Throughput Analytics of Cloud Networks

Derek Phanekham, Suku Nair
AT&T Center for Virtualization
Southern Methodist University
dphanekham@smu.edu, nair@smu.edu

Nageswara S. V. Rao
Oak Ridge National Laboratory
raons@ornl.gov

Mike Truty
Google LLC
truty@google.com

***Abstract*—A network of virtual machines at cloud server sites connected over virtual IO connections is a flexible, easily deployable, and cost-effective alternative to a physical network infrastructure with dedicated servers connected over leased fiber lines. We study the throughput performance of such a cloud network by collecting measurements over Google Cloud infrastructure spanning multiple continents. To study its ideal performance and impact of packet losses, we utilize its emulation using dedicated servers and connection emulation hardware devices. We compare the measurements over the cloud network to those of its emulation over a testbed. We examine the throughput profiles of both networks as a function of the round trip time and their utilization-concavity coefficients, estimated using measurements for common TCP versions. The throughput profile's concave-convex shape and its coefficient are critical indicators of the network performance, qualitatively and quantitatively, respectively. The results indicate their overall agreement between the production cloud network and its emulation using dedicated connections, and a near optimal throughput performance of the former except for a few under-performing connections. Also, the number of parallel flows is found to be a dominant factor in optimizing the throughput across various conditions and TCP versions.**

***Index Terms*—throughput profile, cloud networks, dedicated connection, TCP version, network emulation, concave-convex**

I. Introduction

Data center and scientific computing applications are becoming increasingly distributed across geographically separated compute and storage facilities. Computing and data nodes may be dispersed across different sites to serve various clients, for example, to widely disseminate scientific data and support web searches. Also, there have been recent industry trends towards executing workloads over public clouds with rented virtual and physical servers, storage space and network access, and also over private data centers and supercomputers connected over dedicated networks. For these infrastructures with geographically distributed sites, there is need to ensure effective transfers of various data, which could be in the form of codes, containers, virtual machines, measurements, and input/output data sets.

Cloud networks composed of virtual machines (VMs) at cloud sites connected over virtual IO and connections are flexible, easily deployable alternatives to more traditional network infrastructures, which involve installing and maintaining dedicated servers and network devices connected over leased fiber lines. Public cloud networks and private networks typically utilize shared and dedicated connections, respectively, but have the same goal of maximizing the throughput performance. Our motivation is to study their throughput performance, in particular, the effectiveness of optimization methods (e.g. TCP version, IO and buffer tuning and parallel flows) and analytics (e.g. geometry and machine learning aspects of throughput profiles).

We study the performance of these cloud networks using throughput measurements for memory transfers at different round trip times (RTTs) over the Google Cloud infrastructure; we utilize 10 Gbps internal IP connections for 1-10 parallel flows of multiple TCP versions. We complement these measurements with hardware-emulated testbed measurements collected over connections with similar RTTs under controlled packet loss rates. The throughput profile as a function of RTT and its utilization-concavity coefficient (C_{uc}) [1] are estimated using the measurements. Qualitatively, the profile's concave shape indicates a higher level of optimization, and convexity represents "under-performance", typically due to limitations such as IO bottlenecks and insufficient buffer sizes. The profile's $C_{uc} \in [0,1]$ quantifies the optimization level by taking into account both the achieved throughput across different RTTs and the profile's shape. Our results indicate a near optimal throughput performance of both cloud networks with $C_{uc} \approx 0.7$, which matches that of an ideal emulation under no external losses. Overall, they also indicate a comparable performance of cloud networks as indicated by the similarity of concave-convex geometry of both profiles.

This paper is organized as follows: In Section II, we briefly describe existing works on transport performance over different networks. Then, we describe our measurements collection for both Google Cloud and testbed connections in Section III. In Sections IV and V, we compare the two networks using throughput profiles and C_{uc}, respectively.

II. Related Work

The area of achieving high throughput over wide area networks has been previously examined by several groups. In [2], the authors looked at performance of high bandwidth optical networks and tested different TCP congestion control algorithms and parameters. They found that Scalable TCP (STCP) performs best for single flow, and CUBIC is most fair. There are several other studies that also compare the efficacy of different TCP variants across different types of networks [3] [4]. Additional studies focus on the BBR (Bottleneck Bandwidth and Round-trip propagation time) TCP variant, which we also examine in our work. Crichigno et al. [5] examine the effects of maximum segment size and the number of parallel streams on large file transfer performance when using BBR, which is found to provide larger performance increase than other TCP variants with multiple parallel streams. Jaegar et al. [6] studied how to take reproducible measurements of TCP BBR and Yi Cao et al.

There has also been work in creating and using coefficients of throughput profiles for comparing different networks [7] [8]. They include the utilization concavity-coefficient, which we use to compare our network throughput profiles in Section V.

III. Measurements Collection

We compare throughput and RTT measurements of these networks using measurements collected over a public cloud network, which employs both public and private IP addresses under production conditions over the world-wide Google Cloud infrastructure shown in Fig. 1. Dedicated versions of their connections are emulated under ideal and controlled loss conditions over an emulated testbed.

A cloud network connection is between two VMs in two possibly different regions, referred to as a region pair, and a corresponding testbed connection is between two dedicated servers connected via a hardware emulator with programmable RTT and packed loss rate.

A. Google Cloud: Data Center Connections

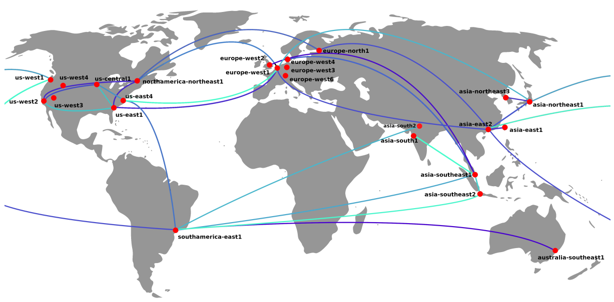


Fig. 1: Google Cloud with lines representing logical connections between region pairs, with RTTs in [1-350] ms range.

Over the Google Cloud network, we set up n1-standard-4 VMs with 4 vCPU cores on Intel Skylake processors in several cloud regions, which are distributed worldwide. Their OS is Ubuntu 20.04 with Linux kernel version 5.8.0 with TCP max send and receive buffers set to 250 MB. They use Virtio virtualized NICs, and we do not have access to the actual hardware specifications for the server or the network. They are networked using a virtual private cloud (VPC) on the Google Cloud network. They can send data at allocated 10 Gbps across this virtual network; the physical link is likely capable of much higher rates. There is possible cross traffic from other machines on VPC network that share the same physical link and other VMs on the same machine. Our measurements consisted of throughput and RTT using *iPerf* and *ping* at different RTTs in [1-350] milliseconds (ms) range. Three TCP variants, CUBIC [9], HTCP [10], and BBR [11] are tested across the VPC network using internal IP addresses for addressing. There is a 10 Gbps sending limit per machine as can be seen in Fig. 2a-c.

The VPC connections are emulated in the tesbed at RTTs that closely match the measurements. Measurements are collected on 32-core Supermicro workstations with Redhat 7 and 8 Linux kernels. Hosts with identical configurations are connected over 10 GigE connections emulated by IXIA hardware emulators with programmable RTT and periodic packet drop rate. Ethernet packets are generated at testbed hosts similar to VPC VMs, and are delayed by RTT and periodically dropped at the specified values by IXIA emulator. Similar to the public cloud measurements, here we collect TCP memory-to-memory throughput measurements for multiple TCP congestion control modules using *iPerf*, including CUBIC, HTCP, and BBR.

The collection of measurements is automated with each set taking 1-2 days. TCP buffer sizes are set to recommended values for 200 ms RTT and the socket buffer parameter for iPerf is set to 2 GB. Thus, their measurements do not embody the effects due to cross traffic or VMs that can manifest and dominate in some VPC connections.

IV. Throughput Profiles

To assess the performance of network transfers, it is helpful to understand the best case and measured throughput between clients and servers (namely, VMs and physical servers in cloud and testbed networks, respectively) over connections with various RTTs. To achieve this objective, we study the throughput and RTT characteristics, and the impact of protocol parameters and network losses, which are incidental and unknown in Google Cloud but are controlled in the testbed.

The TCP throughput profile is determined by: (i) protocol parameters including version representing BBR, CUBIC, or HTCP, the number of parallel flows n, the buffer sizes assumed to be set for high throughput, and (ii) connection RTT, and modality and capacity, e.g., Ethernet at 10Gbps in our case. Let $\theta(\tau, t)$ denote the aggregate throughput at time t over a connection of RTT, τ. The throughput profile based on observation time T_O is given by

$$\Theta(\tau) = \frac{1}{T_O} \int_0^{T_O} \theta(\tau, t) dt \qquad (1)$$

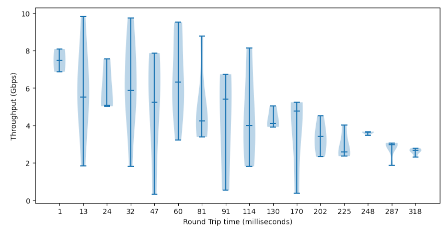
A function $f(\tau)$ is *concave* [12] in interval I if for any $\tau_1 < \tau_2 \in I$, the following condition is satisfied: for $x \in [0, 1]$

$$f(x\tau_1 + (1-x)\tau_2) \geq x f(\tau_1) + (1-x) f(\tau_2).$$

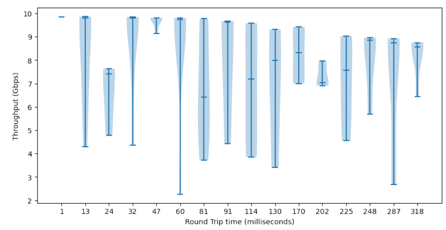
It is *convex* if $\geq$ in the above condition is replaced by $\leq$. In general, profile's shape indicates the optimization level of entire infrastructure [7]: its concavity indicates a higher level of optimization, and convexity represents "under performance", typically due to limitations such as IO bottlenecks and small buffer sizes. For example, the concave region expands as more parallel flows are employed as shown in Fig. 2.

A. Public Cloud Network

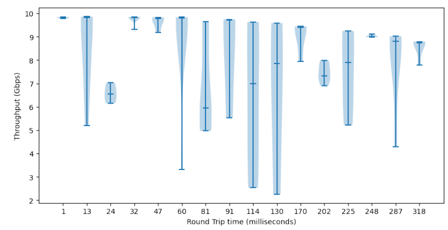
For our VPC network on Google Cloud, the RTTs measurements have a wide range. As the VM pairs under test



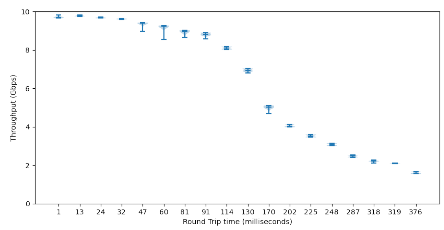
(a) public internal IP 1 flow



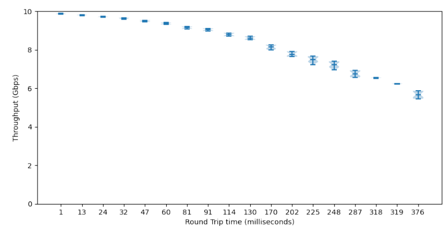
(b) public internal IP 5 parallel flows



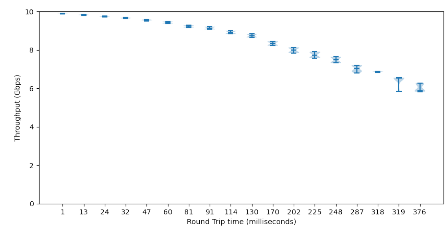
(c) public internal IP 10 parallel flows



(d) emulated network 1 flow



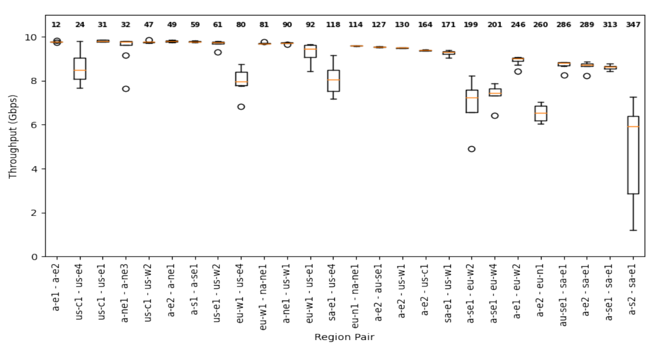
(e) emulated network 5 parallel flows



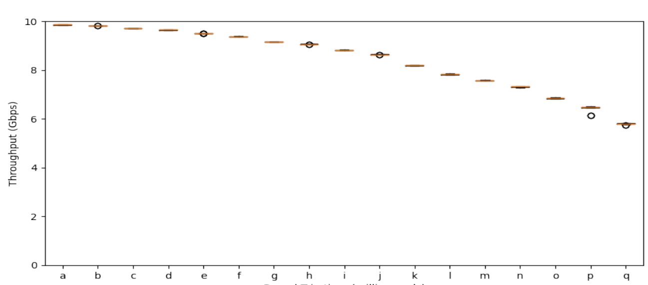
(f) emulated network 10 parallel flows

Fig. 2: Throughput profiles of public cloud network using internal IPs for routing and their corresponding emulated testbed measurements for CUBIC.

become farther apart geographically, their RTT increases as expected. The longest connection has a significant variation in RTT, [340.3-361] ms. At very low RTT (1ms or less), the maximum possible throughput is achieved with low variance As RTT increases, the throughput decreases with generally increasing variance, as seen in Fig. 2a-c.



(a) Google Cloud connections



(b) Testbed connections

Fig. 3: Throughput measurements of BBR with 10 flows

The VPC throughput profiles are not as smooth as those of emulated networks, which is an indication of varying competing traffic on the connection and at end systems. Despite non-smoothness, there is an overall similarity in shape between these and testbed throughput profiles. Overall, the throughput decreases as RTT increases, and improves significantly with additional parallel flows as indicated by the expanded concave regions.

There are a few interesting aspects of VPC connections indicated in Fig. 2a-f. First, there is high variance at some but not all RTTs. In particular, some VM pairs or connections have higher throughput variance or significantly different throughput despite having almost the same RTTs. For example, for connections with average RTT of 80 and 81 ms, the average throughput is 7.96 Gbps and 9.72 Gbps, respectively. Factors contributing to lower throughput include: congestion related losses due to competing traffic on network connections and at client and server hosts; buffer overflows at hosts and intermediate switches and routers; packet and bit errors; and others.

1) Congestion Control: For VPC network, we again tested 3 congestion control algorithms: HTCP, CUBIC, and BBR, whose average throughput profiles are shown in Fig. 4. They all generally performed well at low RTT. For 10 flows, their profiles are comparable, except for a few outliers. BBR achieves higher throughput with lower variance than HTCP and CUBIC, and its throughput profile decreases more slowly with increasing RTT. For all three TCP variants, as we increase the number of flows from 1, 5, to 10, the throughput profile becomes higher, smoother and more concave.

Under internal IPs, with 1 flow, there is a fair amount of variance and the throughput decreases from near 10 Gbps at 1 ms RTT to just above 4 Gbps at 340 ms RTT; this profile is slightly convex. As we increase the flows to 10, slope is shallower and throughput remains around 10 Gbps for higher

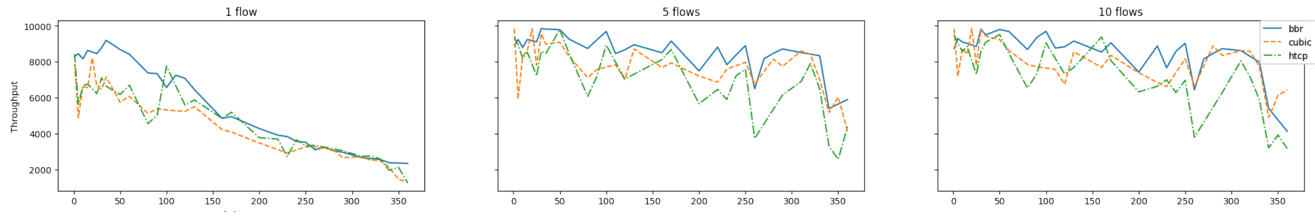


Fig. 4: Throughput profiles for measurements on public internal cloud networks for CUBIC, HTCP and BBR.

RTT, in contrast with 1 flow. Around 300 ms of RTT, there is higher variance in the throughput, but it remains fairly high around 7 Gbps for BBR and HTCP and 8 Gbps for CUBIC. The throughput curves for 5-9 flows look very similar to that of 10 parallel flows (not included here due page limit). From this we can reasonably conclude that there is a optimal number of flows to use, and more will not increase the throughput substantially.

B. Testbed Network Profiles

We emulate VPC connections (or region pairs) using the testbed with the same capacity of 10 Gbps and TCP buffer sizes. We collect throughput measurements for the same average RTTs as the region pairs under different losses introduced by IXIA network emulator; loss rate is periodic specified as 1 in X packets. We estimate testbed profiles at various loss rates, and the Google Cloud VPC profiles with unknown loss rates aligns with the former at low loss rates. A testbed connection closely replicates TCP flows of the corresponding VPC connection; hosts generate packets in a nearly identical environment of VMs, and they are sent through the edge switches and appropriately processed (delayed and dropped) by a IXIA emulator.

In Fig. 2d-f, for 1,5, and 10 flows, generally, the throughput is more stable at lower RTTs and more flows, as seen in particular for RTTs of 1 and 376 ms. Throughput values at 1 ms RTT have very low variance and consistently reach the maximum bandwidth no matter how many flows are used; whereas, throughput values at 376 ms RTT are generally lower and also have higher variance with longer, fatter tails.

We can also see the effect of using multiple flows on network throughput here. For a single flow, the highest achieved throughput is 3.22 Gbps for a 376 ms RTT, with an average throughput of 1.40 Gbps. With 10 flows the highest throughput achieved for the same RTT is around 6.44 Gbps, with an average throughput of 2.85 Gbps. In both the average and the max case, this is a 100% increase. Additionally, there is a maximum increase adding additional parallel flows can provide, with little to no improvement beyond this number. Specifically, Fig. 2e for 5 flows looks very similar to Fig. 2f for 10 flows.

We should also note that at every RTT the throughput values are skewed towards the top of that RTT's throughput range. There is also a consistently long tail that consists of a few outliers on the low end of the range and the throughput decreases steadily as RTT increases. Some of these extreme outliers may be a product of how the tests were performed or the emulated network itself, as they seem to show up in a consistent pattern in the dataset.

1) Congestion Control: Effects of different TCP variants in the testbed network are seen in In Fig. 5 and Fig. 6. Under the no-loss case, these variants perform nearly identically, and differences become noticeable at loss rate of 1/100000 and higher.

2) Loss rate: We emulated loss rates ranging from 1 in 1,000,000 packets to 1 in 500 packets, as shown in Fig. 7. Under no-loss and low-loss cases, the throughput drops steadily with increasing RTT, in a manner similar to VPC profiles. At a high loss rate, the throughput degrades even at relatively low RTT. The drop is more pronounced when using a single flow and more gradual when using multiple flows. As the loss rate is increased, the profile shape changes. Specifically, the concave shape at low error rate becomes convex with a higher loss rate, leading to a situation where throughput drops more quickly as RTT increases. In the most extreme cases (1 in 500 packets), the throughput is less than 500 Mbps for RTT greater than 1 ms.

3) Number of flows: Besides reducing the error rate, the best way to increase the throughput of a connection is to increase the number of flows. By using more flows, we can significantly improve the throughput, especially at large RTTs. For example, in Fig. 7, when there is no loss, we see an improvement from 1.407 Gbps to 3.923 Gbps at RTT of 376 ms as number of flows is increased from 1 to 10. We see similar relative improvements by increasing the number of flows even when the loss rate is high.

C. Comparison of Google Cloud and Testbed Profiles

The no loss testbed scenario represents the ideal case with no competing traffic and external losses; the testbed RTT measurements show negligible variations compared to those on Google Cloud, whose variations are due to the cumulative effects of VMs and virtual connections. In terms of throughput, testbed measurements are relatively more stable as shown in Fig. 3, which results in a smoother concave profile. VPC measurements show more variation and less smoothness but an overall concave profile, which is a net result of virtualization of hosts and IO and the competing traffic both at hosts and over the connection.

The similarity of VPC and testbed profiles can be observed based on results in Fig. 4 and 5. For 1 flow, the throughput profiles for all TCP variants on the cloud match relatively

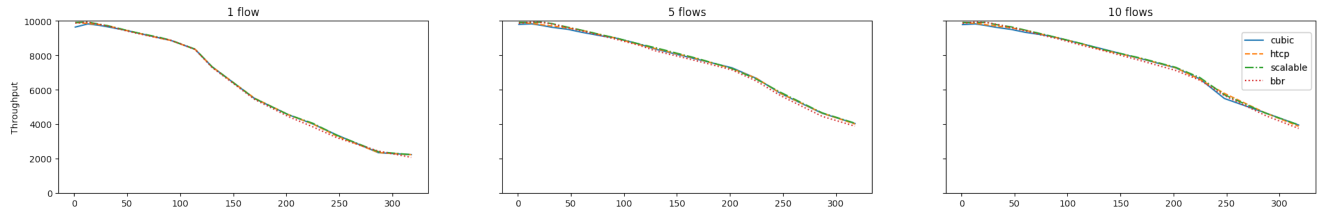


Fig. 5: Throughput profiles for measurements on testbed for CUBIC, HTCP and Scalable TCP with no error

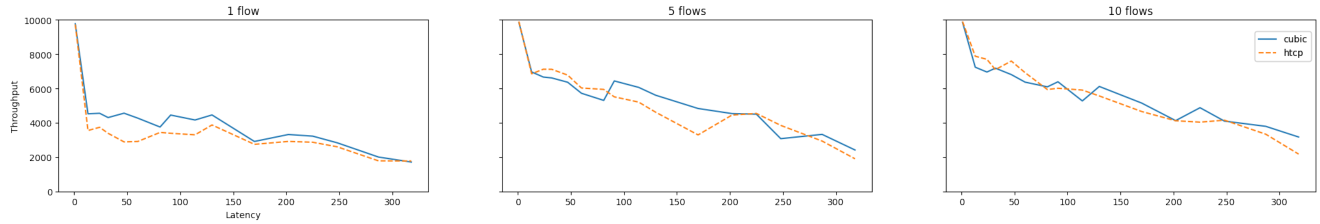


Fig. 6: Throughput profiles for measurements on testbed for CUBIC and HTCP TCP with 1/100000 Error Rate

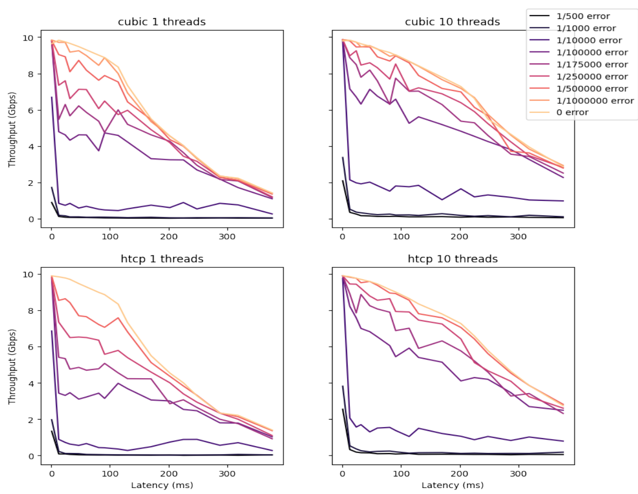


Fig. 7: Effect of loss rate on throughput.

well with those of emulated network, with the exception of there being generally more variance on the cloud network. For 5 and 10 flows, the cloud network is able to maintain higher throughput at higher RTTs than the emulated network. Although the loss rates of VPC profiles are not known, their overall shapes compared to various testbed profiles indicate lower loss rate for most connections, except for the few cases with low throughput and high retransmissions.

V. Utilization Concavity Coefficient

Let L represent the connection capacity, and τ_L and τ_H denote the smallest and largest RTTs, respectively, of the network. The *under utilization coefficient* of $\hat{\Theta}$ is $\mathcal{C}_U(\hat{\Theta}) = \int_{\tau_L}^{\tau_R}\left(L - \hat{\Theta}(\tau)\right)d\tau$. The convex and concave properties of $\hat{\Theta}$ are specified by the area above and below the linear interpolation of $\hat{\Theta}(\tau_L)$ and $\hat{\Theta}(\tau_R)$, respectively. This area is positive for a concave profile and negative for a convex profile. The *convex-concave coefficient* of $\hat{\Theta}$ is defined as [7]

$$\mathcal{C}_{CC}\left(\hat{\Theta}\right) = \int_{\tau_L}^{\tau_R}\left(\hat{\Theta}(\tau) - \left[\hat{\Theta}(\tau_L) + \frac{\hat{\Theta}(\tau_R)-\hat{\Theta}(\tau_L)}{\tau_R-\tau_L}\tau\right]\right)d\tau = (\tau_R-\tau_L)\left[\bar{\hat{\Theta}} - \hat{\Theta}_M\right] \quad (2)$$

Let $\tilde{\Theta} : [0,1] \mapsto [0,1]$ denote a normalized version of $\hat{\Theta}$ such that throughput values are scaled by L, and the operand τ is translated and scaled from interval $[\tau_L, \tau_R]$ to $[0,1]$. The *utilization-concavity coefficient* is defined as [7]

$$\mathcal{C}_{UC}\left(\hat{\Theta}\right) = \frac{1}{2}\left(\left[1 - \mathcal{C}_U\left(\tilde{\Theta}\right)\right] + \left[\frac{1}{2} + \mathcal{C}_{CC}\left(\tilde{\Theta}\right)\right]\right) \quad (3)$$

It takes a simpler form

$$\mathcal{C}_{UC}\left(\hat{\Theta}\right) = \bar{\tilde{\Theta}} - \tilde{\Theta}_M/2 + 1/4 \quad (4)$$

where $\bar{\tilde{\Theta}}$ is the average and $\tilde{\Theta}_M/2$ is throughput at midpoint. The profile's $C_{uc} \in [0,1]$ summarizes the optimization level by taking into account both the achieved throughput across different RTTs and the profile's shape.

A. Coefficients: Ideal and Loss Conditions

We compute C_{uc} for emulated connections under various loss conditions, wherein no loss condition indicates the best case. In Fig. 8, they are shown for different loss rates in our emulated 10 GigE network. The profiles in Fig. 7 illustrate the relationship between C_{uc} and the shape of throughput profile. A C_{uc} closer to 1 indicates a concave throughput profile, for example, the concave shape of profile of 10 flows with 0 loss. A C_{uc} closer to 0 reflects a throughput profile with a convex shape, such as any of profile in Fig. 7 with loss rate $\frac{1}{500}$. There are two overall trends: (i) the coefficient increases with the number of flows indicating the increase in throughput values and expansion of concave shape, and (ii) the coefficient plots become lower as loss rate in increased. Thus, these plots provide a quick, qualitative way to objectively compare profiles from different networks collected under different parameters such as capacity, RTT range, or other characteristics.

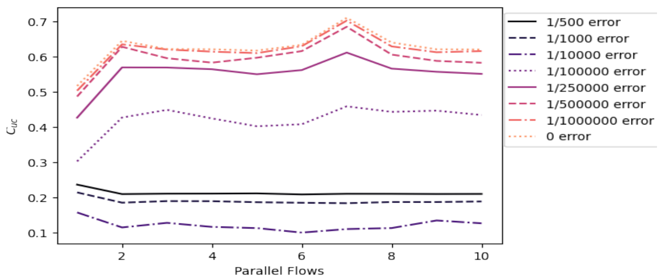


Fig. 8: C_{uc} vs parallel flows in emulated network with CUBIC

B. *Cloud Network Coefficients*

The utilization-concavity coefficients of Google Cloud for three TCP versions are shown in Fig. 9, along with those under no-loss emulated scenarios. It shows C_{uc} for different numbers of parallel flows and different TCP variants in both our public cloud network and our emulated network. This summary plot allows us to directly compare the throughput profiles for these networks across different numbers of parallel flows. For the public cloud network using internal IPs and the emulated 10GigE network we consider a maximum bandwidth of 10 Gbps.

For all connection types, there is a significant increase in the C_{uc} when using more than 1 flow. From 2 to 10 flows, there is a smaller or no increase. BBR on the public cloud network increases the most when using multiple flows which is consistent with findings in the study on BBR [5]. These

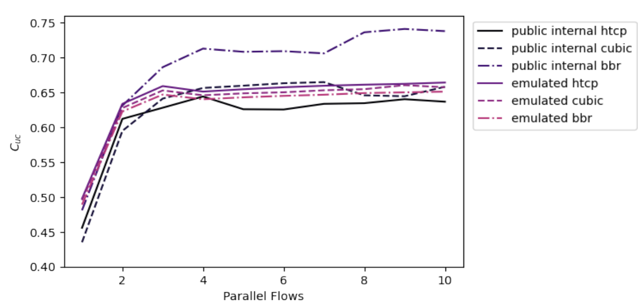


Fig. 9: Comparison of C_{uc} of different TCP variants on emulated (with no error) and public cloud network

results indicate a near optimal throughput performance of both cloud networks with $C_{uc} \approx 0.7$, and also an overall agreement between both production cloud networks and emulated network, as shown in Fig. 2 for TCP CUBIC. These coefficients capture the overall throughput and concave-convex shape of throughput profiles. They are, however, less sensitive to the higher variation and lower throughput at a few RTTs of VPC connections, as expected since they are defined as integrals. The number of parallel flows continues to be a dominant factor in determining the optimization level across various loss conditions and TCP versions, particularly at higher RTTs.

VI. Conclusions and Future Work

Cloud infrastructures support dynamic provisioning of transport networks, which are possible alternatives to dedicated networks in a broad class of scenarios. We studied throughput performance of such networks over a public cloud environment using an emulated, dedicated testbed network with a similar range of RTTs and endpoint settings. We found that measurements and analytics of these two networks to be comparable in terms of throughput profiles and dynamics, despite latter being dedicated and emulated. The cloud network showed more variance in throughput values, most likely attributable to cross traffic on busy connections and hosts, which is not experienced in the emulated network. For all these networks, we found the most reliable way to increase throughput is to use multiple parallel flows. We also found that generally BBR had the best performance on this type of network across all RTTs. On the emulated network we also tested different levels of induced packet losses. We found that a high error rate has significant effect on throughput and this is only somewhat mitigated by using more parallel flows. Thus, the cloud networks are a more flexible, cost-effective alternative to dedicated network infrastructures in cases where their footprint is conducive.

References

[1] N. S. V. Rao, Q. Liu, S. Sen, Z. Liu, and R. Kettimuthu, “Measurements and analytics of wide-area file transfers over dedicated connections,” in *International Conference on Distributed Computing and Networking*, 2019.

[2] Y. Wu, S. Kumar, and S.-J. Park, “On transport protocol performance measurement over 10gbps high speed optical networks,” in *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, 2009, pp. 1–6, ISSN: 1095-2055.

[3] A. Esterhuizen and A. E. Krzesinski, “TCP congestion control comparison,” *Southern Africa Telecommunication Networks and Applications Conference*, p. 6, 2012.

[4] T. Lukaseder, L. Bradatsch, B. Erb, R. W. Van Der Heijden, and F. Kargl, “A comparison of TCP congestion control algorithms in 10g networks,” in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, 2016, pp. 706–714.

[5] J. Crichigno, Z. Csibi, E. Bou-Harb, and N. Ghani, “Impact of segment size and parallel streams on TCP BBR,” in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, 2019, pp. 1–5.

[6] B. Jaeger, D. Scholz, D. Raumer, F. Geyer, and G. Carle, “Reproducible measurements of TCP BBR congestion control,” vol. 144, pp. 31–43, 2019.

[7] N. S. V. Rao, Q. Liu, Z. Liu, R. Kettimuthu, and I. Foster, “Throughput analytics of data transfer infrastructures,” in *Testbeds and Research Infrastructures for the Development of Networks and Communities*, H. Gao, Y. Yin, X. Yang, and H. Miao, Eds. Springer International Publishing, 2019.

[8] Q. Liu and N. Rao, “On concavity and utilization analytics of wide-area network transport protocols,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2018, pp. 430–438.

[9] “CUBIC: a new TCP-friendly high-speed TCP variant,” vol. 42.

[10] D. Leith and R. Shorten, “H-TCP: TCP for high-speed and long-distance networks,” p. 16, 2004.

[11] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, “BBR: Congestion based congestion control,” *ACM Queue*, vol. 14, no. 5, Dec. 2016.

[12] M. Avriel, W. E. Diewert, S. Schaible, and I. Zang, *Generalized Concavity*. SIAM, 2010.