the these proposals take care of the traffic issue at physical level that is sufficient capability with regards to each host to be able to transmit flat-out to another randomly selected host. The issue of routing of traffic depends on density of interconnection.

B. Network Routing

Multiple number of parallel paths are generated between pair of hosts when the topology is densely interconnected. Host may not be knowing the load level of these paths or which path is less loaded, hence the network routing system should distribute traffic over these paths. The dynamic load balancing technique is used, where each stream is allocated an arbitrary path from the available paths.

In today's switches dynamic load balancing can be implemented in several ways practically. For an instance, in order to send ECMP, a switch can utilize a protocol called link state routing protocol then, flow will be divided similarly across equivalent length paths depending on a hash of the five-tuple in every data packet. VL2 gives simply such control over a virtual layer 2 foundation.

Most of the topologies have varying path length like Bcube, where simple ECMP can't get to a large number of such paths since it just hashes between the shortest paths. This problem can be solved by uncovering all underlying network paths [7] by utilizing numerous paths provided by various static VLANs. Either the host or the primary hop switch would then be able to hash the five tuple to figure out which path is utilized.

C. Network Path Selection

Dynamic load balancing mechanism is selected as the default path selection system by ECMP or various VLANs. However, in many of the topologies the Dynamic load balancing mechanism can't accomplish the full cross-sectional network bandwidth nor it is particularly reasonable. This is because many hot-spots will create for dynamic selection, where few links will get under loaded for an unfortunate combination of dynamic selection and some links will have little or zero load.

A centralized flow schedulers is proposed in order to address these issues. In order to maximize or boost the total throughput [2] the existing flows are reassigned and large flows re linked to minimal loaded paths. The performance of the scheduler is good provided, if the flows are limited with exponentially circulated sizes and Poisson appearances, as appeared in Hedera [2].The intention is to utilize the complete bandwidth by scheduling only big flows and also by maintaining lower scheduling cost.

However, the traffic analysis in data centers indicates that flow distribution is not following Pareto distribution system [5].In such scenarios, in order to keep aware of the flow arrivals, the scheduler needs to run continuously or frequently. Still the response time of scheduler is basically limited, since it has to compute placements, extract statistics and trigger initiation, in this scheduling period.

D. Network Congestion Control

Many applications acquire TCP's congestion control system by using single path TCP, which performs very fair operation on the selected path by matching offered load to available capacity. Many recent study proves that tuning TCP to data center has several advantages, like reduction in retransmission timeout [10], but yet the issue TCP solves stays unchanged.

Multi Path TCP makes use of congestion control scheme [9] which works across all sub-flows. One of the design objectives is to modify the present TCP network connections 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 every sub-flow. Along with coupled policy, the literature [10, 11, and 12] 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 [13].

IV. 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 Gaussian Mixture Model (GMM) based Expectation and Maximization (EM) 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 Gaussian mixture model (GMM) Expectation Maximization (EM) clustering that we use to cluster the big data which is loaded in cloud network. EM algorithm belongs unsupervised learning. The objective of this algorithm is to segregate the unlabelled cloud data into N number of groups. Figure 4 and Figure 6 shows the graphical visualization of big data form different number clusters head or centroids.

B. Expectation Maximization

EM is a probability distribution based clustering technique deployed using Gaussian mixture model. The EM method deal with several data points to form multiple clusters, where every cluster form a Gaussian distribution. This results in study on Gaussian mixture model. The univariate one dimensional Gaussian distribution function can be written as in (1)

$$G(x|v,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-v)^2}{2\sigma^2}} \quad (1)$$

Where, v is the average value, σ is the Standard deviation, σ2 is Variance and x is the data point (pixel) with location (m, n) in a two dimensional plane.

The Multivariate Gaussian distribution can be written as in (2). Where Σ is covariance matrix.

$$N(x|v,\Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}}\left\{-\frac{1}{2}(x-v)^T\Sigma^{-1}(x-v)\right\} \quad (2)$$

The mean v and covariance Σ of a Gaussian distribution are estimated in order to extend the multivariate distribution to spread over all the data points. Maximum Likelihood method (ML) is used to estimate mean v and covariance Σ.

Logarithm of Gaussian distribution is calculated for parameter estimation using Maximum Likelihood method as given in (3)

$$\ln P(x|v,\Sigma) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-v)^T\Sigma^{-1}(x-v) \quad (3)$$

The mean v and covariance Σ can be identified using Maximum Likelihood method by applying derivative for (3), with respect to v and Σ and equating it to zero.

$$v_{ML} = \frac{1}{N}\sum_{n=1}^{N} X_n \quad (4)$$

$$\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - v_{ML})(x_n - v_{ML})^T \quad (5)$$

Where N is total number of data points.

The probability distribution for 'x' number of data points for multiple Gaussians can be given in (6)

$$P(x) = \sum_{k=1}^{K} \pi_k N(x|v_k, \Sigma_k) \quad (6)$$

Where, k represents Total number of Gaussians, πk represents Mixing coefficients for the kth Gaussian and $N(x|v_k, \Sigma_k)$ Represents Normal multivariate distribution for the class k. Log likelihood can be obtained from (6) by applying logarithm

$$\ln P(x|v,\Sigma,\pi) = \sum_{n=1}^{N} \ln P(x_n) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k N(x_n|v_k, \Sigma_k\right\} \quad (7)$$

C. Expectation Maximization (EM) technique

The EM algorithm is a looping technique which is performed to find out the values of mean (v_k), covariance (Σ_k) for 'k' number of clusters. It consists of mainly two steps: Expectation step and Maximization step

Step 1: Initialize the parameters mean (v_j), covariance (Σ_j) by calculating the histogram of the image for k number of clusters. Calculate the log likelihood from (8)

$$\ln P(X|v,\Sigma,\pi) = \sum_{n=1}^{N} \ln P(X_n) = \sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{K} \pi_j N(X_n|v_j, \Sigma_j)\right\} \quad (8)$$

Where N is the total number of data points or pixels, P is the probability for the distribution for the samples X, $N(X_n|v_j, \Sigma_j)$ Normal multivariate Gaussian distribution for the cluster j.

Step 2: E-Step - Using the log likelihood for all observed and unobserved data obtained in step 1, compute the expected probability of unobserved data. Hence the step Expectation given in (9).

$$\lambda_k(X) = \frac{\pi_k N(X|v_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(X|v_j, \Sigma_j)} \quad (9)$$

Step 3: M Step - Update the parameters.

$$V_j = \frac{\sum_{n=1}^{N} \lambda_k(X_n) X_n}{\sum_{n=1}^{N} \lambda_j(X_n)} \quad (10)$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \lambda_k(X_n)(X_n - v_k)(X_n - v_k)^T}{\sum_{n=1}^{N} \lambda_j(X_n)}$$

(11)

$$\pi_j = \frac{1}{N}\sum_{n=1}^{N} \lambda_k(X_n) \quad (12)$$

Step 4: Calculate the log likelihood from (13).

$$\ln P(X|v,\Sigma,\pi) = \sum_{n=1}^{K} \ln\left\{\sum_{k=1}^{K} \pi_k N(X_n|v_k, \Sigma_k)\right\} \quad (13)$$

D. Public Cloud setup

Our experiment is conducted with two distinct setup environment 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.

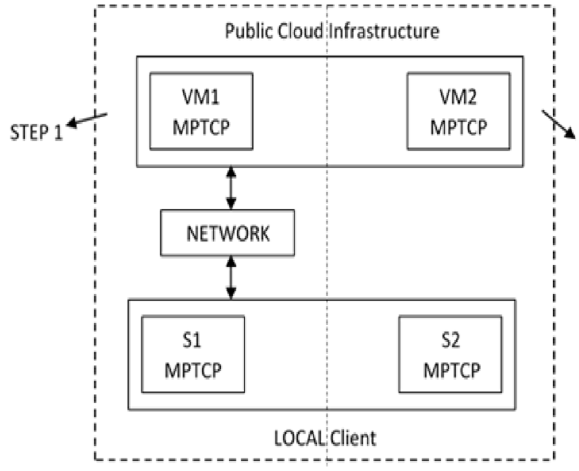

Fig. 2.Shows the proposed experimental setup

Two VMs are named as VM1 and VM2 as shown in Figure 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.

E. 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. 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.

F. 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

G. 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.

V. RESULT

The Cloud service providers such as IBM soft layer, Amazon and Microsoft Azure, all provide a virtual information centre 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 environment 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.

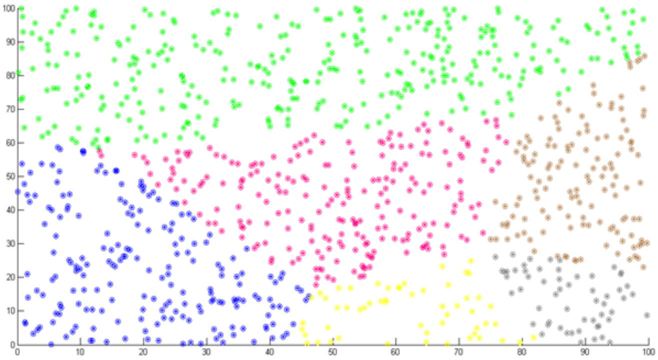

Fig. 3.Graphical visualization of big data form six clusters head or centroids.

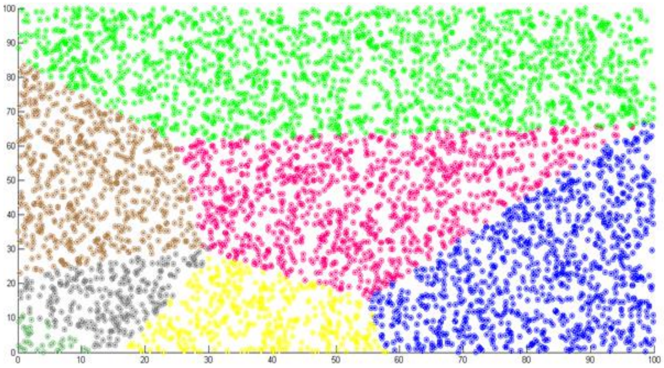

Fig. 4.Graphical visualization of big data form seven clusters head or centroids.

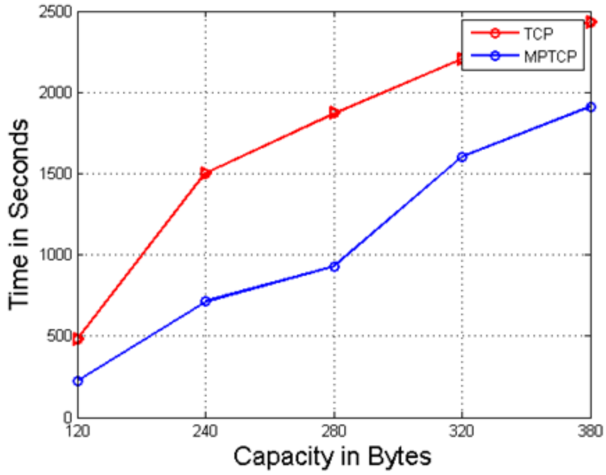

Fig. 5.Represents the performance of MPTCP over TCP

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.

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.

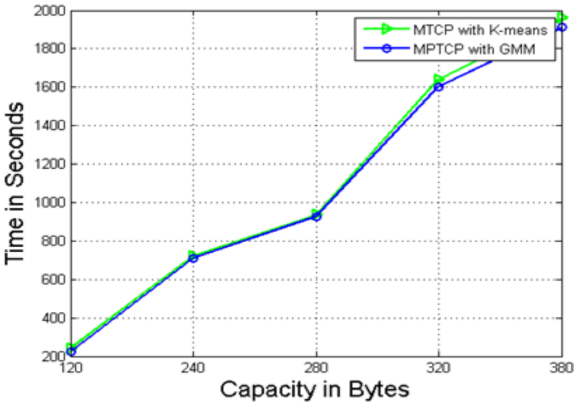

Fig. 6.Represents the performance of MPTCP over different clustering algorithms

Fig. 7.Table 1. Comparison of network delay obtained while transferring the data between MPTCP and TCP

File Size	TCP Time Seconds	MPTCP Time Seconds
120 MB	480	223
240 MB	1500	712
280 MB	1870	928
320 MB	2205	1604
380 MB	2434	1912

VI. 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 Gaussian Mixture Model (GMM) based Expectation and Maximization (EM) 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.

REFERENCES

1. M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In Proc. SIGCOMM 2010.
2. M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic flow scheduling for data center networks. In Proc. Usenix NSDI 2010.
3. C. Clos. A study of non-blocking switching networks. Bell System Technical Journal, 32(5):406–424, 1952.
4. A. Ford, C. Raiciu, M. Handley, and S. Barre. TCP Extensions for Multipath Operation with Multiple Addresses. Internet-draft, IETF, 2009.
5. A. Greenberg el al. VL2: a scalable and flexible data center network. In Proc. ACM Sigcomm 2009.
6. C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. Bcube: a high performance, server-centric network.
7. J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul. Spain: Cots data-center ethernet for multipathing over arbitrary topologies. In Proc. NSDI 2010.
8. V. Vasudevan et al. Safe and effective fine-grained tcp retransmissions for datacenter communication. In Proc. SIGCOMM 2009.
9. M. Handly, D. Wischik and C. Raiciu. Coupled congestion control for multipath transport protocols. RFC 6356, Oct 2011.
10. J. Iyengar, S. Hassayoun , and D. Ros. Dynamic window coupling for multipath congestion control. In IEEE ICNP, 2011, pages 341–352.
11. 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.
12. M. Handley, D.Wischik, and C. Raiciu. Balancing resource pooling and equipoise in multipath transport. ACM SIGCOMM, 2010.
13. 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.
14. A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses,” RFC 6824, 2013.
15. C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, “Experimental Evaluation of Multipath TCP Schedulers,” in Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop. ACM, 2014, pp. 27–32.
16. C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, “Opportunistic Mobility with Multipath TCP,” in Proceedings of the 6th International Workshop on MobiArch. ACM, 2011, pp. 7–12.
17. C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, “Exploring Mobile/Wi-Fi Handover with Multipath TCP,” in Proceedings of the 2012 ACM SIGCOMM workshop on Cellular Networks: Operations, Challenges, and Future Design. ACM, 2012, pp. 31–36.
18. S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, “Wi-Fi, LTE, or Both?: Measuring Multi-homed Wireless Internet Performance,” in Proceedings of the 2014 Conference on Internet Measurement Conference. ACM, 2014, pp. 181–194
19. B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo, “Impact of Path Characteristics and Scheduling Policies on MPTCP Performance,” in 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2014. IEEE, 2014, pp. 743–748.
20. R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec, “MPTCP is Not Pareto-Optimal: Performance Issues and a Possible Solution,” in Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. ACM, 2012, pp. 1–12.
21. D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, “Design, Implementation and Evaluation of Congestion Control for Multipath TCP.” in NSDI, vol. 11, 2011, pp. 8–8.
22. C. Raiciu, D. Wischik, and M. Handley, “Practical Congestion Control for Multipath Transport Protocols,” University College London, London/United Kingdom, Tech. Rep., 2009.
23. K. Noda and Y. Ito, “Proposal of Novel MPTCP Congestion Control to Suppress QoS Fluctuation for WebQoE Improvement,” in 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCEBerlin). IEEE, 2018, pp. 1–3.
24. F. Yang, P. Amer, and N. Ekiz, “A Scheduler for Multipath TCP,” in 2013 22nd International Conference on Computer Communication and Networks (ICCCN). IEEE, 2013, pp. 1–7.
25. F. Yang, Q. Wang, and P. D. Amer, “Out-of-Order Transmission for InOrder Arrival Scheduling for Multipath TCP,” in 2014 28th International Conference on Advanced Information Networking and Applications Workshops. IEEE, 2014, pp. 749–752.
26. N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, “DAPS: Intelligent delay-aware packet scheduling for multipath transport,” in 2014 IEEE International Conference on Communications (ICC). IEEE, 2014, pp. 1222–1227.
27. Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, “A Measurement-based Study of Multipath TCP Performance over Wireless Networks,” in Proceedings of the 2013 Conference on Internet Measurement Conference. ACM, 2013, pp. 455–46

AUTHORS PROFILE

Shashikala S V Research Scholar, Department of Computer Science & Engg., BGSIT, with 25+ years of teaching. She received M Tech degree from VTU in 2005 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.

Dr. Ravikumar G K ,Professor & Head of Research Center, Department of CS&E, 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 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.