



Overcoming the Limits of TCP on High-Speed WANs

Executive Summary

Today's IT organizations rely heavily on wide-area networks (WANs) for application performance and data protection. Key to meeting these goals is the behavior of TCP, the underlying protocol used by most business applications, which must be able to utilize the WAN link completely. It is well known, however, that the effectiveness of TCP drops dramatically as the distance between endpoints increases and/or the quality of the WAN link drops.

This white paper reviews the causes of poor TCP performance on high-speed WAN links, and explains why adding more bandwidth cannot solve the issue. It also looks at the data transport requirements of the long-distance replication, periodic offsite backups, and data migrations that are critical for today's data protection strategies.

The second portion of the paper examines the Infineta Data Mobility Switch (DMS). This technology is the only solution available that can accelerate TCP-based connections at speeds of 10 Gbps, regardless of distance. To understand how, we analyze Infineta's approach to TCP congestion control and traffic management.

The TCP Bottleneck

Since 1983 when it became the standard protocol for ARPANET, TCP has remained the most extensively used transport protocol on the Internet. No other protocol has proven more robust in the face prodigious change. Over the past 30 years, the Internet has grown by six or seven orders of magnitude, from a few thousand networked computers to billions, and all the while TCP has prevented it from congestive collapse. Even now, with IPv6 promising support for an unfathomable 340 undecillion (10^{36}) endpoints, nothing is poised to replace TCP.

The only significant weakness of TCP is that it underperforms when sending large amounts of data across high-speed WANs (also known as Long Fat Networks, or LFNs). It is a well-known characteristic of TCP that as the Bandwidth Delay Product (BDP)¹ grows, end-to-end throughput decreases and the LFN becomes a serious bottleneck. By the early to mid 2000s, the issue had become a major pain-point. Organizations, faced with mounting pressure to protect their digital assets through off-site storage, and to transport bulk data between locations, found themselves constrained with regards to data mobility. Adding bandwidth, data centers, or even sacrificing protection, did not solve the problem.

¹ 1988. Van Jacobson, RFC 1072. BDP = (bandwidth in bps) X (round-trip time in seconds). You can think of BDP as the total number of "bits in flight" between two endpoints, i.e., the total amount of unacknowledged bits that TCP will allow on the link.

Putting throughput into perspective

To illustrate the concept of BDP and its role in throughput, let's use the example of a 2.5 GB file being transferred from New York to Chicago using FTP over a dedicated 2.5 Gbps WAN. Assume the round trip time between the two cities is 30 ms, and that the file is being sent using TCP Reno on the sender side stack (TCP Reno is the most prevalent "flavor" of TCP used on the Internet today).

Given that two-and-a-half gigabytes is really twenty gigabits, one might expect that it would take eight seconds to transfer the file. In reality, however, the transfer would take about 19.5 minutes. Because of the latency between sites, and because TCP is a connection-oriented protocol, the theoretical maximum throughput for the connection is 17.1 Mbps.² Considering the fact that "large" file transfers, and data operations such as replication, are now routinely measured in terabytes—even petabytes—these low throughput rates are nowhere near what businesses require to meet their performance needs. Even if one were to double the link bandwidth to 5 Gbps, throughput for this FTP transfer would not increase.

Requirements for Today's LFNs

There is no question that businesses today are absolutely committed to digitization. They have billions of dollars of value expressed only in digital assets, and many are shaped by their need to collect, store, and transfer data digitally. It is not uncommon to have petabytes of storage, and storage requirements are growing at an estimated rate of more than 40% a year.³ These organizations require data mobility to keep pace with their changing environments. They demand distributed compute platforms, private clouds, and new data center efficiencies. At the same time, Business Continuity and Disaster Recovery (BC/DR) plans are fundamental to investor and customer confidence. BC/CR typically requires that critical digital assets are available in multiple off-site locations to prevent loss in the event of disruption.

Never before has the need for high-performance WAN transport been greater or more crucial, but previous WAN optimization solutions have not been able to provide the kind of performance required by businesses now.

Legacy WAN optimization and the "first byte"

Early attempts to address TCP performance on the WAN resulted in solutions targeted at high-impact points in the network. These included application delivery between branch offices and headquarter and speeding the delivery of Web pages. The focus was on addressing the efficiency with which the client received the **first byte** of data in order to reduce user wait-times and thus provide a better user experience. These solutions provided WAN optimization by addressing the "flaws" that emerged when chatty LAN protocols such as CIFS started to be used on WAN in order to extend the range of LAN applications. The main technique employed was to reduce the overhead required to set up and maintain connections between the client (branch) and the application (headquarters). Because the WANs used tended to be low-speed, these solutions were able to provide results without having to tackle the root causes of insufficient TCP performance.

² Throughput = TCP window size / RTT. The default window size used in TCP Reno stacks is 64,000 bytes, or 512,000 bits. Latency for the 700-mile link between New York and Chicago is 15 ms latency, for a 0.03 round trip time (RTT). $512,000/0.03=17.06$ Mbps.

³ [IDC predicts](#) (page 14) that in 2011, the "digital universe"—the amount of information and content created and stored digitally—will grow to 1.8 ZB (zetabytes!), up 47% from 2010, and will exceed 7 ZB by 2015. One ZB = 10^{21} bytes, or 1 billion terabytes.

Root Causes of WAN Under-Utilization

TCP is a connection-oriented protocol, which means that there is a lot of back-and-forth between the sender and receiver in order to set up the connection, transmit the data, acknowledge receipt (ACK), and perform other control functions necessary to ensure reliable transfer. TCP was designed to protect the network from collapse, even when there are trillions of connections on the Internet at the same time.⁴ TCP achieves this by using a congestion-avoidance technique called windowing, which limits the amount of unacknowledged data a sender can send without receiving an acknowledgement from the recipient. Whenever the maximum window size is reached on a connection, the sender must stop transmitting new data until it receives an ACK for the data sent.

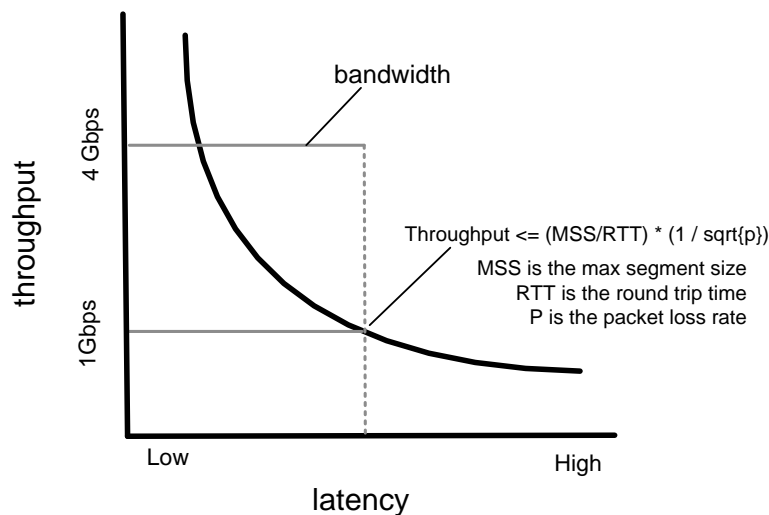


Figure 1. TCP throughput is largely determined by window (segment) size, latency (RTT), and loss (recovery behavior).

Regardless of how important it may be, congestion control is central to the inability of TCP to fully utilize bandwidth on LFNs. Because it must guarantee reliability, TCP transfers data in windows and practices fair play during error recovery. This works well across short distances because the RTT is low (RTT, or Round Trip Time, is the time it takes to send a packet across the WAN, plus the time it takes to send the ACK back to the sender). As RTT grows, so to does the amount of time spent waiting to receive ACKs, during which no new data can be sent.

Effects of window size and latency on throughput

When setting up a connection between endpoints, TCP starts by identifying the maximum amount of data the receiving stack can handle at a time. This is called the “receive window” (r_{wnd}). The sender maintains a variable window to track how much data it can send at a time. Known as the congestion window (c_{wnd}), TCP constantly adjusts this metric to improve throughput for a given connection. The c_{wnd} size is incremented by some factor after successful delivery (confirmed via ACKs) of data to the receiver. In the case of TCP Reno, the congestion window is increased by one packet for every successful

⁴ Originally, TCP did not have congestion avoidance. Prior to its introduction in TCP Tahoe, networks were subject to a condition known as “congestive collapse,” in which all traffic would come to a standstill.

round-trip. If an expected ACK is not received, however, TCP cuts the window size by some percentage (50% in the case of TCP Reno) to back off from contributing to congestion. This act of scaling back is known as fair play because it gives congested networks a chance to recover, and it is the single-most important attribute of TCP because it prevents congestive collapse of the network.

To see how this works, let's say that a TCP connection has been running for some time, and the `cwnd` has reached 500 packets. Then the sender detects a packet loss, and it reacts by cutting the `cwnd` in half, to 250 packets. For the application to return to the performance level it was experiencing before the loss, the sender and receiver will need to complete another 250 round trips (in the case of TCP Reno) to rebuild the `cwnd` to 500 packets. Until that time, throughput will be reduced and application performance will suffer. Packet loss can always be expected to occur, so both it and TCP recovery behavior play a significant role in contributing to poor long-term link utilization and sub-optimal application performance.

Addressing the "Last Byte"

In contrast to the first crop of WAN optimization solutions that addressed user experience by focusing on the "first byte," subsequent efforts have focused on completing large data transfers across long distances as quickly as possible, i.e., delivering the **last byte**. Starting in the mid 2000s, a number of researchers began experimenting with TCP modification in order to improve throughput over the LFN. The result has been a series of proposals leading to TCP variants, most of which involve some form of increasing the TCP window size and/or modifying the congestion control algorithm.

However, if there is one weakness shared by all the TCP variants, it is that they function on a per-connection basis, in isolation from the other TCP connections on the link. As conditions on the link change, for example as flows burst with activity or drop-back in response to packet loss, each connection reacts individually to reach a new balance. The time this takes can be thought of as "slack," and when applied to the link as a whole, the slack from all these individual adjustments combines to produce an overall sluggish effect. No matter how efficient a given connection may be, the link as a whole will always under-perform because there is no mechanism for overall visibility that can take up slack as it appears.

Breaking the TCP Bottleneck

Clearly filling a WAN with TCP traffic is not a trivial task, especially if it is an LFN and if the number of connections is limited. Until recently, no "silver bullet" had emerged that could fulfill the responsibilities of TCP while providing high throughput across distances.

To provide businesses with the ability to minimize the effects of latency on TCP throughput and enable full WAN bandwidth utilization, Infineta Systems has developed the Data Mobility Switch (DMS) to accelerate high-capacity WAN links. The DMS is designed for latency-sensitive traffic and can deliver sustained throughputs of 10 Gbps. It works across any distance, with any number of connections, and on all WAN links.

The DMS takes a fresh approach to the long-standing issue of poor link utilization with an innovative TCP implementation called the Velocity Transport Engine (VTE). The VTE technology builds on the time-tested safeguards and proven features of traditional TCP, but then goes further to provide relativistic connection management and aggressive flow control for designated connections.

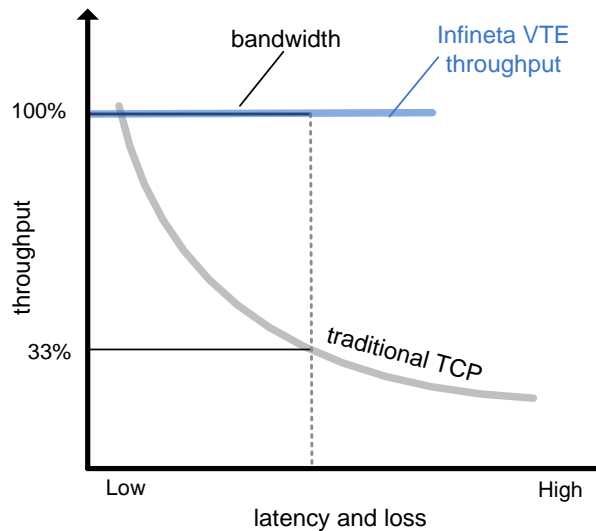


Figure 2. The DMS overcomes the effect of latency on throughput.

Responsive window scaling removes all “slack”

The DMS is typically deployed at strategic points in the data center network where it can accelerate a large number of TCP connections. By making the `cwnd` size decisions for all connections it is accelerating, VTE ensures a fully utilized link regardless of the number of connections established and actively sending traffic. Each VTE instance continuously updates its view of the available WAN bandwidth by tracking active connections, usable bandwidth, and the speed at which the remote endpoints can receive traffic.

As noted, while individual flows can grow aggressively under the high-speed TCP variants, total end-to-end throughput suffers from the lack of central oversight, because individual flows react to asymmetric information (packet loss). VTE on the other hand, always knows how much bandwidth is usable at any given instant, and so proactively distributes bandwidth across the active connections, instead of needlessly reacting to lost packets by reducing throughput. This also allows VTE to continue to be aggressive in how it utilizes available bandwidth without worrying about causing performance-impacting congestion events. Default TCP stacks are unable to scale `cwnd` over 64KB, which leads to very low throughput over LFNs. These stacks may also not support selective ACKs and other TCP extensions proposed in RFC 1323.⁵ VTE is able to apply all necessary extensions and keep each connection fully optimized across any WAN without the endpoint requiring any changes.

⁵ V. Jacobson, R. Braden, D. Borman. “TCP Extensions for High Performance.” <http://www.ietf.org/rfc/rfc1323.txt>

Rapid start and recovery

In addition to optimizing congestion windows over LFNs, VTE uses proprietary techniques that rapidly scale and maintain $cwnd$ at optimal levels to deliver immediate performance. VTE also takes a unique approach to “slow start,”⁶ whereby $cwnd$ can be adjusted based on available bandwidth instead of grown gradually from a low starting point. These techniques allow VTE to keep an LFN fully utilized even when connection counts change too frequently for individual TCP stacks to react.

As seen in Figure 2, traditional TCP (the red graph line) starts gradually and reacts dramatically to congestion events such as packet loss. The long-term effect of this reactive behavior is that traditional TCP underperforms on the average, and this under-performance is only exacerbated over LFNs. Compare this to the behavior of VTE (the blue graph line), which is able to start immediately, based on the bandwidth available, and does not sacrifice performance by tentatively rebuilding throughput.

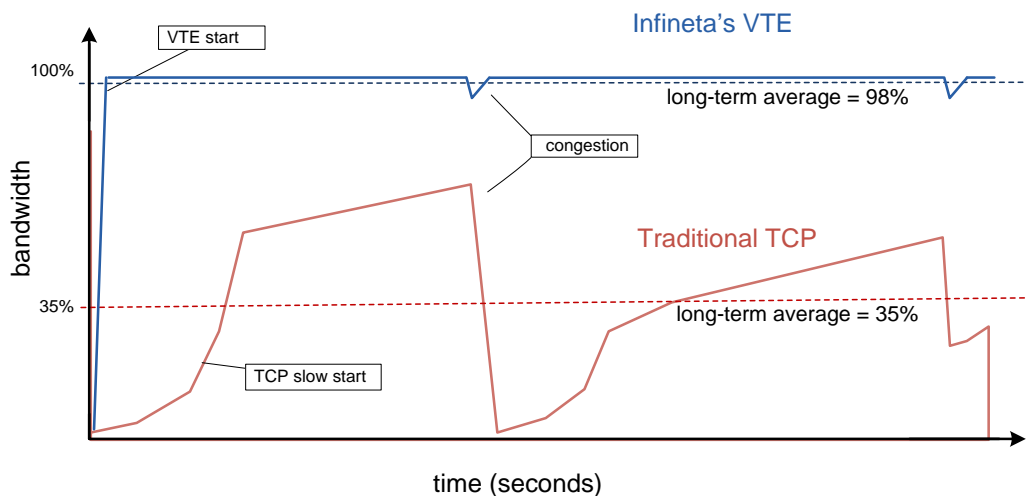


Figure 3. VTE fully utilizes LFNs, unlike traditional TCP.

Handling packet loss and packets out-of-order

Loss on LFNs tends to be “bursty” and not uniform. In other words, LFN loss events typically affect a series of packets rather than just one, so whenever loss does occur it is rarely followed by a similar event in the same time frame.

Infineta specifically developed a “fast retransmit” algorithm for VTE to address bursty loss events. It directly readjusts throughput for all effected connections, minimizing the impact of packet loss on the sender. This is an “optimistic” approach and is ideally suited for the characteristics of LFNs. In contrast, some of the legacy WAN optimization solutions rely on a “pessimistic” approach to loss that requires adding parity packets to the data streams to correct for potential loss. The problem here is not only that such an approach adds significant overhead to each connection (as much as 10% of the WAN), but that it

⁶ See “TCP Congestion Control,” 3.1 Slow Start and Congestion Avoidance, and Section 3.2 Fast Retransmit/Fast Recovery. <http://www.ietf.org/rfc/rfc2581.txt>

is only effective with LAN-like packet loss, i.e., one or two packets are dropped, and the pattern occurs with some regularity. The method is not effective for the “bursty” loss conditions that prevail on LFNs.

One other point of architecture which explains how the VTE is able to sustain high TCP throughput despite high RTTs is that it retains a copy of each packet that is sent across the WAN (once the packet has been delivered and verified at the other end of the connection, these buffered packets are flushed). If packet loss occurs, VTE retransmits the packet from its buffer instead relaying the request to the endpoint, which is especially important because recovering from packet loss can cause the endpoints to slow down significantly. By shielding the endpoints from loss, VTE enables them to always perform as aggressively as the application demands. Temporarily buffering packets also makes it easy for VTE to address out-of-order packet delivery, another relatively frequent occurrence on LFNs. Normally when packets arrive out of order, the endpoint interprets this as packet loss and must signal for a resend, a costly process in terms of retransmit time and TCP recovery behavior. VTE, however, tracks the correct sequence for in-flight packets, so it is able to differentiate between out-of-order deliveries and true loss events. Instead of incurring the cost of a packet loss, VTE simply reorders the packets before delivering them to the endpoint, a much more efficient alternative.

From Business Bottleneck to Business Enabler

LFNs have traditionally been considered a bottleneck, something to work around by constantly tweaking applications, reducing workloads, and sacrificing on Service Level Agreement (SLA) commitments. The Infineta VTE, however, transforms the LFN from a pain-point requiring frequent bandwidth upgrades to an agent for business growth. Applications that previously performed below capacity can scale up to their full potential. The task of moving around terabytes and petabytes of data becomes routine, and organizations can finally realized their investment in high-capacity WAN infrastructure and bandwidth.

Once the transformation from bottleneck to enabler occurs and an organization catches up to its WAN traffic, it is not uncommon to see IT’s perspective on the WAN begin to change. The LFN becomes a force multiplier, giving IT the leeway to start thinking more strategically about how to make the organization’s data an “inventoried asset” instead of putting out fires. Business goals that had been on the back burner can be addressed because data mobility, distributed processing, and virtualization are no longer limited by the old LFN. And at the same time, different lines of business within the organization start thinking strategically about their digital assets, and how to use them to create new opportunities for innovation and productivity.

About Infineta Systems

Based in San Jose, California, Infineta Systems is a privately-held provider of Hyper-scale WAN optimization systems. The company's patent-pending Velocity Dedupe Engine™ delivers unprecedented levels of throughput, scalability and bandwidth capacity to support critical machine-scale workflows across the data center interconnect. Its flagship product, the Infineta Data Mobility Switch, accelerates multi-gigabit BC/DR (business continuity/disaster recovery), cross-site virtualization, and Big Data traffic. The company is backed by Rembrandt Venture Partners, Alloy Ventures and North Bridge Venture Partners. For more information, visit www.infineta.com.

Infineta, Infineta Systems, Data Mobility Switch, and Velocity Dedupe Engine are trademarks or registered trademarks of Infineta Systems, Inc., in the U.S. All other product and company names herein may be trademarks of their respective owners.

Infineta Systems

2870 Zanker Road, Suite 200
San Jose, CA 95134

Phone: (408) 514-6600

Sales / Customer Support: (866) 635-4049

Fax: (408) 514-6650

General inquiries: info@infineta.com

Sales inquiries: sales@infineta.com

©2011 Infineta Systems, Inc. All rights reserved. No portion of the Documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written permission of Infineta.

Infineta disclaims all responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in the Documentation. Infineta reserves the right, but has no obligation, to modify, update, or otherwise change the Documentation from time to time.