



High Performance Precision Emulators

GEM Delay & Impairment Emulator

WAN ACCELERATION

- Compression • Caching • Session Optimization •

Overview

Multi-site organizations use a Wide Area Network (WAN), whether the Internet or a private network, to connect the Local Area Networks (LANs) in each location. Many uses of the WAN result in large amounts of data moving between sites, such as hits on high-traffic website hosts, employees accessing centralized information stores, processes consolidating branch information to centralized servers, administrators backing up servers to offsite locations for disaster recovery and business continuity, servers distributing email for thousands of users, applications processing a high volume of transactions, or even communication via IP phones and videoconferencing.



As organizations place greater demands on their corporate networks, skinny WAN pipes can create response and performance problems that affect both the top and bottom line. Slow response from online retail systems can result in abandoned shopping carts and reduced revenues. Slow response from back-office systems, extended time for backups and mirroring, or poor audio and video conferencing quality can translate into higher operating costs and reduced profits.

Increasing bandwidth or increasing the efficiency of the current network can address these issues. WAN acceleration takes the second approach, reducing costs by increasing the effective throughput of the existing network. Most solutions use an array of techniques to get a 10X to 40X improvement in response time or transaction processing rate.

Some specialized techniques targeted to specific protocols, such as the Messaging Application Programming Interface (MAPI) and the Common Internet File System (CIFS), can achieve up to a 100X improvement in performance. With results like these, it's not surprising that organizations are turning to WAN acceleration devices or services rather than carriers to address their WAN performance issues.

WAN applications run over a network that will introduce packet delay, loss and reordering with limited available bandwidth. A successful acceleration solution must provide quality of experience for the user by compensating for negative conditions that occur in IP networks. Anue Network Emulators allow test engineers to introduce those conditions in a controlled and repeatable fashion to predict quality of experience in actual networks.

→ The Technology

Early solutions used compression or TCP windowing optimization to increase WAN efficiency. The current products available have a wider range of techniques available.

Compression. Typical compression techniques replace repeating bit patterns with a short label before sending the data across the WAN. On the receiving side the label is removed and the original pattern is inserted back into the bit stream. (See Figure 1.)

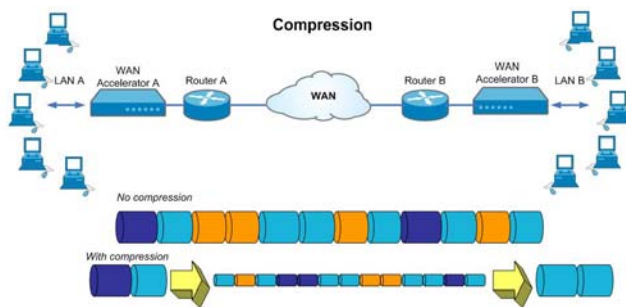


Figure 1: Compression

Depending on the compression scheme and the data type, the number of bits transmitted can be reduced by 20% to 70%. Some proprietary techniques that use parsing and hierarchical trees claim to reduce data size by up to 98%. In addition, by offloading compression to the WAN accelerator, servers can serve more requests, improving throughput and response time.

TCP Optimization. Internet Protocol (IP) sends packets to the destination without checking to see if they arrived. Transport Control Protocol (TCP) maintains the end-to-end connection by requiring periodic acknowledgments from the destination. TCP uses a sliding window to avoid congestion. The window size denotes the number of packets that can be sent before an acknowledgment from the receiver is required. A larger window size means more data can be transmitted before waiting, which results in greater throughput.

When the sender receives an acknowledgement (ACK) before it times out, it increases

the window size for the next transmit. (Scenario A in Figure 2.) If the transmit or ACK is lost, the sender times out and re-transmits. (Scenario B in Figure 2.) If the ACK is de-

layed due to network latency and arrives after the time-out value, the sender assumes the packet was lost and re-transmits. (Scenario C in Figure 2.) Re-transmissions cause TCP to reduce the window size and/or increase the time-out value, which can result in inefficient use of bandwidth.

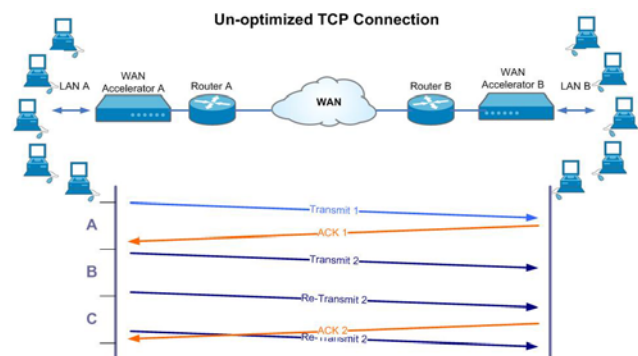


Figure 2: Un-Optimized TCP Connection

TCP uses a slow-start method to gradually ramp up the window size, which can cause sluggish performance. Enabling RFC 1323 TCP Extensions for High Performance can improve throughput, but for high-latency connections the round trip time for acknowledgements can cause long waits for acknowledgement even if there is no congestion and bandwidth is available.

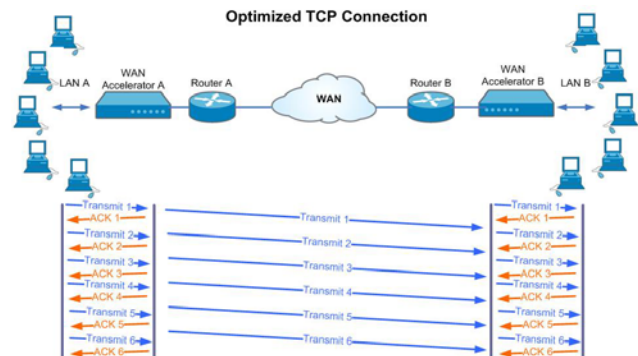


Figure 3: Optimized TCP Connection

WAN acceleration solutions use connection optimization and aggressive windowing methods to improve performance. Rapid accelerate-and-fallback windowing can achieve 700 Mbps, even on intercontinental links. Connection optimization reduces the wait for acknowledgements by terminating the TCP session at the local accelerator rather than at the destination, resulting in near-instantaneous responses rather than long waits for acknowledgments. (See Figure 3.) In addition, connection pooling reuses WAN-side connections for multiple requests, saving setup and teardown time and CPU cycles.

Caching and Mirroring. Caching is used at many levels in computing and networking, from onboard caches in system processors to print job spooling. Client-side WAN accelerators store web pages and other data as it passes through so that subsequent requests can be answered locally by the accelerator rather than requiring retransmission from the remote server. Database applications can provide read-ahead caching to transmit additional pages for a query before they are requested. Caching improves response time and reduces demand on the servers. (See Figure 4.)

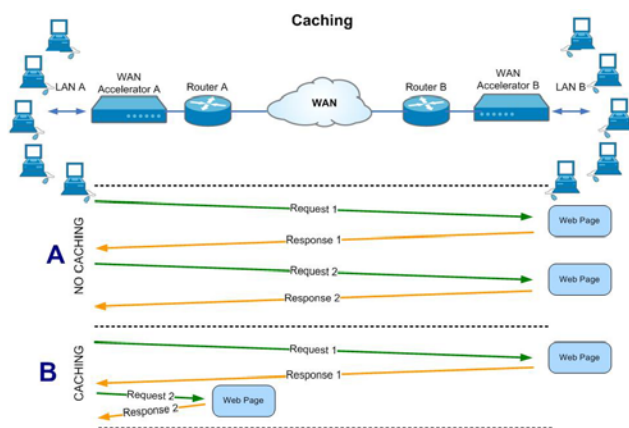


Figure 4: Caching

High-availability mirror systems provide synchronized copies of entire databases at multiple locations. When a change is made locally, it is replicated at all the other sites in the background so that requests from users in those locations can provide the most recent data without having to query a remote server.

Traffic Shaping through QoS. Some WAN accelerators offer QoS features to prioritize traffic. Although not a unilateral acceleration solution, combining QoS with other techniques is useful. Traffic shaping improves throughput for preferred traffic at the expense of lower-priority traffic. Delay-sensitive traffic, such as video or voice, is prioritized and guaranteed delivery. Traditional data traffic, such as email or web page requests, are delivered if available bandwidth permits or are dropped if the higher priority traffic has consumed the available bandwidth. (Can use QoS solution diagram if desired.)

Off-loading CPU-Intensive Tasks. Multi-processor systems are typically licensed per processor, adding hardware and software costs as more processors are required. As the number of transactions increase, so increases the demands on processors that may already be pushed to their limits with ancillary tasks such as compression, encryption and XML processing. By offloading CPU-intensive tasks to an appliance, organizations can not only improve WAN performance but also handle increased traffic without the expense of licensing additional processors. (See Figure 5.)

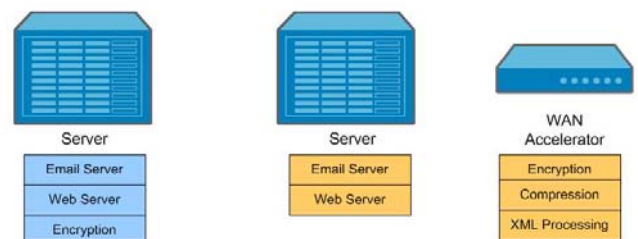


Figure 5: CPU Offloading

Forward Error Correction (FEC). Some WAN acceleration solutions use FEC to reduce the need for retransmissions on lossy networks. FEC sends recovery information in-band with data for indexing to allow for reconstruction of lost packets.

➔ **Testing Considerations**

WAN accelerators are no longer simple compression or TCP optimization appliances. They now employ a wide range of sophisticated techniques to attack the problem on multiple fronts. To understand how the various components contribute to overall per-

formance improvements, each can be tested individually, followed by a system test with all features enabled.

In a Perfect World: The Performance Baseline.

The first step in testing a WAN acceleration solution is to evaluate it in a test lab under optimum conditions. The infrastructure of the test lab should provide an environment with no packet loss, no packet reorder and minimal delay. As all aspects of the solution are tested, such as throughput, transaction response time and scalability, the maximum performance limits of the system are determined. These limits establish the baseline that will be used for comparison against the results of tests in a more realistic environment.

Real-World Testing: The Reality Check.

All the various WAN acceleration techniques have one thing in common – they are designed to improve the performance of systems running across a WAN. To implement a WAN acceleration solution with confidence, test and debugging iterations must be performed under the conditions in which the solution will actually be deployed, and also under worse conditions to account for equipment failure and disaster conditions to verify that they can actually improve performance on real networks. Rather than test on a live network, which is expensive, impractical, and not repeatable, real-world testing is achieved by using network emulation in the test lab.

Each of the tests performed during baselining are performed again with a network emulator injecting impairments such as packet loss, delay, bit errors, sequence errors or duplication in a controlled fashion. The emulator can also model the effects of high traffic load by constraining available bandwidth. The actual delay, impairment and bandwidth values used are determined by various methods.

<p>Educated guess Trial and error</p>	<p>Values known to affect the service can be tweaked to determine failure thresholds. For example, TCP windowing uses packet loss and large round-trip times as indicators for decreasing window size. Values for these impairments can be ramped up, both individually and jointly, to determine the thresholds for performance degradation.</p> <p>Pros: granular control over specific impairment metrics, goal-seeking of performance thresholds.</p> <p>Cons: can be time-consuming if not automated; combinations tested may not reflect real-world conditions.</p>
<p>Network Management System or Protocol Analyzer</p>	<p>A network management system or protocol analyzer can be used to determine the performance of the live network on which the solution will be deployed. Measured values for packet loss, delay, duplication and sequence errors can be used to configure the emulator during testing.</p> <p>Pros: test conditions reflect the actual conditions measured on the production network.</p> <p>Cons: conditions in IP networks are not static. They vary over time as events such as traffic congestion, router flaps and topology changes occur. Testing with a single value averaged over time will not produce an accurate reflection of actual performance. Realistic testing requires impairments that change dynamically over time. Taking thousands of measurements over short intervals and automating the changes in configuration of the network emulator can accomplish this, but it will be complex and time consuming.</p>
<p>Standards-based model</p>	<p>A standard network model for IP network impairment can be used. TIA-921 and ITU-T G.1050 describe a Network Model for Evaluating Multimedia Transmission Performance over Internet Protocol based on actual network data provided by anonymous IP service providers and IP network equipment manufacturers. An algorithm uses typical LAN, access and core network characteristics as input and produces as output dynamic values for end-to-end packet delay, loss and reorder that vary over time in the same manner as conditions in deployed networks.</p> <p>Pros: the model more accurately reflects the way actual IP networks behave than other methods of testing.</p> <p>Cons: since the impairments are changing dynamically, in the case of failure of a test case, the impairment that caused the failure may not be immediately clear. Further troubleshooting may be required.</p>

Before and After

Features that are susceptible to specific impairments should be baselined and then tested in the presence of the impairments known to affect them. As issues are revealed, troubleshooting uncovers root causes and facilitates the debugging of applications or tuning of network parameters. This iterative process is required to assure robust solutions and minimize support costs.

TCP Optimization. TCP windowing is designed to avoid congestion by throttling back transmission when detecting packet loss (as indicated by a lack of a response packet) or excessive delay (as indicated by round trip time values in response packets). After baselining the system without impairments, the thresholds of delay and loss for acceptable throughput are measured without TCP optimization turned on. An analyzer measures throughput metrics while the network emulator ramps up loss and delay, both individually and together. Once the threshold of acceptable performance has been established, TCP optimization is enabled and the tests are performed again to find the new thresholds. Values of interest include:

- 1) the improvement of throughput at the failure thresholds measured without optimization and
- 2) the impairment values at the failure threshold with optimization enabled.

Traffic Shaping through QoS. Bandwidth oversubscription should be tested to verify that lower priority traffic is dropped to accommodate video traffic. Impairment profiles can be configured based on where the QoS is implemented, such as in the experimental bits in the MPLS header (core network traffic engineering), the DSCP/ToS bits in the IP header (aggregation-level QoS) or the 802.1p-bits in the VLAN tag (DSLAM or PON QoS).

→ Conclusion

A successful WAN acceleration solution requires testing under real-world conditions to guarantee that it can provide the desired performance improvements under conditions that occur in IP networks. Using Anue Network Emulators enables engineers to predict application performance, thereby reducing risk, accelerating time to market, and allowing deployment with fewer problems and more confidence.

Forward Error Correction (FEC). FEC is designed to compensate for errors during transmission by reconstructing the actual values based on recovery information transmitted with the data. Without FEC, errored packets are discarded and retransmission is required, resulting in lower effective throughput. Transaction response time (TRT) is measured without FEC under expected error-rates. These times are compared to TRT with FEC enabled under the same error conditions.

Compression. When data is compressed the effect of impairments is increased. Consider a scenario with 2:1 compression. In this case each bit in the compressed data stream is twice as important! Packet loss, therefore, is more significant. One lost packet on the WAN becomes two lost packets at the receiving end after decompression.

Caching and Mirroring. Network emulation can determine whether or not cache performance is sufficient, even in the presence of delays and impairments. Emulation can also help with many caching and mirroring design and implementation considerations. For example, cache sizing can be evaluated under different network scenarios. Other parameters, such as the “cache miss ratio” can also be measured under different WAN latencies.

Offloading CPU-Intensive Tasks. When tasks such as encryption and XML processing become bottlenecks, it means the WAN is no longer constraining performance. Nonetheless, offloading CPU intensive tasks to an acceleration appliance creates testing requirements well suited to a network emulator that can target specific impairments and traffic patterns. As an example, the Anue Emulator can validate the performance of encryption algorithms not only in the presence of impairments, but also by introducing specific traffic patterns known to create challenges.