Improving Effective WAN Throughput for Large Data Flows

By Peter Sevcik and Rebecca Wetzel

November 2008

When you buy a broadband Wide Area Network (WAN) you want to put the entire bandwidth capacity to work for you. But technical and physical constraints often stack the cards against you to prevent efficient use of bandwidth for large data transfers. To determine how effectively you use WAN capacity for large data transfers you need to understand how much available capacity you actually put to work. The higher your effective throughput, the more efficiently you use your WAN capacity.

This NetForecast report describes challenges preventing efficient bandwidth usage, and it identifies technical solutions that mitigate those challenges.

What Is Effective Throughput?

When you purchase a high-speed WAN circuit you buy access to all the available bandwidth on that circuit. To understand how much of that capacity you actually use you first need to know your *effective throughput*, which is *the number of bits per second successfully delivered from source to destination for an individual data flow*.

Once you know your effective throughput, you need to analyze it relative to the total available bandwidth by calculating the effective throughput ratio. This entails dividing effective throughput by total available bandwidth. The *effective throughput ratio tells you how much of the total available bandwidth your single flow uses*.

The effective throughput ratio for a single flow is highest near the data source and decreases as distance increases. It is orthogonal to bandwidth utilization, which is a measure of the capacity consumed by many flows. Unlike the effective throughput ratio, bandwidth utilization typically starts low and increases with the number of simultaneous users on the path.

The effective throughput ratio (ETR) communicates how efficiently bandwidth is being used by a single flow. The higher your effective throughput relative to available bandwidth, the more efficiently you are putting your circuit's bandwidth capacity to work.

Why Is Effective WAN Throughput Often Poor?

Three factors conspire to lower effective WAN throughput when transferring data over high-speed WAN links: distance, TCP window size, and packet loss. It may surprise you to learn that more bandwidth does not necessarily mean higher effective WAN throughput. That can be frustrating given the amount of money spent on WAN bandwidth.

Distance, TCP window size, and packet loss make effective throughput lower than the bandwidth of the slowest link along the path, which is usually an access link. The reason for this hinges on the Automatic Repeat Request (ARQ) mechanism within TCP.

ARQ uses a sliding window to enable the sender to transmit multiple packets before waiting for an acknowledgement from the receiver. The idea behind the sliding window is to move more data at a time over the network to minimize network idle time. (For a detailed description of the ARQ process, see Appendix A.)

NetForecast Report 5095

The effective throughput for a single flow traversing a WAN is calculated using the following formula:

$$\mathrm{ET} = \min\left[\left(\mathrm{B}\min\left(1, \frac{8\mathrm{W}}{\mathrm{BD}}\right)\left(1-\sqrt{\mathrm{L}}\right)\right), \left(\frac{8\mathrm{MSS}}{\mathrm{D}\sqrt{\mathrm{L}}}\right)\right]$$

Where:
ET= Effective Throughput (bits/sec)
B = Bandwidth (slowest link) (bits/sec)
W = TCP Window size (bytes)
D = Round trip delay (RTT) (sec)
L = Packet loss (>0) (fraction)
MSS = Maximum Segment Size (1460 bytes)

The effective throughput ratio (ETR) is then calculated as follows:

$$\mathrm{ETR} = \frac{\mathrm{ET}}{\mathrm{B}}$$

Where:
ETR = Effective Throughput Ration (fraction)
ET = Effective Throughput (bits/sec)
B = Bandwidth (slowest link) (bits/sec)

Bandwidth

The bandwidth value in the effective throughput formula is for the slowest access line or trunk circuit along the network path. This value typically maps to the bits per second of a T1, MPLS, DS3, OC3, GigE link, etc.

Delay

Round trip time (also known as delay) can have a tremendous range. This is because round trip is distance related, and WAN distances can sometimes be as great as 18,000 miles—two thirds of the way around the world. We model delay (time) as a function of the distance between the two TCP endpoints and show it as distance (miles) in this report.

TCP Window Size

The TCP window size range in most operating systems is 8K to 64K bytes, and most corporate servers support 64KB. The maximum window size in the original TCP protocol specification is 65,535 Bytes—a number set in the 1970s when T1 circuits were the fastest available option. However, RFC 1323 describes a method for supporting larger receive window sizes by allowing TCP to negotiate a scaling factor to achieve a window size of up to 1GB.

While the commonly used 64KB TCP window size works well to synchronize data transmission with T1 pipe capacity and feed data into the circuit with negligible idle time waiting for an ACK to return, it does not work well for higher capacity WAN connections. As circuit capacity increases, the time it takes to transmit the return ACK becomes increasingly important because the circuit remains idle while the source TCP implementation waits for the ACK before it can send more data.

A faster circuit puts a window's worth of data in flight faster and then must wait before sending more, so the percentage of idle time increases compared to a slower circuit. The larger the circuit, the more dramatically the ACK wait time lowers WAN effectiveness because no data can be transmitted while both the data and the ACK are in flight.

We modeled the effect of this window size/data synchronization problem and plotted the results in Figure 2. Given a 64KB TCP window size, you can see that a very large file transfer happens extremely efficiently across a 1.5Mbps T1 connection. But as circuit speeds increase to 10 Mbps through to 1,000Mbps, the effective throughput ratio degrades dramatically. The effective throughput ratio for a 1,000Mbps circuit even after a few hops across a campus is a mere 0.5—and it plunges to almost zero 1,000 miles away from the data center.

Figure 1 shows that if a constraining circuit is less than 1Mbps, upgrading to a T1 circuit will not harm your effective throughput ratio. But upgrading from a T1 to a higher-bandwidth circuit will degrade your effective throughput ratio substantially as you move further from the data source—and the more bandwidth you add to the circuit, the worse your effective throughput ratio will be.

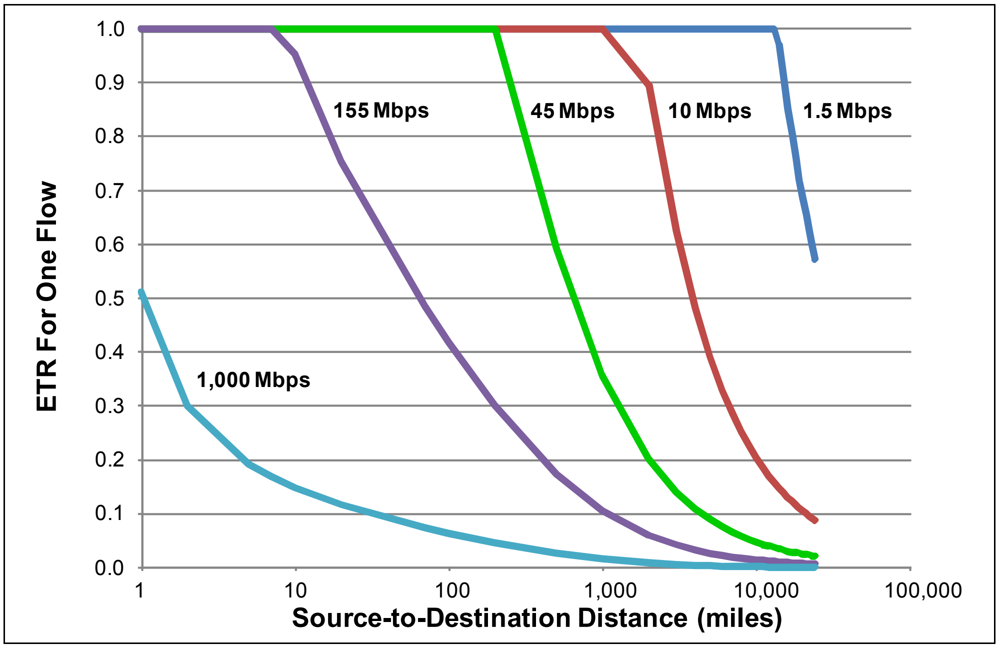

Figure 1 – Effective Throughput Ratios for One Flow - 64KB TCP Window

Packet Loss

Packet loss is cumulative as the data traverses multiple hops. Packet loss in private networks or top-tier ISPs generally falls between 0.1 and 1 percent. In parts of the developing world, however, packet loss is often 2 to 4 percent, and on highly congested links packet loss of 3 to 5 percent is not unusual.

Packet loss degrades effective throughput—and the greater the packet loss, the worse the degradation. Figure 2 shows how packet loss impairs the effective throughput ratio for the 45Mbps circuit shown on the previous chart (see the green line in Figure 1). Given 0.1 percent packet loss, a user 100 miles from the server would experience an effective throughput ratio decline of only 0.03 to 0.97. But when packet loss reaches 1 percent, the effective throughput ratio for the same user dives to an intolerable 0.32, and it declines to 0.18 when packet loss reaches 3 percent.

The same loss-induced degradation pattern also applies to the other line rates. Note that even without loss the 1Gbps circuit has a very poor effective throughput profile, and the slightest packet loss renders 1Gbps and other high bit rate circuits essentially useless.

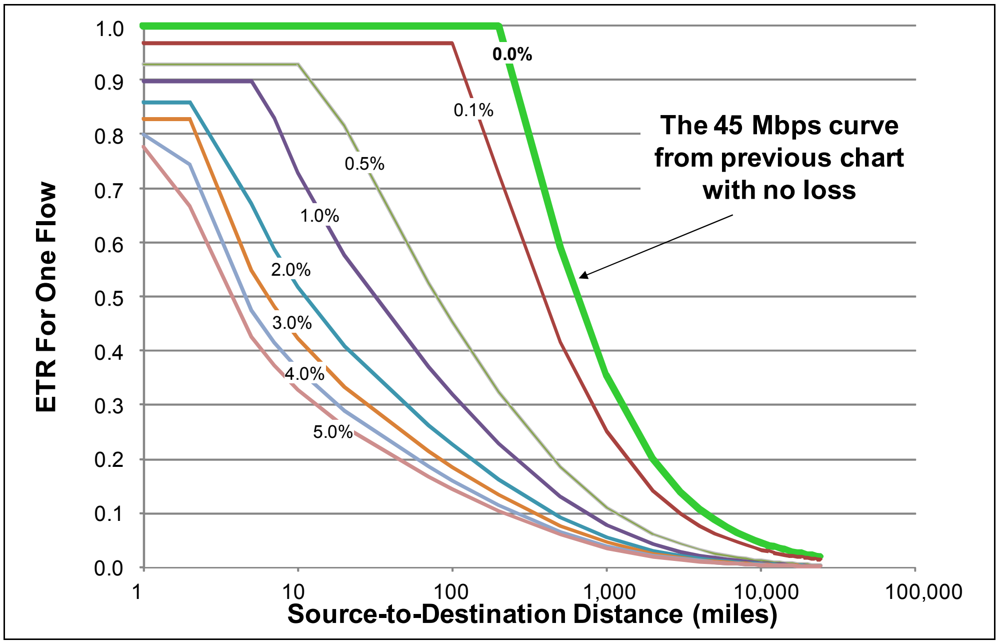

Figure 2 – Effective Throughput Ratios for One Flow with Packet Loss

How to Improve Effective Throughput

Although window size and packet loss prevent efficient bandwidth usage for large file transfers over broadband circuits, the good news is there are ways to mitigate both problems. As figure 4 shows, you can apply TCP optimization to increase window size, and you can use Forward Error Correction (FEC) and Packet Order Correction (POC) to lower packet loss. Here's how these performance-enhancing techniques work to improve effective throughput.

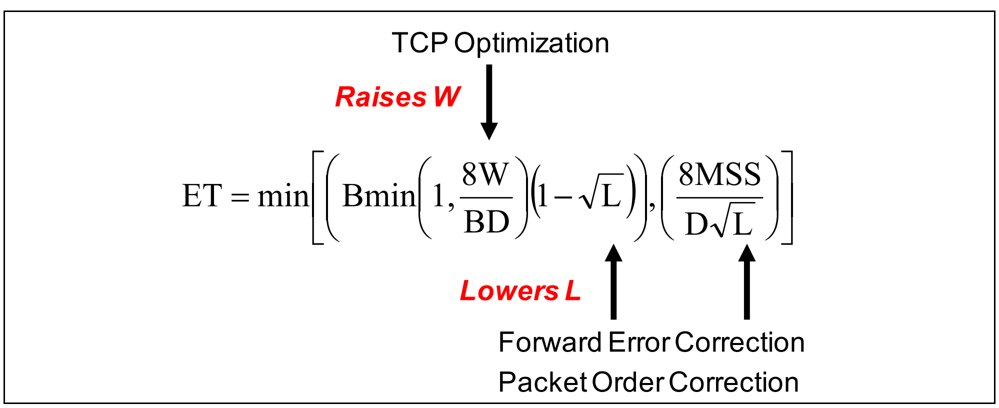

Figure 3 – Mechanisms for Improving Effective Throughput

TCP Optimization

TCP Optimization can include a variety of actions such as sending pre-emptive data receipt acknowledgements that maintain high throughput to speed data from the source, and ramping up the TCP transmission rate more quickly by bypassing TCP's 'slow start' function. TCP optimization also uses a selective acknowledgement (SACK) feature that retransmits only bytes lost rather than returning to the last continuously received data, and it increases TCP window size, which puts more data in flight on long latency paths.

Figure 4 shows how performance curves improve when TCP window size is increased to 256K Bytes. When you compare the new curves to those for a 64K Byte window shown in Figure 2, you can see that effective throughput ratios increase dramatically.

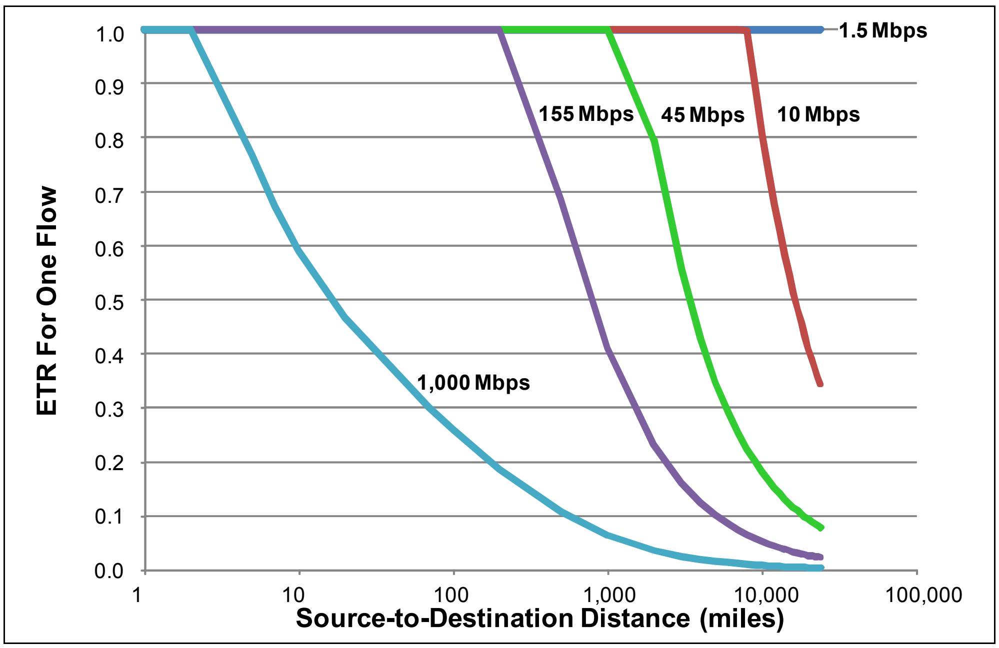

Figure 4 – Effective Throughput Ratios after Applying TCP Optimization

Figure 5 illustrates that a user located 2,000 miles from the server and connected via a 45Mbps circuit will experience a four times better effective throughput ratio with TCP optimization in place. Even a user located five miles from the server and connected via a 1,000 Mbps circuit will experience a four times better throughput ratio using TCP optimization. The four times improvement continues as distances increase from these examples.

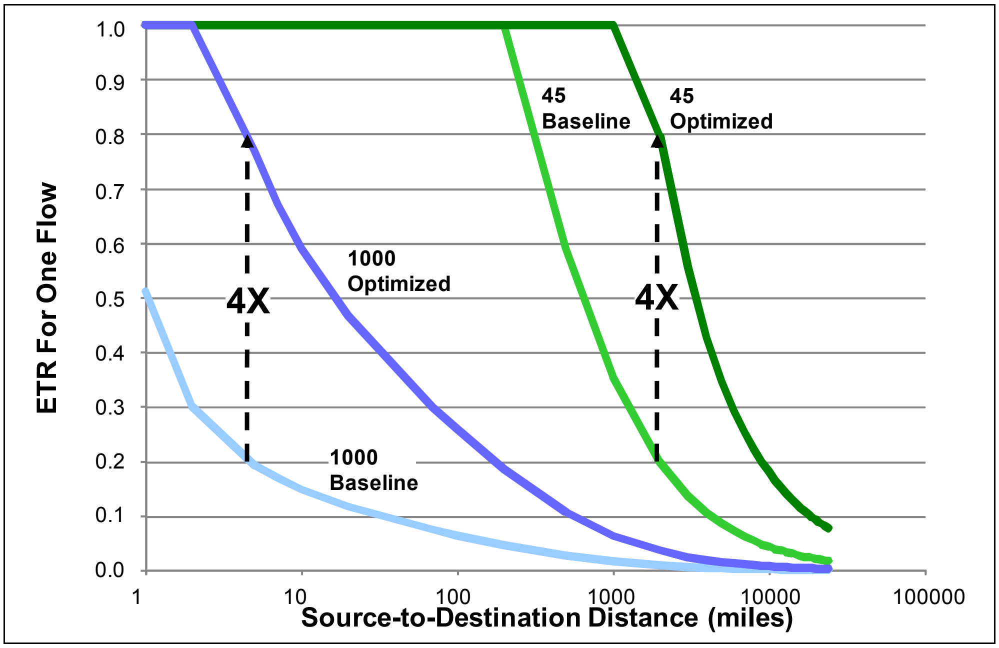

Figure 5 – Before and After TCP Optimization

> **Understanding What "X" Factors Really Mean**
>
> Although it sounds impressive, a performance "improvement" of 10x or even 100x has limited real-world value. A 10x improvement really means that 10 is now 1—a difference of 9. So the improvement is 9 divided by 10, which is 0.90 or 90%.
>
> As a buyer, it is useful to know that a range of big "X" claims are actually very close in true performance. For example, the difference between a 100x and a 10x claim may sound huge, but your users are unlikely to notice the actual performance difference between 99% and 90%. And any claim greater than 100x applies to the last 1% of capacity before the effective throughput ratio reaches full bandwidth capacity. How much are you willing to pay for that?

Forward Error Correction and Packet Order Correction

Forward error correction fixes errors in real time as data is received, avoiding the need to retransmit data when packets are lost. Although forward error correction adds overhead, the benefits make the tradeoff acceptable for underutilized high-capacity WAN links. Some WAN optimization solutions minimize overhead by dynamically matching forward error correction levels to loss levels.

Despite carrier assurances, MPLS and IP VPN links are not immune to packet loss. Carriers' QoS techniques often start dropping packets at 80 percent bandwidth utilization. Furthermore, other QoS mechanisms may prioritize traffic flows higher than the large file transfer.

MPLS and IP VPN environments also routinely suffer from out-of-order packet delivery. TCP identifies more than three packets received out of order as packet loss and calls for packet retransmissions and smaller TCP window size. This response can be particularly vexing when trying to keep many bytes in flight using a large window size. Packet order correction properly sequences out-of-order packets on the fly, thus avoiding retransmissions.

Mitigating loss using forward error and packet order correction techniques shifts effective throughput ratio curves toward the 'no loss' curve. As our model illustrates (see Figure 6), improving loss from 2 percent to 0.1 percent results in a 4x improvement for a user at 200 miles from the server.

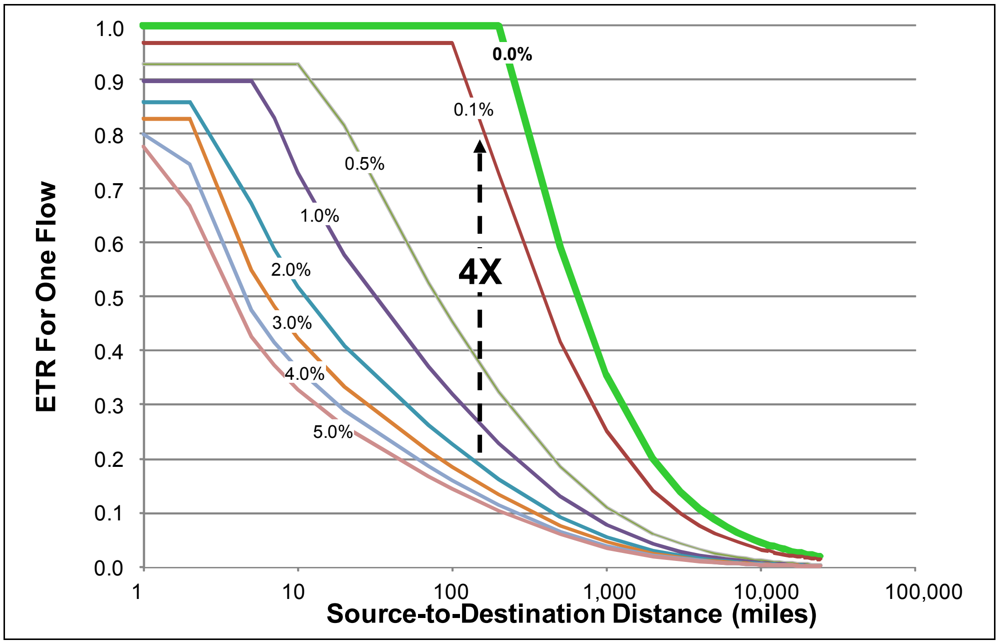

Figure 7 – Reducing Loss Shifts Effective Throughput Ratios Up (45 Mbps)

Conclusions

If efficiently transferring large amounts of data over high-capacity WANs is important to your business, chances are that you are not getting the effective throughput you need, and upgrading your bandwidth will not help. Instead, you need WAN optimization solutions that address the problems that degrade effective throughput. These include TCP acceleration to maximize TCP window size, forward error correction to address packet loss, and packet order correction to fix out-of-order packets.

According to NetForecast's research, WAN optimization improves effective throughput ratios by 5x to 10x on average, with peaks as high as 50x. For this reason we suggest you seriously consider WAN optimization solutions in high-speed WANs.

Appendix A – The ARQ Mechanism within TCP

ARQ adheres to the following steps to ensure reliable data transfer of a single flow between source and destination (see Figure 7).

Step 1: The source releases up to a window's worth of payload bytes into the network and waits for an acknowledgement (ACK) from the destination.

Step 2: The bytes pass through the circuit and multiple network hops during which time they are "in flight". While in transit, the sent bytes are beyond the control of the source and not yet received by the destination.

Step 3: The destination receives the bytes, checks to ensure they are valid, and sends an ACK for them along with an updated window to send the next set of payload bytes.

Step 4: The ACK and the updated window pass through the circuit and are "in flight" in along the return path.

Step 5: The source processes the ACK and releases the buffer of data it was holding on behalf of the application. The counters are then updated the cycle restarts on the next buffer's worth of data to transmit.

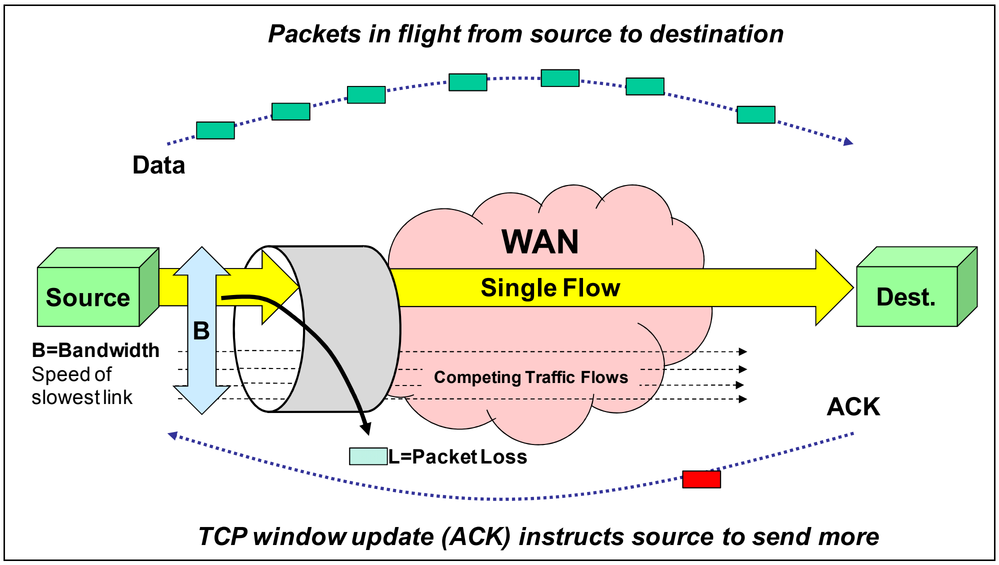

Figure 7 – How One Flow Works across the WAN

References

To read more about the complete array of application delivery system techniques, we encourage you to read our "Field Guide to Application Delivery Systems", NFR5085 September 2006. It is available at the NetForecast website under Reports.

About the Authors

Peter Sevcik is Principal of NetForecast and is a leading authority on Internet traffic, performance, and technology. Peter has contributed to the design of more than 100 networks, including the Internet, and holds the patent on application response-time prediction. He can be reached at peter@netforecast.com.

Rebecca Wetzel is a Principal of NetForecast and a veteran of the data networking industry with unparalleled inside knowledge of the Internet service and product markets. She works with network product vendors and service providers to develop and implement product strategies. She can be reached at rebecca@netforecast.com.

NetForecast helps enterprises understand and improve the performance of networked applications.

Additional information is available at: www.netforecast.com