What is Network Acceleration?

How do WAN Optimization, Network Acceleration, and Protocol Streamlining work, and what can they do for your network?
Contents

Introduction 3
Availability Improvement 4
Data Reduction 5
Transport Acceleration 6
Application Optimization 7

What is Network Acceleration?
Introduction

We’ve all heard the frustration in users’ voices as they complain about a web page that just won’t load, no matter how many times they mash the refresh button. And we’ve felt the rage through the phone when they describe how the application froze, then crashed, just as they pushed the save button.

Each of the situations described above are caused by poor network performance. In response to the growing pain of subpar performance, innovators have developed a wide range of solutions known broadly as “WAN Optimization” or “Network Acceleration”.

In a previous piece, “10 Reasons your WAN is Broken”, we discussed many of the underlying causes of network impairments. In this eBook, we will examine the techniques used to overcome poor network performance.

Symmetric versus Asymmetric Acceleration

Network Acceleration can be deployed two different ways. Symmetric acceleration techniques are deployed on the two sides of a network connection. This can take the form of two hardware devices, or a central acceleration server and distributed software clients. In either case, systems on both sides of the network coordinate to accelerate traffic between them.

In asymmetric acceleration, instead of an acceleration system on each side, there is a single, centralized acceleration system. This centralized system provides improved network performance for all clients connected to the data center without requiring any modification to the clients.
Availability Improvement

The first set of network acceleration techniques are Load Balancing and Offloading. These are a form of asymmetric acceleration and center around increasing availability – that is, multiplying the ability of servers to provide content.

**Load Balancing** splits a workload across multiple physical or virtual servers. Incoming requests can be directed to the least busy system, resulting in faster response times for the client. In addition, load balancing allows services to survive hardware failure. If a server goes down, requests will be directed to another system.

**Offloading** moves some of the server workload onto a different device. By reducing the work required to process each request, servers can handle more requests. There are many different server functions that can be offloaded:

- **SSL**: Secure Sockets Layer (SSL) encrypts communication between the client and server. Moving SSL from the server to an offload device is valuable for two reasons. First, the overhead of maintaining SSL connections and encrypting and decrypting packets is removed from the server. Second, because the offload device sees traffic from the server in clear text, it can perform other optimizations like compression and caching.
- **Compression**: Modern browsers can accept compressed content, which is computationally expensive but excellent for reducing bandwidth. Instead of the server compressing each response, it sends uncompressed data which the offload device intercepts, compresses, and passes on.
- **Caching**: Some offload devices are capable of detecting commonly accessed resources and caching them locally. When future clients request the cached data, the offload device intercepts the request and responds with the content. This reduces the number of requests the server must handle.

Local versus Global Load Balancing

Local Load Balancing directs traffic within a datacenter. It is accomplished through bidirectional Network Address Translation. When an incoming request is intercepted, the load balancer determines which host should service the request, and then rewrites the packets to forward the request to the appropriate host. When the load balancer receives the response from the host, it rewrites the packets with its own IP address before sending the response on to the client. Thus, the load balancer hides the pool of servers, and clients believe they are communicating with a single host.

Instead of balancing traffic across servers within a datacenter, Global Load Balancing distributes traffic across data centers. When a user accesses a website, the domain name must be converted to an IP address through a process called DNS Resolution. Global Load Balancing operates by returning different IP addresses to route traffic to different data centers. Users can be directed to the data center with the lowest load or to the geographically closest data center to reduce latency.
Data Reduction

The next set of techniques improve performance by reducing the bandwidth of the data transmitted across a network. They are Data Deduplication and Caching. Unlike Availability accelerations which can be done by a single centralized system, these techniques are symmetric and are deployed on either side of a link.

Data Deduplication is a form of compression where blocks of data are assigned a unique reference. Redundant copies of these blocks can be replaced with the reference. When a network accelerator sends data, it generates references for the individual blocks, and sends the references with the blocks. The next time its peer across the link requests a block, the accelerator sends the reference instead. The receiving system stores the block the first time it is seen, and in the future can replace the reference with the original block to reproduce the original data stream. Data pattern analysis can reduce the amount of storage required by only deduplicating the most commonly transmitted data. A further optimization is variable sized blocks, also called stream-based deduplication. While this may sound similar to Compression Offload discussed earlier, they are some key differences. Compression offload interoperates with compression capability built-in to browsers, and thus is asymmetric. It only compresses repeated data within a single file. Deduplication is symmetric and is deployed on both sides of the WAN. It can deduplicate data transmitted across days or weeks.

Caching is similar to Data Deduplication in that network accelerators store frequently seen data. However, instead of using references to communicate data across the network, a caching system intercepts the request for data, and serves a local copy from its store. Thus, no traffic is sent across the link. Unlike Offload Caching, this symmetric caching operates over the network link and requires two accelerators to coordinate to prevent stale data from being served.
Transport Acceleration

While Data Reduction accelerates performance by decreasing the amount of data that needs to be transferred across the WAN, transport optimization reduces the time required to complete a transfer by increasing goodput. It accomplishes this by improving the efficiency of the transport layer protocol. Like Data Reduction, this is a symmetric technique that is deployed on both sides of a link.

TCP Acceleration: TCP is a core Internet protocol and serves as a base for many higher level protocols. Over the years, many optimizations have been made to TCP to improve its efficiency. For example, by default, when the first in a sequence of packets is lost, the entire sequence is resent. Selective Acknowledgement (SACK) allows a receiver to communicate exactly which packets need to be retransmitted. However, many clients and servers do not implement all the modern TCP extensions. A network accelerator can improve efficiency by replacing these systems’ native TCP with the more efficient version.

High Performance Protocols: While TCP is an excellent general purpose protocol, it is not ideal for every application. For example, one of TCP’s key features is protocol friendliness. A friendly protocol detects when multiple flows are competing for bandwidth, and lowers its transmit rate to help reduce congestion. For strategic applications with critical data, friendliness means essential data will be delayed to accommodate other, less important users. All of this means that for critical, real-time, or latency sensitive applications, something better than TCP is required.

High Performance Protocols are purpose built to deliver high-bandwidth data across impaired networks. They use low-latency retransmission and packet forward error correction to repair packet losses. Latency aware reordering windows are used to eliminate duplicate and out of order packets. High Performance Protocols guarantee lossless delivery and maximum goodput even in the face of severely challenged networks.

What is Goodput?
When people talk about the speed of a network, they could be referring to one of three different metrics:

Bandwidth
This is the theoretical maximum amount of data that could be sent down a channel. It is limited by the clock rate of the physical media. For example, 1 Gig Ethernet can transport a billion bits a second.

Throughput
This is how much data can actually be transported, which is less than bandwidth for a few reasons. First, minimum gaps are required between packets. Second, not all devices in the link path may be capable of sustaining the line rate. Finally, multiple paths utilize a single network segment, so traffic from A to B competes for bandwidth with traffic from C to D.

Goodput
This is how much useful information is transported. Protocol overhead and retransmission consume bandwidth, but are not seen by the application. Flow control and congestion avoidance may reduce the amount of data sent beneath maximum capacity. Goodput is the amount of data delivered to the application and is often the only metric that users care about.

What is Network Acceleration?
Application Optimization

Application optimization streamlines network performance by improving the operation of application layer protocols. Some of these optimizations can be deployed asymmetrically with a single device, while others are symmetric and require devices on both sides of the link.

Application layer protocols handle high level tasks like network file access (CIFS, NFS), database access (TDS, Oracle 9i/10gR2), email (MAPI), and web browsing (HTTP, HTTPS). Many of these protocols were designed to operate over LANs, and are ill-suited for high-latency WAN links.

While the mechanics of acceleration are specific to each individual protocol, the general principle is to improve latency by reducing protocol chattiness. For example, to read a single file, CIFS requests the first chunk, and then waits for the response. Then it requests the next chunk, and waits for a response. An acceleration device can accelerate CIFS through a technique called Read Ahead. First, it detects a client performing a sequential bulk read. Then, it fabricates read requests to the server for the rest of the file. When the client requests the next chunk of the file, the acceleration device returns it immediately. The write path can be optimized in a similar manner through Write Behind. Here, the acceleration device responds to the client by immediately acknowledging the write request. The effect is that the client perceives the exchange occurring at LAN speed, while the acceleration device buffers data for communication across the WAN.

What is Network Acceleration?
Now you understand the different techniques used for accelerating network performance, but choosing the right mix is difficult. Which techniques will yield the most benefit for your application?

In the next piece, *Choosing the Right Network Acceleration Solution*, available at [www.rtlogic.com/products/datadefender](http://www.rtlogic.com/products/datadefender), we discuss how these techniques are deployed in commercial products, and which techniques benefit different applications.