

Application-Delivery.org

Technology Primer –

**Understanding WAN Acceleration
Techniques**

June 7, 2006

TABLE OF CONTENTS

Overview	3
Data Reduction	3
Compression	5
Latency Mitigation	6
TCP acceleration	6
<i>Window Scaling</i>	7
<i>Selective Acknowledgement</i>	7
<i>Round-Trip Measurement</i>	7
<i>HighSpeed TCP</i>	7
CIFS acceleration	8
Application-specific Acceleration	9
Quality of Service (“QoS”)	9
Loss mitigation	10
Packet Coalescing	11
Wide Area File services (WAFS)	11
Conclusion	12
Appendix A – Technology Comparison across Sample Vendors	13

Technology Primer - Understanding WAN Acceleration Techniques

Overview

WAN acceleration is increasingly becoming more important to enterprises as a key enabler of strategic IT initiatives, including branch office server and storage centralization and business continuity planning.

Recent advances in technology have enabled WAN acceleration to transition from a tactical fix to a strategic IT investment. However, these advances have also led to increased confusion in the marketplace. Never before have there been more WAN acceleration products, leveraging a wider variety of technologies, with varying levels of results.

This paper helps to cut through the confusion by providing a technical overview of the most common WAN acceleration technologies being employed today. It describes how these technologies work, why they are important, and how they vary across different vendors' implementations.

Data Reduction

The most efficient way to accelerate the transfer of information across the WAN is to not send it in the first place. This is the major principle employed by “data reduction”, a new WAN acceleration technology that provides significant benefits in the form of increased WAN bandwidth efficiency and reduced application response time.

In a data reduction scenario, acceleration appliances examine all data in real-time prior to it being sent across the WAN. This information is stored in local data stores on each appliance. Whenever duplicate information is detected, references are sent to the appropriate appliance instructing the device to deliver the information locally (instead of re-sending it across the WAN). By preventing repetitive information from traversing the WAN, data reduction can reduce over 90% of WAN bandwidth. By delivering information from local data stores, data reduction helps to provide LAN-like performance across the WAN.

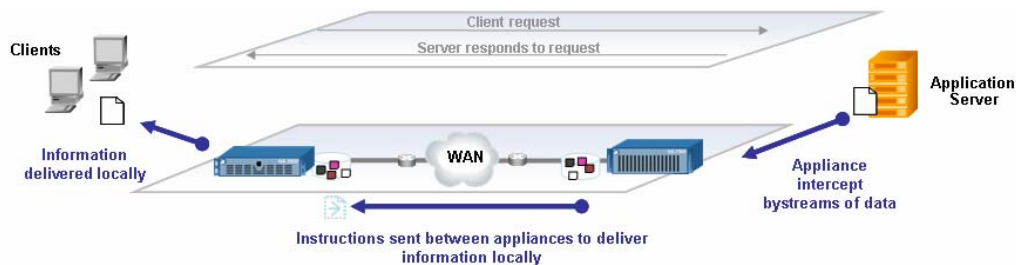


Figure 1. Data reduction detects duplicate information and delivers it locally for LAN-like performance across a WAN

On the surface, data reduction resembles caching, but there are several major differences:

- *Application breadth*: Data reduction detects patterns across many different types of traffic. Caches, on the other hand, work at the object level, and are therefore only applicable to a specific application.
- *Application transparency*: There are no client/server modifications when deploying data reduction. In some caching environments, clients need to be re-configured to point to proxy devices.
- *Coherency*: By preserving all client/server communications, there is no chance to deliver stale or inconsistent information in a data reduction environment.
- *Effectiveness*: Data reduction fingerprints at the byte level, not the object level. This provides a higher hit rate when looking for duplicate data, including the detection of similar information— e.g., files that have been renamed or data that has changed slightly.

Unlike many of the other technologies that have been around for a while and are often well documented in industry standards, data reduction capabilities can vary quite significantly from one vendor's solution to the next. Benefits are often directly related to the architecture that a product employs to achieve data reduction. For example, the following differences are worth noting:

- *Disk vs. RAM*. The most effective data reduction solutions leverage hard drives on appliances to store up to several months worth of traffic patterns. This efficiently eliminates the transfer of duplicate data by leveraging information collected over extended periods of time. A small subset of solutions perform data reduction in RAM (as opposed to using disk drives.). While the two solutions may provide similar short term benefits (e.g., in simple lab tests), significant differences will be seen over time and under increased load. Naturally, the solutions that have better “long term” memory and higher storage capacity will perform better in real world environments as resources fill up and more information is learned about the particular network environment.
- *Effective storage capacity vs. stated capacity*. Vendors use different methods to parse and store data as part of a data reduction solution. Some techniques are more efficient than others, resulting in better usage of available storage space. For example, in a bidirectional solution, a single reference is stored if a piece of information is sent in both directions of a link. This is much more efficient than a unidirectional solution, which needs to re-learn and re-store the data for each direction. Similarly, some solutions will store a single instance of information for all offices connected to an appliance; other solutions store a separate instance for each individual WAN link. The effective storage capacity of the first method is much higher than the second as greater economies of scale can be achieved. Given these differences, two appliances with equal stated storage capacities may not perform identically over time given differences in the two devices' effective storage capacities.
- *Application breadth*. Data reduction solutions that originate and terminate TCP flows are designed to fingerprint and store TCP traffic only. They are best suited for bulk

TCP applications, such as file and email. Data reduction solutions that work at the network layer can support any transport protocol, including both TCP and UDP. This expands the breadth of applications that can be supported to also include VoIP, video streaming, data replication, and other applications. (As the typical large enterprise has over 80 applications deployed across their network, breadth of application support is particularly important when performing data reduction.)

- *Data protection.* Data reduction solutions will take varying measures to protect end user information. Most vendors provide encryption between appliances to protect data traversing the WAN. Some vendors also provide encryption of the local data store to protect information at rest. For maximum protection, 128 bit encryption (or higher) is typically considered best practice. Some vendors perform encryption in hardware so as not to adversely impact performance.
- *Matching granularity and effectiveness.* Each solution differs in how it searches for duplicate data, both in the granularity of matches and deltas and in how effective its fingerprint database is over time. Some solutions work well for identical data sent quickly in succession, but work poorly for derived data, or are not as good at identifying duplicate data sent after several days of intervening usage.

Compression

Compression is used to reduce the bandwidth consumed by traffic traversing the WAN. “Payload” compression uses algorithms to identify relatively short byte sequences that are repeated frequently over time. These sequences are then replaced with shorter segments of code to reduce the size of transmitted data. Simple algorithms can find repeated bytes within a single packet; more sophisticated algorithms can find duplication across packets and even across flows. “Header” compression can provide additional bandwidth gains by reducing packet header information using specialized compression algorithms.

The gains realized by compression techniques vary depending on the mix of traffic traversing the WAN, but are fairly consistent across different vendors’ solutions. Text and spreadsheets, for example may yield 2-5x compression ratios. On the other hand, pre-compressed content, like zip files, cannot be compressed much further. Therefore, additional compression does not help these file types. Enterprises deploying compression technology will typically see around a 50% improvement in WAN utilization, which is the equivalent of doubling the effective WAN bandwidth. Some additional benefits may be garnered from solutions that apply compression across various flows of traffic (called “crossflow compression”) and can employ compression techniques on UDP traffic. VoIP, for example, can significantly benefit from UDP header compression when used in conjunction with other techniques, such as packet coalescing (see below).

Many of the compression techniques are based on algorithms that have been developed over the last few decades. A common basis is Liv-Zemfel coding (LZ77, LZ78). A plethora of useful resources have been collected on the subject of compression, some of which can be found at: <http://www.ics.uci.edu/~dan/pubs/DC-references.html>

Latency Mitigation

The time for information to go from a sender to receiver and back is called the latency of the network. Since the speed of light is constant, minimum latency is directly proportional to the distance traveled between the two endpoints of communication. In other words, the longer the distance, the longer the minimum delay. In real-life, the latency is also impacted by queuing and processing delay in routers and other network elements along the path.

Latency often has a big impact on the performance of applications across the WAN. For TCP bulk data transfers, latency can severely limit throughput. This is primarily because TCP congestion control limits the amount of unacknowledged data in transit. Once the amount of unacknowledged data reaches the congestion window size, transmission of new data is postponed until older data is acknowledged. Since the maximum window size in a default implementation is 64 KB, maximum throughput is calculated by the following equation: 64 KB / latency. For example:

Latency	Maximum throughput
100 ms	5 Mbps
200 ms	2.5 Mbps
600 ms	853 Kbps

In addition to the basic TCP limits, some common communications protocols, such as Microsoft CIFS, further exacerbate the challenges of WAN latency. That is because simple actions, such as retrieving the attributes of a file, require numerous round trips across the network. While this is fine when the client and server are located on the same LAN, it can take significant time when performed across high latency environments. As a result, it is often difficult to deploy many enterprise applications across wide geographic distances.

Going “up the stack” there are several acceleration techniques that are used to overcome the latency issues associated with application delivery across a WAN. These include TCP acceleration, CIFS acceleration, and application-specific acceleration techniques:

TCP acceleration

The TCP protocol was designed to operate reliably over almost any transmission medium regardless of transmission rate, delay, corruption, duplication, or reordering of segments. However, the introduction of high speed telecommunications links has resulted in ever-higher transmission speeds, which often exceed the domain for which TCP was originally engineered.

Various extensions have been proposed over the years to improve the performance of TCP across high latency (and loss) links. These “TCP acceleration” techniques fall into 3 primary categories:

- 1) *Window Scaling* - The Bandwidth*Delay Product (BDP) determines the amount of data that can be in transit in a network. It is the product of the available bandwidth and the latency. The TCP Window determines how much data can be transferred before the end-system stops and waits for acknowledgements of received packets. Throughput is bound by BDP - if the TCP window is lower than the product of the latency and available bandwidth, clients cannot send acknowledgements back fast enough. To address the above problem, the TCP Window needs to be large enough to fit the $\text{maximum_available_bandwidth} \times \text{maximum_anticipated_delay}$.

The TCP header uses a 16 bit field to report the receive window size to the sender, which creates static window sizes that are limited to 65,535 bytes. This relatively small window size limits the amount of data “in-flight” and reduces throughput, especially in higher-latency networks. To circumvent this problem, Window Scaling was introduced. This defines a Window Scale TCP option that is multiplied against the window size value found in a TCP header to obtain window sizes as large as 1GB.

- 2) *Selective Acknowledgement* - Packet losses can have a catastrophic effect on throughput. Originally, properly-operating TCP implementations would cause the data pipeline to drain with every packet loss, triggering the recovery process built into the protocol. The Fast Retransmit and Fast Recovery algorithms were created to alleviate this problem by providing a mechanism whereby the network can recover from one packet loss per window without draining the pipeline. However, these algorithms do not address environments where more than one packet is lost per window, which often happens in long WAN links (with high latency).

Selective acknowledgements (ACKs) supplement these algorithms by providing a mechanism for handling multiple packet loss in a WAN environment. Unlike the normal process whereby a cumulative acknowledgment is provided across all TCP packets, selective acknowledgments give the sender a complete picture of which segments are queued at the receiver and which have not yet arrived. Therefore, more granular decisions can be made regarding packet loss and recovery.

- 2) *Round-Trip Measurement* - TCP implements reliable data delivery by retransmitting segments that are not acknowledged within some retransmission timeout (RTO) interval. Accurate dynamic determination of an appropriate RTO is essential to TCP performance. RTO is determined by estimating the mean and variance of the measured round-trip time (RTT), i.e., the time interval between sending a segment and receiving an acknowledgment for it. A TCP option exists, called "Round Trip Time Measurement", enables RTTs to be calculated more efficiently. Rather than using a single sample per window, a timestamp is placed on all packets (including retransmissions). This leads to more accurate RTO behavior with negligible computational cost. *There are also proprietary methods for estimating round trip time that can be used by some vendor solutions.*
- 3) *HighSpeed TCP*- TCP performance starts to degrade beyond 100M bit/sec due to its window-adjustment algorithm. In its congestion-avoidance phase, ordinary TCP

increases its sending window by one packet every round-trip time. When it detects congestion, it cuts the window in half. For a high-bandwidth, high-latency connection, this can result in several hundred seconds of latency.

HighSpeed TCP is a modification to TCP's congestion control mechanism for use with TCP connections with large congestion windows. It alters how the window is opened on each round trip and closed on congestion events as a function of the absolute size of the window. When the window is small, HighSpeed TCP behaves exactly like ordinary TCP. But when the window is large, it increases the window by a larger amount and decreases it by a smaller amount, where these amounts are chosen based on the precise value of the window in operation. The effect of these changes is that TCP achieves high throughput with more realistic requirements for packet drop rates. Equivalently, HighSpeed TCP has more realistic requirements for the number of round-trip times between loss events, enabling TCP to perform better in high-bandwidth, high-latency environments.

CIFS acceleration

Common Internet File System (CIFS) is a protocol developed by Microsoft for remote file access that allows most applications to open and share files across the Internet or other IP based networks. Some specific capabilities of CIFS include file access, record locking, read/write privileges, change notification, server name resolution, request batching, and server authentication

CIFS is a fairly “chatty” protocol, requiring hundreds, or even thousands, of round trips to successfully transfer a single file. This is typically not an issue when file servers are deployed on the same LAN as clients. However, when CIFS is used across a WAN, as is the case when branch offices are accessing file servers located within a centralized data center, both latency and bandwidth constraints across the WAN can adversely impact file sharing performance.

To overcome the limitations of using CIFS across the WAN, several vendors have implemented the following:

- *CIFS Read-ahead* - when a user is working with a file, acceleration devices can generate read ahead requests within the file in order to pipeline operations to the server, thus eliminating the round-trip delay associated with waiting for acknowledgement. This minimizes the latency associated with read operations.
- *CIFS Write-behind* – acceleration appliances can pipeline write operations on behalf of a client, thus eliminating round-trip delays associated with waiting for acknowledgements. This minimizes the amount of round trips required to perform write operations, improving performance without risking data integrity.

Application-specific Acceleration

Some vendors perform application-specific latency optimization techniques to improve the performance of specific types of traffic across the WAN, including SQL, HTTP and Microsoft's Messaging API (MAPI). Furthermore, "prefetching" of content can be used to overcome latency when delivering some applications across the WAN.

While some incremental performance benefits can be had using these techniques, there can be management complexities associated with these approaches. In addition, altering client/server behavior can sometimes lead to unexpected results, including the possible corruption of data. There is also a potential security concern with some application-specific acceleration techniques, such as prefetching, whereby user credentials can be left in the clear on unencrypted drives, violating corporate security/compliance standards. Lastly, many of the benefits of application-specific acceleration are often short lived as new versions of application code are released to address the shortcomings of previous versions. For example, many of the MAPI issues associated with Outlook 2000 were addressed when Microsoft introduced Exchange 2003, which includes a new "cached" mode of operation that improves performance across a WAN.

Quality of Service ("QoS")

In an effort to maximize WAN utilization, most enterprises will oversubscribe their WAN links. When demand exceeds the capacity of a WAN link, and traffic is contending for the same limited resource, less important traffic (such as web browsing) may take bandwidth away from business-critical applications. To prevent this, some WAN acceleration solutions implement Quality of Service techniques to classify and prioritize traffic based on applications, users, and other criteria.

QoS involves three functions: 1) classification of packets into traffic classes based on characteristics such as source, destination addresses, and/or applications and 2) queuing and service mechanisms that are used to apply service policies based on these classifications, including bandwidth allocation.

Application acceleration appliances can introduce a challenge to the way that QoS is implemented. These devices sit on the LAN side of the WAN router, in both the data center and branch offices. As traffic flows from the internal network to the WAN, application acceleration appliance remove repetitive information, compress headers and payload content, modify IP addresses and port numbers, and sometimes even encrypt traffic. By obscuring the original data (and its headers), application acceleration appliances can prevent downstream devices, such as WAN routers, from applying QoS classification logic based on normal packet inspection.

To account for this, QoS classification can be performed upstream of the application acceleration appliance (by a host, Ethernet switch, or LAN router) and then honored by the appliance. Or, QoS classification can be performed by the application acceleration appliance itself. This latter approach is often more desirable as enterprises sometimes do

not have intelligent upstream devices in a branch office that are capable of performing QoS classification.

With respect to QoS queuing and service disciplines, most application acceleration appliances can pass existing tags to downstream WAN routers, enabling these devices to participate in the QoS process as they would normally. In addition, queuing and service disciplines can be enforced within the application acceleration appliance itself. This is often desired, because application acceleration appliances are well equipped to collect real-time metrics, like packet loss and delay, and adapt QoS techniques accordingly. (This is usually not part of a typical WAN router's feature set).

Loss mitigation

Even when the physical-layer of a WAN is error-free, some technologies and provisioning practices still lead to packet-loss at the network layer. In fact, it is not unusual to see network packet loss rates as high as 8% in some networks. When this type of loss is coupled with high latency and the retransmission and congestion-avoidance behavior inherent to TCP, it is not surprising that application performance suffers across a WAN.

Forward Error Correction (FEC) is a technology that is well known for its ability to correct bit errors at the physical-layer. This technology is often adapted to operate on packets at the network-layer to improve application performance across WANs that have high loss characteristics. FEC works by adding an additional error recovery packet for every "N" packets that are sent across the WAN. This FEC packet contains information that can be used to reconstruct any single packet within the group of N. If one of these N packets happens to be lost during transfer across the WAN, the FEC packet is used on the far end of the WAN link to reconstitute the lost packet. This eliminates the need to retransmit the lost packet across the WAN, which dramatically reduces application response time and improves WAN efficiency. An advanced implementation will dynamically adjust FEC overhead in response to changing link conditions for maximum effectiveness in environments with high packet loss.

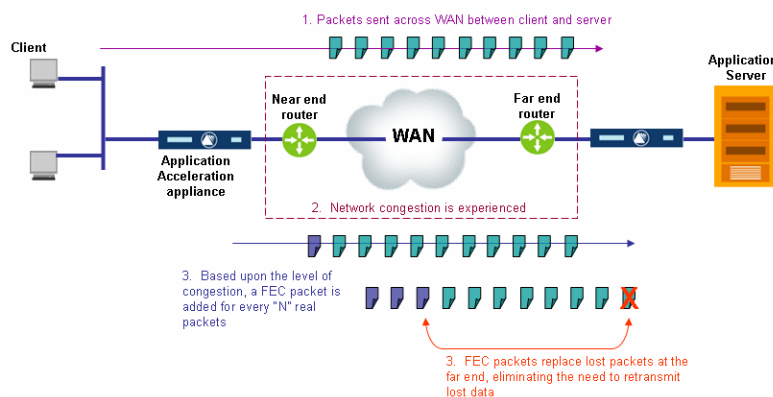


Figure 2. Forward Error Correction eliminates the need to retransmit the lost packet across the WAN, which dramatically reduces application response time and improves WAN efficiency

Packet-level FEC avoids delays that come with multiple-round-trip retransmissions. This enables WANs to easily recover from packet loss due to a variety of network layer conditions, such as queue overflows and constrained bandwidth links. With packet level FEC, enterprises commonly see significant improvements in application performance – up to a ten-fold performance increase in some WAN environments.

Packet Coalescing

When packets are small, packet headers consume substantial bandwidth in comparison to the amount of end-user data transferred. Packet coalescing combines multiple user packets traveling between the same two sites into a single coalesced packet. Used in conjunction with header compression, this amortizes a single header over multiple packets thus decreasing overhead, and therefore bandwidth requirements. Packet coalescing is particularly beneficial for Web applications, VoIP, and interactive applications, like Citrix.

Wide Area File services (WAFS)

“WAFS” is a widely mis-used term that has sometimes come to mistakenly represent any solution that accelerates the performance of file services. In reality, WAFS is a caching technology that makes it possible to access remote file services as though they were local. Like other caching solutions, a WAFS appliance simulates an application server, enabling local delivery of specific content. It sits between clients and applications, watching all requests and locally saving copies of the responses. If another request is made for the same “object”, the WAFS appliance acts as a “proxy”, delivering information directly without having to go back across the WAN to the original server.

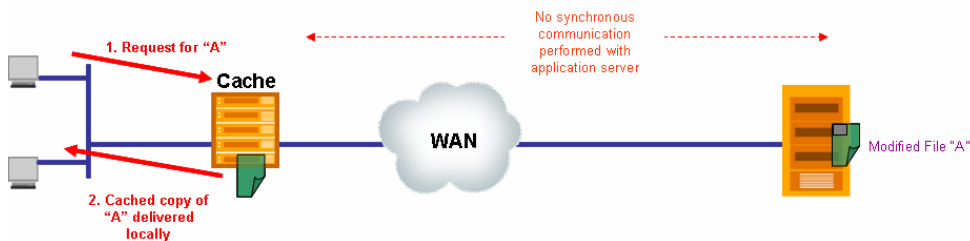


Figure 3. A WAFS appliance acts as a proxy, delivering information directly without going back to the original server.

WAFS enables users to access files services faster, and reduces costs as remote offices don't need file servers, backup equipment, and additional staff to look after remote office storage concerns. Also remote data is consolidated in the data center and thereby held more securely. In addition, by acting as a proxy file server, WAFS helps to ensure business continuity in the event that WAN connectivity is lost.

However, WAFS has struggled to gain industry acceptance for a variety of reasons. The biggest is due to a lack of applications supported – i.e. WAFS only accelerates file services, and is therefore not as cost effective when compared to other WAN acceleration solutions that also accelerate other key enterprise applications, such as email, web, VoIP, etc. In addition, because WAFS is using caching technology, it has a potential for coherency issues. In other words, when clients are retrieving and modifying information stored in a local cache, it is easy for this information to get out of synch with information stored in the original application server. Other challenges pertaining to ease of use, security, and scalability have hampered WAFS from achieving widespread acceptance.





For the above reasons, WAFS is often marketed as a feature within vendors' larger WAN acceleration solutions. In other words, it is a mode that enables files to be delivered locally via a proxy. This might be desired in environments where local file services must be protected against WAN failures. However, it does not protect other applications against WAN failures, and the coherency issues that this solution presents may outweigh the resiliency benefits. It is important to point out, however, that other techniques like data reduction and compression are still required to accelerate email, web, and other business applications when WAFS is employed. There has been some confusion in the marketplace around this point – while a solution that includes WAFS may benefit many applications, WAFS in and of itself accelerates only file services.

Conclusion

There are many technologies available to address the various issues associated with application delivery across a WAN. The best solution will implement a variety of techniques, both old and new, to improve bandwidth efficiency while reducing perceived application response time.

For more information on WAN acceleration technologies, including detailed examples of how distributed enterprises have deployed these technologies to address application delivery and disaster recovery challenges, please visit www.application-delivery.org

Appendix A – Technology Comparison across Sample Vendors

				
Data Reduction	“Network Sequence Caching” -Uni-directional -Up to 500 GB per appliance -Bulk TCP only -Tens of thousands of flows -128 bit software encryption (WAN only)	“Scalable Data Referencing” -Bi-directional -Up to 512 GB per appliance -Bulk TCP only -Up to 4,500 flows -64 bit software DES encryption (WAN only)	“Network Memory” -Bi-directional -Up to 2 TB per appliance -TCP, UDP, and real-time -Tens of thousands of flows -128 bit AES hardware encryption (local store + across WAN)	No (Local file caching only)
Compression	-Crossflow payload -Header compression	-Single flow payload -Header compression	-Crossflow payload -Header compression	Router only
TCP Acceleration	Yes (“Packet Flow Acceleration”)	Yes (Limited to 4,500 TCP flows per appliance)	Yes	Router only
CIFS Acceleration	Yes -Read ahead -Write behind	Yes -Read ahead -Write behind	Yes -Read ahead -Write behind	WAFS appliances only
QoS	Application based policies and tagging	No	Application based policies and tagging	Router only
Loss Mitigation	Adaptive FEC	No	Adaptive FEC	Router only
Coalescing	Yes	No	Yes	Router only
Offline File Services (WAFS)	No	Yes (Option for “proxy file services”)	No	WAFS appliances only