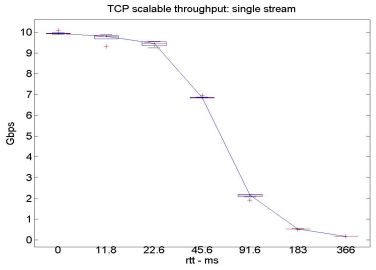
Sustained Wide-Area TCP Memory Transfers Over Dedicated Connections

Nageswara S. V. Rao*, Don Towsley†, Gayane Vardoyan†, Bradley W. Settlemyer§, Ian T. Foster‡, Raj Kettimuthu‡
*Oak Ridge National Laboratory, †University of Massachusetts, §Los Alamos National Laboratory, ‡Argonne National Laboratory

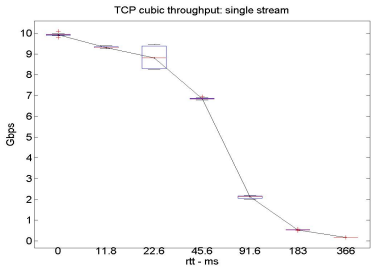
***Abstract*—Wide-area memory transfers between on-going computations and remote steering, analysis and visualization sites can be utilized in several high-performance computing scenarios. Dedicated network connections with high capacity, low loss rates and low competing traffic, are typically provisioned over existing infrastructures to support these transfers. To gain insights into such transfers, we collected throughput measurements for different versions of TCP between dedicated multi-core servers over emulated 10 Gbps connections with round trip times (rtt) in the range 0-366 ms. Existing TCP models and measurements over shared links are well-known to exhibit monotonically decreasing, convex throughput profiles as rtt is increased. In sharp contrast, our these measurements show two distinct regimes: a concave profile at lower rtts and a convex profile at higher rtts. We present analytical models that explain these regimes: (a) at lower rtt, rapid throughput increase due to slow-start leads to the concave profile, and (b) at higher rtt, TCP congestion avoidance phase with slower dynamics dominates. In both cases, however, we analytically show that throughput decreases with rtt, albeit at different rates, as confirmed by the measurements. These results provide practical TCP solutions to these transfers without additional hardware and software, unlike Infiniband and UDP solutions, respectively.**

I. Introduction

A number of High-Performance Computing (HPC) workflows require wide-area data transfers over dedicated networks in a variety of scenarios. Traditionally, such transfers involve file transfers between supercomputers and storage systems. Increasingly, however, recent workflows require memory transfers from on-going computations on supercomputers to remote analysis and visualizations sites, and also between computations coordinated over geographically separated supercomputer sites. To support such capabilities, network infrastructures (e.g. Department of Energy's ESnet) are being built to provide on-demand, dedicated connections with very low losses and no competing traffic. Also, end hosts are being equipped with multiple cores some of which can be dedicated for network tasks while others perform computations. In some sense, these scenarios represent a convergence of data transfer capabilities that have been traditionally carried out over short distances using InfiniBand (IB) and those over long-haul connections using Transmission Control Protocol (TCP). In view of long distances between the transfer sites, TCP is a natural candidate for such data transfers. However, sustaining high transfer rates for these tasks requires optimized TCP parameters that match dedicated connections in ways that appear fundamentally different from data transfers over traditional shared Internet environments.



(a) Scalable TCP

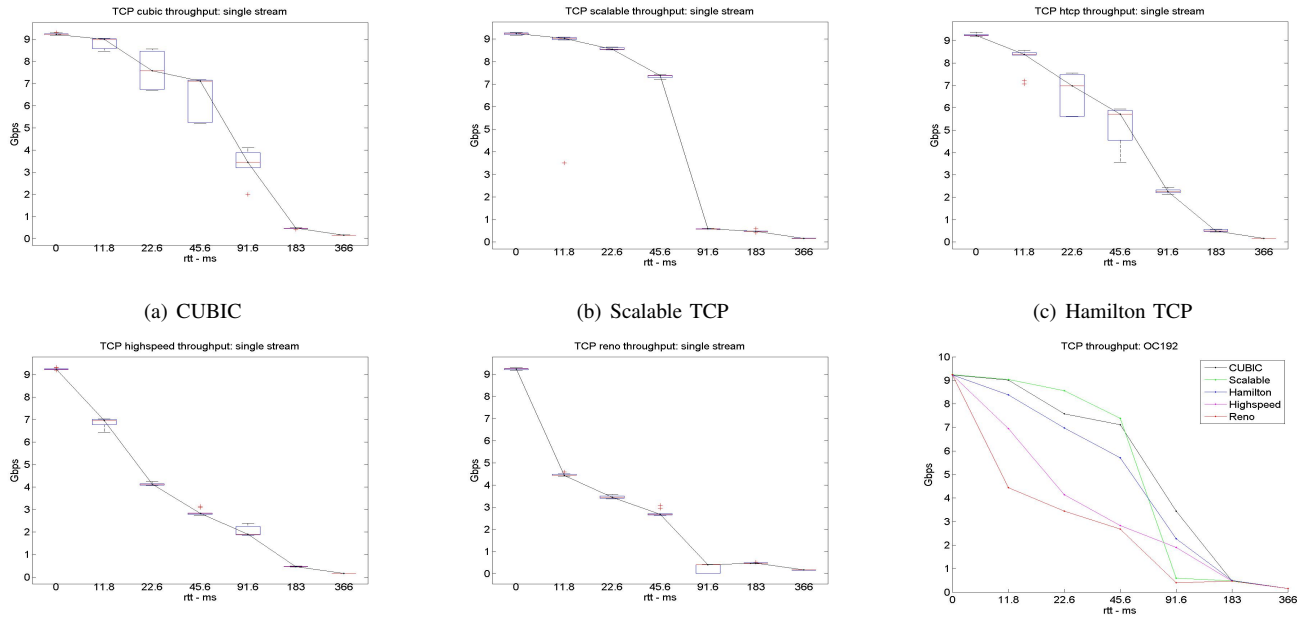


(b) CUBIC

Fig. 1. TCP throughput measurements over dedicated 10GigE connections.

A wide variety of TCP analytical models have been developed and experimental measurements have been collected over the past decades [2], [6], [10]. Typically, it is expected that TCP throughput decreases as a function of the round trip time (rtt). Furthermore, conventional TCP models based on different loss models lead to *convex* throughput profiles [9] that indicate a rather sharp drop as rtt is increased.

To gain insights into TCP optimizations needed for these transfers, we systematically collected throughput measurements using different TCP versions over dedicated connections emulated in hardware for a wide range rtts (0-355 ms). As shown in Figure 1 for Scalable TCP [3] and CUBIC [8], the throughput profile $\mathcal{T}(\tau)$ is *concave* for lower rrt τ, and it switches to *convex* for larger rtt. Indeed, this concave profile is very desirable since it represents less sharp throughput drop as rtt is increased; this is particularly important for 50-100 ms rtt range, which corresponds to distances between national-scale HPC facilities. This *dual-regime* throughput profile is in sharp contrast with typical convex profiles predicted by conventional TCP models. Indeed, such a dual-regime throughput profile observed over dedicated connections is not adequately explained by conventional TCP analytical models, since they focus more on lossy shared connections [4], [5], [9].



(a) CUBIC (b) Scalable TCP (c) Hamilton TCP

(d) Highspeed TCP (e) Reno (f) summary

Fig. 2. Throughput measurements collected between 32-core host systems over dedicated OC192 connections using five TCP methods.

In this short note, we present analytical results that explain the observed dual-regime TCP throughput profiles:

(a) at lower rtt, TCP throughput reaches the connection capacity quickly through the slow-start phase, and its fast exponential growth leads to the concave profile, and
(b) at higher rtt, TCP enters the congestion avoidance phase before reaching the connection capacity, and the relatively slower growth of throughput leads to the convex profile.

The boundary between these two regimes is specific to the TCP version, and is determined by which parameter, the delay-bandwidth product or slow-start parameter `ssthresh`, is crossed by the congestion window first. In both cases, however, we analytically show that the throughput decreases with rtt, albeit at different rates for various TCP versions, as confirmed by the measurements.

Our measurements combined with analytical results provide practical guidelines for utilizing TCP solutions for these specialized data transfers, which do not require additional hardware or software. Indeed, the concave profiles can be achieved by the congestion control modules available in Linux 2.6 and later distributions. These solutions require minimal development and modifications, and do not require special-purpose WAN accelerators needed for IB transfers [7] and additional software needed for User Datagram Transport (UDT) method [1]. In addition, our measurements provide useful practical insights into the concave regime: (a) for a significant rtt range, Scalable TCP provides higher throughput than CUBIC, which is the default in current Linux distributions, and (b) throughput deviations are more robust to variations in rtt compared to those predicted by conventional TCP models.

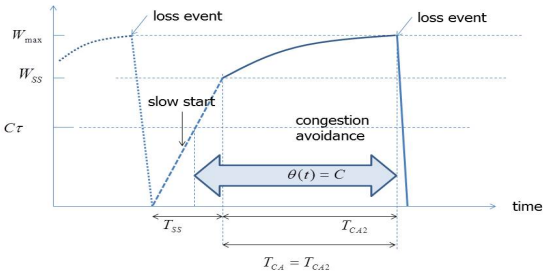
We describe throughput measurements and experimental configurations in Section II. We present analytical results in Section III-A, and present proofs of dual-regime profile and monotonicity in Sections III-B and III-C, respectively.

II. Throughput Measurements

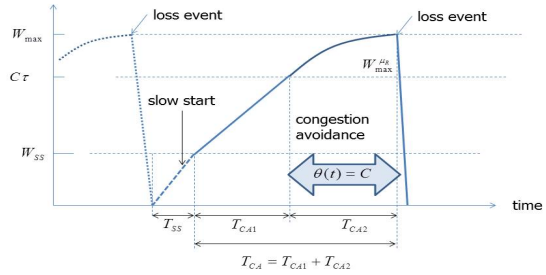
TCP memory-to-memory throughput measurements are collected using iperf-2 between two 48-core Linux host systems over emulated dedicated 10GigE connections (Figure 1). In this configuration, 10GigE NICs of host systems are directly connected to two ANUE emulator ports. The hardware emulator carries the physical packets between hosts, and delays them during the transit by a specified amount; in general, it provides more accurate real-time TCP dynamics compared to a simulator. We collected TCP throughput measurements for rtt τ=11.6, 22.6. 45.6, 91.5, 183 and 366 ms; the lower rtts match the physical connections in our testbed and are used to verify the measurements. Also, lower rtts represents US cross-country connections, for example, ones between DOE sites provisioned using OSCARS system, and higher rtts represent trans-continental connections. To mitigate configuration-specific artifacts, throughput measurements are also collected over more complicated OC192 ANUE emulated connections. In this case, 10GigE NICs of 32-core hosts are connected to a Force10 E300 switch that converts between 10GigE and SONET frames, and OC192 ANUE emulator is in turn connected to the WAN ports of e300. TCP throughput measurements are collected using using five TCP congestion control modules, namely, CUBIC, Scalable TCP, Hamilton TCP and Reno (all available with Linux 2.6 distribution). Each measurement is repeated 10 times, and the boxplots with one-σ interval and outliers, and median profiles are shown for the two configurations in Figures 1 and 2, respectively.

The overall dual-regime profiles observed for CUBIC and Scalable TCP over 10GigE configuration are also confirmed in the more complicated OC192 configuration. Furthermore, additional measurements in OC192 configuration showed a similar profile for Hamilton TCP, but showed only convex profiles for Reno and HighSpeed TCP, as shown in Figure 2. Thus, measurements provide practical information for the

selection of TCP modules to support concave profiles, and also provide relative comparison between them within specific rtt ranges as shown in Figure 2(f). Overall, however, throughput is a decreasing function of rtt, albeit at different rates, in all TCP versions. Measurements collected under external frame losses resulted in convex profiles in all versions but such profiles are not representative of the dedicated connections.



(a) lower rtt τ



(b) higher rtt τ

Fig. 3. Dynamics regions of TCP between two loss events.

III. TCP Throughput Analytical Models

TCP dynamics are characterized by the *congestion window* $w(t)$ and throughput $\theta(t)$, and average throughput $\mathcal{T}(\tau)$ during the data transfer. Its performance is determined by the slow-start threshold W_{SS} and the delay-bandwidth product of connection capacity C and rtt τ. The dynamics of TCP can be characterized by (at most) three regions in-between two consecutive loss events as shown in Figure 3:

(a) In the slow-start region, w increases exponentially for the duration T_{SS} until it reaches W_{SS}, and then TCP switches to the congestion avoidance mode. The average throughput in this region is given by $\Theta_{SS} = 1/T_{SS} \int_0^{T_{SS}} \theta(t)dt$.

(b) If $W_{SS} > C\tau$, TCP enters the congestion avoidance region CA2 immediately after the slow-start, and $w(t)$ is updated after each acknowledgment based on TCP version, e.g., $w \leftarrow w + a/w$, for Reno. The throughput remains constant $\theta(t) = \Theta_{CA2} = C$ for the duration T_{CA2}, which is also the rate at which acknowledgments are received.

(c) If $W_{SS} < C\tau$, TCP enters the congestion avoidance region CA1 immediately following the slow-start, wherein w is incremented somewhat slowly by an amount specified by the TCP variant, e.g., by 1 for each rtt τ for Reno,

For a fixed W_{SS}, such as `ssthresh` retained by Linux across multiple TCP sessions, increase in τ may result in moving from condition (b) to (c); this in turn results in a slower growth of $w(t)$ and lower average throughput $\mathcal{T}$.

A. TCP Model for Dedicated Connections

For dedicated connections, there are two basic cases:

(a) *Concave Regime:* For smaller rtt t, there is no CA1 region, and as $w(t)$ crosses W_{SS}, $\theta(t)$ switches from exponentially increasing to a constant value C (Figure 3(a)). This behavior leads to a concave profile for small rtt observed in measurements (Section III-B).

(b) *Convex Regime:* For larger rtt, $w(t)$ crosses W_{SS} before reaching $C\tau$, and its slower growth in region CA1 (Figure 3(b)) leads to a convex profile (Section III-C).

In these cases, occasional losses occur but have a limited effect on the average throughput, which is mainly determined by τ and W_{SS}. Such behavior is confirmed by tcpprobe traces that showed mostly monotonically increasing $w(t)$. Indeed, by maintaining high values of W_{SS}, the desired concave profile is achieved (by some TCP versions). Traditional TCP models, driven primarily by losses, lead to throughput profiles in the generic form $\hat{\mathcal{T}}(\tau) = a + b/\tau^c$, $c \geq 1$. They indicate convex profiles, and do not adequately account for the details of dedicated connections to capture the concave portions in the observed profiles.

B. Concave Regime

During slow start, $\theta(t)$ is doubled every rtt t, for the duration $T_C = \tau \log C$, during which the average throughput is $\Theta_C \approx \frac{2C}{\tau \log C}$. Thus, we have

$$\begin{aligned}\mathcal{T}(\tau) &= \Theta_C \frac{T_C}{T_{SS}+T_{CA}} + C\left[1 - \frac{T_C}{T_{SS}+T_{CA}}\right] \\ &= C\left[1 - \frac{(\tau \log C - 2)}{\tau \log W_{SS} + T_{CA}}\right].\end{aligned}$$

We now first show that $\mathcal{T}(\tau)$ decreases with τ. The condition $\mathcal{T}(\tau) < \mathcal{T}(\tau + \delta)$ is equivalent to

$$\begin{aligned}&(\tau \log C - 2)\left[(\tau+\delta)\log W_{SS} + T_{CA}\right] \\ &\quad < \left[(\tau+\delta)\log C - 2\right](\tau \log W_{SS} + T_{CA}).\end{aligned}$$

This condition in turn is equivalent to $C^{T_{CA}} > 2/W_{SS}^2$, which is satisfied for $W_{SS} > 1$ and $C > 1$ and $T_{CA} > 1$.

Now, we show the concave profile by considering the following condition: for $\tau_2 > \tau_1$, and $x \in [0, 1]$,

$$x\mathcal{T}(\tau_1) + (1-x)\mathcal{T}(\tau_2) < \mathcal{T}(x\tau_1 + (1-x)\tau_2).$$

By substituting the above formulae, we obtain

$$x\left[\frac{\tau_1 \log C - 2}{\tau_1 \log W_{SS} + T_{CA}} + \frac{\tau_1 \log C - 2}{(x\tau_1 + (1-x)\tau_2)\log W_{SS} + T_{CA}}\right]$$

$$< (1-x)\left[\frac{\tau_2 \log C - 2}{(x\tau_1 + (1-x)\tau_2)\log W_{SS} + T_{CA}} + \frac{\tau_2 \log C - 2}{\tau_1 \log W_{SS} + T_{CA}}\right].$$

Since $\tau_2 > \tau_1$, it suffices to show the concavity of $\left[1 - \frac{1}{\tau \log W_{SS}+T_{CA}}\right]$ or convexity of $\frac{1}{\tau \log W_{SS}+T_{CA}}$. This condition is shown by the convexity of function of the form $f(t) = \frac{1}{a\tau+b}$. To show latter, we note that the derivative $\frac{df}{d\tau} = \frac{-a}{(a\tau+b)^2}$ at $\tau = \tau_1$ is smaller than the slope of segment joining $(\tau_1, f(\tau_1))$ and $(\tau_2, f(\tau_2))$, which is given by $\frac{-a}{(a\tau_1+b)(a\tau_2+b)}$. Thus, in every neighborhood, $f(.)$ decreases faster than this linearized segment, and hence is convex.

C. Monotonicity of TCP Throughput Profile

In this section, we show that throughput $\mathcal{T}(\tau)$ decreases with τ when all three regions, namely slow-start, CA1 and CA2 are present. Based on the results of previous section it suffices to show that region CA1 leads to decreasing $\mathcal{T}(\tau)$; combined with such property of the slow-start region, it follows that overall average TCP throughput decreases with τ even though there are no losses. The growth of $w(t)$ during CA1 region, which depends on TCP version, directly impacts the throughput $\theta(t)$, unlike in CA2 region where $\theta(t) = C$. We restrict our derivation to a generic additive increase method considered in [9] in this section, and our approach is applicable to multiplicative and other increase methods. In CA1 region of congestion avoidance, the number of packets sent are

$$N_{CA1} = \frac{C\tau - W_{SS}}{2\tau} + \frac{W_{SS}T_{CA1}}{\tau}$$

and its duration is $T_{CA1} = C\tau^2 - W_{SS}\tau$. The average throughput in this region is $\Theta_{CA1}(\tau) = \frac{C}{2} + \frac{W_{SS}}{2\tau}$. Then $\mathcal{T}(\tau+\delta) = \frac{C}{2} + \frac{W_{SS}}{2(\tau+\delta)}$. Also, $T_{CA2}(\tau) = \tau_0 + k\tau$, where t_0 is the time before the first packet is sent, and $k\tau$ is time needed to infer a packet loss. Note that during T_{CA1}, $\theta(t)$ grows slower than during slow-start but $\theta(t) = C$ during T_{CA2} as in the previous case.

We consider

$$\mathcal{T}_{CA}(\tau) = \Theta_{CA1}(\tau)\frac{T_{CA1}}{T_{CA1}+T_{CA2}} + C\frac{T_{CA2}}{T_{CA1}+T_{CA2}}.$$

By using $\mathcal{T}_{CA}(\tau) = \Theta_{CA1}(\tau)f_{CA1}(\tau) + C(1 - f_{CA1}(\tau))$ and

$$\mathcal{T}_{CA}(\tau+\delta) = \Theta_{CA1}(\tau+\delta)f_{CA1}(\tau+\delta) + C(1 - f_{CA1}(\tau+\delta))$$

it suffices to show that

(i) $\Theta_{CA1}(\tau+\delta) < \Theta_{CA1}(\tau)$, and
(ii) $f_{CA1}(\tau+\delta) > f_{CA1}(\tau)$.

For the former, we note that

$$\Theta_{CA1}(\tau+\delta) = \frac{C}{2} + \frac{W_{SS}}{2(\tau+\delta)} = \Theta_{CA1}(\tau) - \frac{W_{SS}}{2}\frac{\delta C}{\tau(\tau+\delta)},$$

where the second term is positive. For the second part, we note that

$$f_{CA1}(\tau) = \frac{C\tau^2 - W_{SS}\tau}{C\tau^2 - W_{SS}\tau + \tau_0 + k\tau} = \frac{\alpha}{\alpha+\beta}.$$

Thus, it suffices to show that

$$\frac{\alpha+\delta\alpha}{\alpha+\delta\alpha+\alpha+\delta\alpha} - \frac{\alpha}{\alpha+\beta} > 0$$

or equivalently $\beta\delta\alpha - \alpha\delta\beta > 0$, which in turn evaluates to

$$\tau_0\left[C\left((\delta\tau)^2 + 2\tau\delta\tau\right) - W_{SS}\delta\tau\right] + k\tau\delta\tau\left[C\delta\tau + C\tau\right].$$

Both terms in the above expression are positive which shows the result.

Throughput $\mathcal{T}(\tau)$ decreases with τ, and the profile in this region is convex. Informally, the "slower" increase of $w(t)$ in CA1 region, as opposed to the exponential growth during slow-start, is not sufficient to sustain a concave profile. When both regions are present, their relative durations determine which profile will prevail: T_{CA1} becomes larger with rtt τ, and the convex profile dominates, as observed in our measurements.

IV. Conclusions

To study TCP versions and their parameters to support HPC transfers over dedicated connections, systematic measurements were collected using emulation devices. The observed dual-regime of throughput profiles necessitated analyses that are different from conventional TCP models developed for shared environments. We presented analytical results that highlight TCP regions specific to low-loss dedicated connections and established the concave profile for smaller rtt values. The combination of measurements and analytical models provided us practical guidelines to sustain high throughput by choosing TCP method such as CUBIC, Scalable TCP and Hamilton TCP. It would be of future interest to investigate methods such as IB and UDT for their ability to mitigate the sharp throughput drop at higher rtt inherent to all TCP versions. Also, it would be interesting to extend these results to more complex cases where data rates are limited by a combination of file systems, IO and host systems.

Acknowledgments

This work is funded by the RAMSES project, Office of Advanced Computing Research, U.S. Department of Energy.

References

[1] Y. Gu and R. L. Grossman. UDT: UDP-based data transfer for high-speed wide area networks, 2007.

[2] M. Hassan and R. Jain. *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*. Prentice Hall, 2004.

[3] T. Kelly. Scalable TCP: Improving performance in high speed wide area networks. *Computer Communication Review*, 33(2):83–91, 2003.

[4] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The mascroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), 1997.

[5] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, 2000.

[6] N. S. V. Rao, J. Gao, and L. O. Chua. On dynamics of transport protocols in wide-area internet connections. In L. Kocarev and G. Vattay, editors, *Complex Dynamics in Communication Networks*. Springer-Verlag Publishers, 2005.

[7] N. S. V. Rao, W. Yu, W. R. Wing, S. W. Poole, and J. S. Vetter. Wide-area performance profiling of 10gige and infiniband technologies. In *SC2008: Supercomputing Conference*, 2008.

[8] I. Rhee and L. Xu. Cubic: A new tcp-friendly high-speed tcp variant. In *Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks*, 2005.

[9] Y. Srikant and L. Ying. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2014.

[10] T. Yee, D. Leith, and R. Shorten. Experimental evaluation of high-speed congestion control protocols. *Transactions on Networking*, 15(5):1109–1122, 2007.