CHAPTER 6

WAN and Application Optimization Technologies

WAN and application optimization comprises a framework of technologies that improves the network application experience at the branch, and makes better use of limited network resources. In some cases, the user experience simply needs to be maintained while other changes occur. For example, during server consolidation (a process where branch-based servers are relocated to a centralized data center over the WAN, the reliance as well as stresses on the WAN network increase but the user experience needs to be at least maintained. Additionally, given resource contention on the WAN, business criticality must map closely to network usage and access.

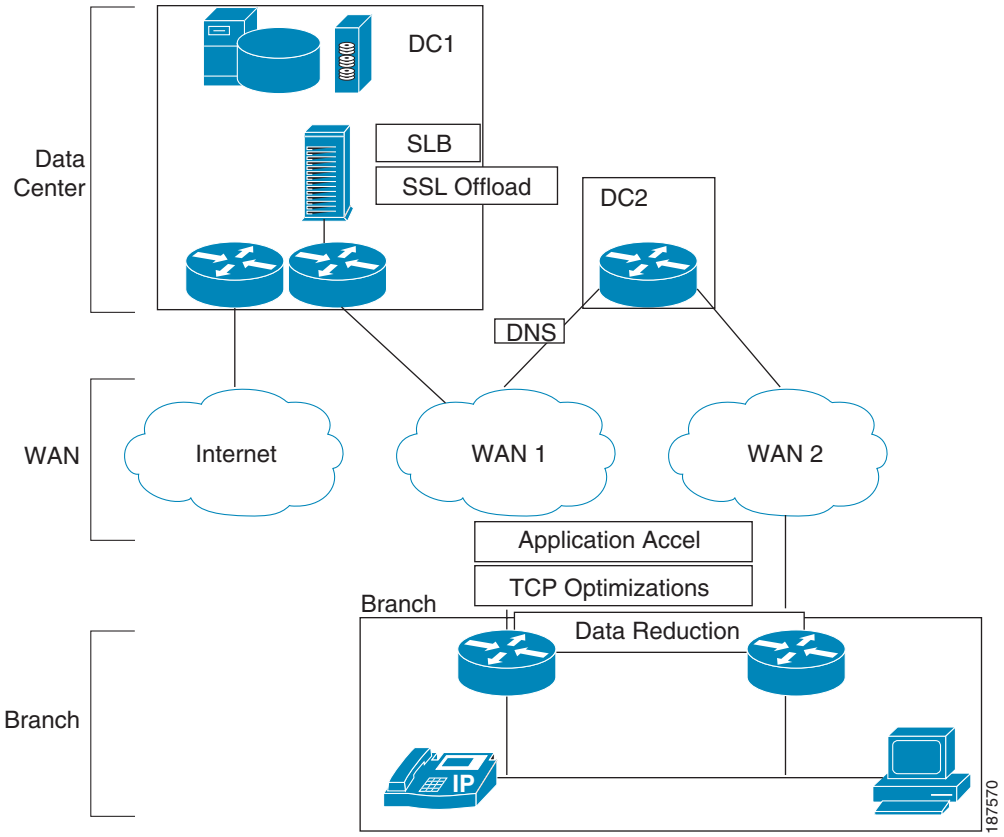
Various mechanisms are available for optimizing the WAN, ranging from technologies that allow horizontal scalability (the ability to cluster multiple devices, rather than vertical scalability- which requires more power in each device), such as server load sharing, advanced compression technologies, dynamic routing to place traffic on the best path, and intelligent flow replication reduction tools, to name a few.

This chapter explores this optimization framework, with an emphasis on basic issues being targeted, and on individual technology solutions. Note that not all of the technologies described in this chapter are part of the currently packaged WAN optimization solution.

6.1 Areas of Interest

As shown in Figure 6-1, the network between the branch and the datacenter can be divided into multiple spheres of work, each with its own challenges and opportunities for optimization.

Figure 6-1 Simplified View of a Typical WAN Topology



6.1.1 Layer 3 End Point Optimization and Server Selection

Because network communications is about connecting parties so that meaningful transactions can take place, one of the first and easiest optimizations is selecting the specific end-hosts involved in the transactions. When a network application user wants to use the application, he or she will interact with a server or a set of servers. The specific server is not relevant to the network application, as long as the server can handle the transaction.

Therefore, it is possible to have multiple servers that can each service the network application. These servers can even be geographically dispersed so that they are not only redundant and share the overall load, but can also be closer to the various clients. The capability to match a specific server to a client request is a form of optimization. After the server Layer 3 (L3) endpoint is established, it is generally very disruptive to the user, the network application, and the transaction to try to move the transaction to a different L3 address.

6.1.2 DNS-Based Optimization

When starting a network transaction, one of the first things that a client attempts to do is identify the L3 address for the remote peer. Generally, this L3 translation is done using Domain Name System (DNS), where a human-readable name (for example, erp.example.com; this hyperlink is not real) is converted to a routable IP address (such as 209.165.201.1 or 2001:0DB8::1). Along with translations to IPv4 addresses (called A records) and IPv6 addresses (AAAA records), DNS can translate services into specific L4 endpoints. For example, DNS service (SRV) records enable a client to locate a LDAP server (for example, Query sent for_ldap._tcp.example.com).

Because the authoritative DNS server returns the mapping between the DNS name and the L3 address, there is an opportunity to select a specific geographic location, or even a specific server, that is best positioned to handle that specific client. For example, assume that the client is located in Australia, and that servers in India, Germany, and the United States can all handle the transaction equally well.

An optimizing authoritative DNS solution, such as the Cisco Global Site Selector (GSS) product, can measure the latency between each server location and the client, and consider liveliness, loads, and so on. GSS can then determine which server is best able to answer the client query. Using our previous example, the Australian client might be directed to India or the United States, depending on the location of the lowest latency and best network conditions.

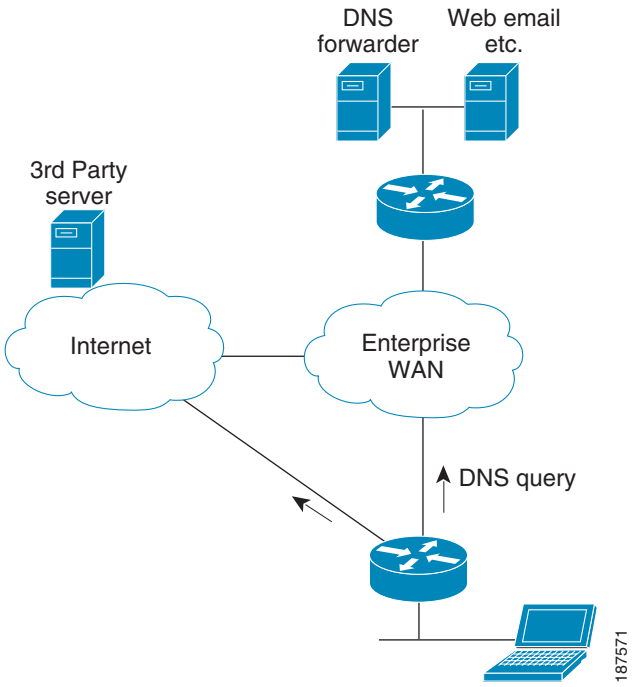
6.1.3 IOS DNS Views feature

In some cases, DNS can direct the client to a suboptimal server because of how the DNS query is sent. In many enterprise networks, clients are configured with the address of a DNS caching forwarder that queries the actual DNS system on behalf of the client. This approach is a problem when the DNS forwarder queries the optimizing authoritative DNS solution; the address and network location of the DNS forwarder is used instead of the address and network location of the client. Therefore, the client is sent to the server that is best for the corporate DNS forwarder, not for the client, which might actually talk to the server over a completely different path.

Figure 6-2 shows such an example: the server that the client needs to work with is on the Internet path available through the branch router. The corporate DNS forwarder query to the Internet returns the best third-party server address for the corporate network. With the IOS DNS views feature, the IOS router can intelligently send queries to the corporate DNS forwarder instead of the Internet based DNS forwarder. The separation of DNS queries is based on the domain name.

For example (the hyperlinks are not real), suppose that the third party server is at purchasing.thirdparty.com and the corporate email cluster is at email.example.com. The DNS views feature can send queries in the example.com domain to the corporate DNS servers, while everything else is sent to Internet servers using the branch site IP addressing. Additionally, this feature can be used for non-Internet cases, such as when the branch is connected over a single link to a MPLS VPN that provides extranet services to multiple companies that each operate their own optimizing authoritative DNS servers.

Figure 6-2 DNS Views Feature



6.1.4 Anycast Addressing

Another, much simpler but more limited mechanism for redirecting clients to a better server is to use IP anycast addressing schemes. With anycast, the same L3 address is assigned to multiple physical servers, and the network routing protocol selects the “nearest” server. Because the same IP address is used for each server, this severely limits the kind of connection that can be established. The network cannot guarantee that the next packet sent to the server IP address will be received by the same server. While anycasting has been used with great success to optimize communication and distribute the load for some specific applications (for example, the root DNS nameservers use anycasting), it should be noted that the application and network protocol being used must be compatible with anycasting technology and is not commonly deployed.

6.1.5 Layer 7 Redirection

In L7 redirection, a client can initially connect to a well-known L3 address, and then is redirected to the actual best server using the L7 protocol. This approach, provided by the Cisco Application Control Engine (ACE) product, is often used with HTTP redirects. For example (the hyperlinks are not real), a user might try to connect to http://wwwin.example.com/erp, but the server at wwwin.example.com knows that the erp network application is best served from another server called www-erp1.example.com, which hosts the erp application, is lightly loaded, and is geographically closer to the client. The wwwin.example.com server sends an HTTP redirect message to the client, asking it to connect to the www-erp1.example.com server. The initial connection and subsequent redirect request can be obtrusive and alarming to the user, and many non-HTTP applications do not support L7 redirection.

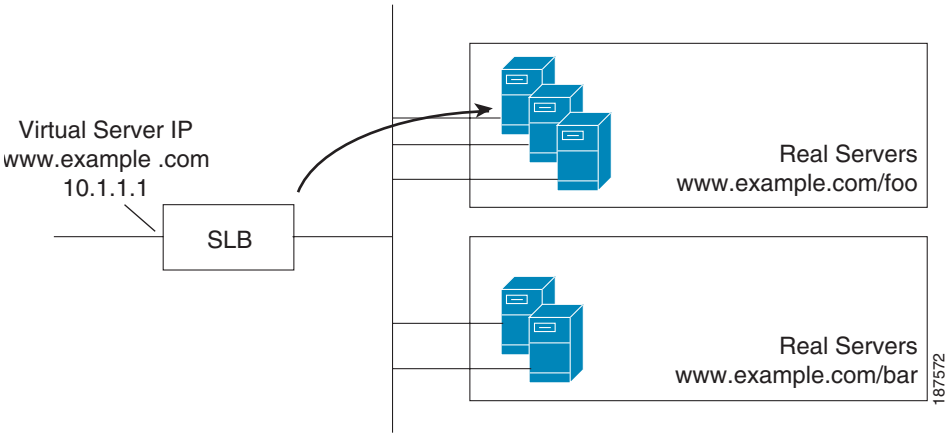
6.1.6 Local Server Load Balancing

Server load balancing (SLB) is based on the concept of front-ending a number of similar real servers with a virtual IP address. The client establishes an L3 (and possibly an L4) relationship with a server load balancer which, at connection initiation time, selects the best real server to terminate the connection on. This provides optimization, in that the client gets access to the network application in a resilient manner (if a real server is down, the SLB device takes that server out of the pool), that can also consider performance.

Applications or subapplications that are known globally by the same IP address can be divided into multiple pools of real servers that service the application. In the multipool case, the SLB device, such as Cisco IOS or Cisco ACE, parse beyond L4 (TCP/UDP port information) and try to identify the subapplication in order to map the session to a real server. For HTTP, a specific URL can denote the subapplication, as shown in Figure 6-3.

Because SLB devices hide the L3 information about the real server from the client, the client is never aware of the specific real server that that serviced it. While for a specific TCP session the same real server will definitely be used, it is less guaranteed that across multiple TCP sessions that the same real server will be used for a client. Depending on the implementation of the network application, "stickiness" to a specific real server may be required. The SLB devices generally provide a variety of mechanisms for stickiness including client IP, insertion of HTTP cookies etc. It is important to understand that while the network operator may not have a choice about whether to apply stickiness, it will limit the flexibility of loadsharing amongst the real servers.

Figure 6-3 SLB Example



Many SLB devices, such as Cisco ACE, incorporate a Secure Socket Layer (SSL) offload feature that is not directly related to L3 endpoint selection and optimization. Using the SSL offload feature, the SLB device terminates the SSL session and exposes only a cleartext session to the real server. Other common high-CPU utilization functions, such as HTTP compression, are often available on SLB devices. This frees valuable compute cycles on the real servers. In the case of HTTP compression, there are also benefits of WAN bandwidth reduction and faster load times for the HTTP client.

It should be noted that many of the various techniques that lock down transactions between a client and a specific physical server can be used concurrently. For example, an optimizing intelligent DNS authoritative nameserver can be used with L7 redirects, and with local server load balancers.

6.1.7 Path Optimization

After the L3 endpoints are established, communication between the client and server becomes a matter of IP packets traversing a path between the two. However, there are typically multiple possible paths with many different attributes. Some paths might have high bandwidth and low latency, while other paths have little bandwidth but also provide low latency. The challenge of path optimization (implemented as a feature in Cisco IOS) is to manage traffic so that specific traffic classes are placed on the best network path.

Traditional routing is only responsible for announcing reachability to an L3 prefix. Reachability is based on the ability of the routers along the path to maintain adjacencies to each other. Therefore, even though the routing protocols advertise reachability for a prefix, actual reachability might not exist, or might even be degraded (for example, congestion along a path results in dropped packets). Traditional routing protocols also use static metrics for evaluating paths and do not account for changing network conditions. In practice, congestion, delay, and varying link utilization loads affect network application performance, even though regular routing does not take these network path attributes into account.

Because the WAN edge of a network is also a flow aggregation point, possibly a bandwidth bottleneck, and usually the last point of control in routing policy, the WAN edge becomes the natural place to make path selection decisions. Traffic at this point is already treated with optimization technologies, so the flow bandwidth observed by the WAN edge router is after optimization.

Generally, delay, bandwidth, and other types of issues are less prevalent and much easier to solve within a local branch or campus network. In a local network, it is relatively straightforward to add additional links or upgrade the bandwidth. However, on the WAN side, increasing bandwidth can add significant continuing costs, and there is little control over congestion and latency, especially in shared infrastructure networks, such as MPLS-VPN networks, and in cases where the networked sites are geographically dispersed.

In traditional packet forwarding systems, only a limited number of mechanisms are available to distribute traffic among links. Placing packets in round-robin fashion is not viable: different paths have varying latencies that cause out of order packets, eventually resulting in far lower performance for the network application.

Hash based mechanisms, for example, EtherChannel or Cisco Express Forwarding (CEF), were devised so that flows would be statistically distributed, based on mathematical functions, among different paths. However, the traffic rate for a flow is not considered, and it is possible for multiple high bandwidth flows to be sent on an already congested link while other links remain underutilized. Finally, link bonding technologies, such as Inverse Multiplexing over Asynchronous Transfer Mode (ATM-IMA) and Multilink Point to Point Protocol (MLPPP) try to slice packets so that they each packet is simultaneously sent on multiple links.

Link bonding technologies generally place a substantial load on the fragmenting device and the reassembly device, and are sensitive to intramember-link delay variation. Finally, the remote peer must be the same type of device, limiting the use of multiple WAN providers.

Figure 6-4 Path Optimization for Voice and Email Traffic

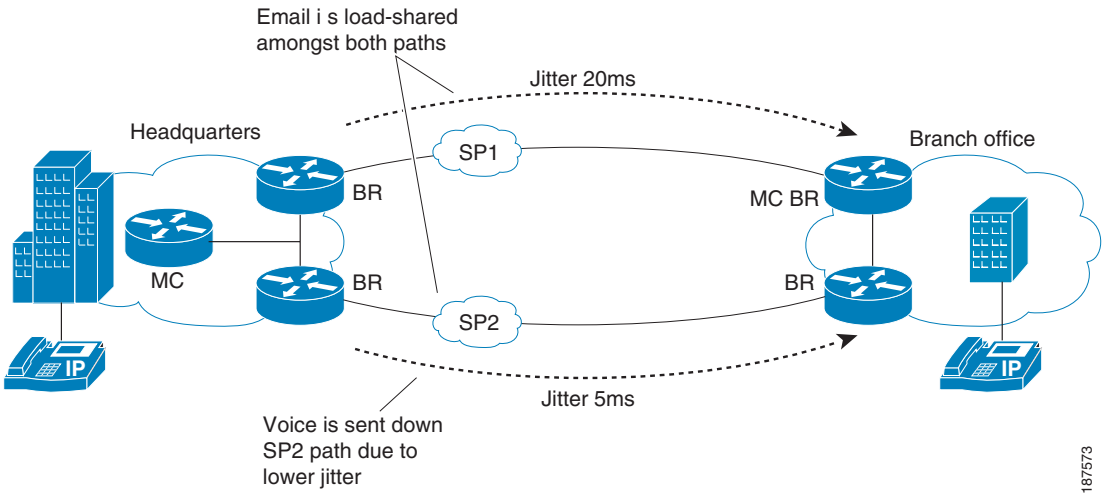


Figure 6-4 illustrates path optimization for different traffic classes. Voice traffic requires low latency; if a path provides acceptable minimum network latency compared to other paths, voice traffic is best placed on that low-latency path. However, other traffic classes, such as email, are not sensitive to latency, and could be placed on a more varied set of paths. Identifying traffic classes, measuring network attributes such as latency and packet loss, and placing traffic on the most appropriate path is the role of path optimization.

Path optimization can have varying levels of granularity, ranging from manipulating only the path to an L3 destination network, to per-flow manipulation and per-application path selection so that traffic uses the best available path. Using flow selection mechanisms and awareness of exit link utilization enables path optimization technologies, such as Cisco IOS Performance Routing (PfR), to distribute flow load across multiple paths on a per flow basis.

Using path optimization, the full aggregate bandwidth for a network site can be effectively used by performing link utilization based traffic load-sharing. Additionally, pockets of free bandwidth are created by the traffic distribution across all the links to support temporary data bursts (called microbursts) that exist in most real-world application profiles.

6.2 Layer 4 Optimizations

L4 optimizations include TCP stack optimization and payload compression.

6.2.1 TCP Stack Optimization

The previously described optimization technologies involved endpoint and path selection. Notably, these technologies do not change the of the IP traffic payload. However, there are opportunities for optimization at the TCP stack level, on the actual payload, and finally on exploiting behaviors and patterns at the network application.

There are many implementations of the TCP stack across operating systems and even among different versions of the same operating system. Not all the stacks are alike, and some perform better under certain conditions than others. Of course, there are also cases where TCP stacks are severely out of date and

have not been maintained. Over the years, there have been many advances in TCP technology, and only some advances have trickled down to implementations. TCP stack optimization uniformly applies the behavior of an advanced TCP stack across network applications.

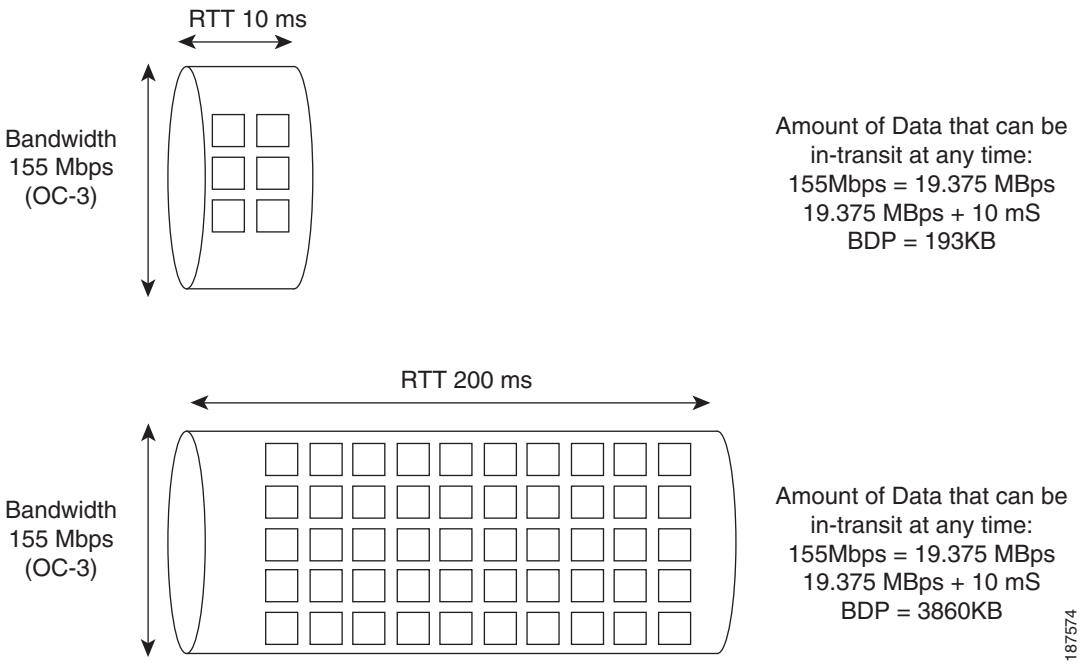
TCP is responsible for reliably transporting a message stream from one computer to another. As data is acknowledged by the remote host, succeeding data in the stream can be sent. This enables a network application to work on good in-order data without missing parts.

TCP controls how much data can be in transit and not acknowledged at any particular time, and this effectively controls the overall data transmission rate. The number of outstanding packets allowed without acknowledgement is known as the maximum window size (MWS). Over time, if there are no lost packets, the MWS expand to take more of the available bandwidth in the network path.

A network path, depending on its bandwidth and end-host to end-host latency, can hold a certain amount of traffic in transit, called bandwidth delay product (BDP). As shown in Figure 6-5, a network path with 155Mb/s of bandwidth and a round-trip time (RTT) of 10ms can hold 193KB in transit. Meanwhile another path (possibly on the same link, but to another host much further away) with the same bandwidth but with a 200ms RTT can hold 3860 KB in transit.

Because the TCP header used to specify window size contains two bytes, the MWS is only 65KB. The example in Figure 6-5 requires a much higher window size to pack the long network path full of data. RFC 1323 introduces a mechanism for scaling window sizes that supports much larger window sizes.

Figure 6-5 Comparing BDPs

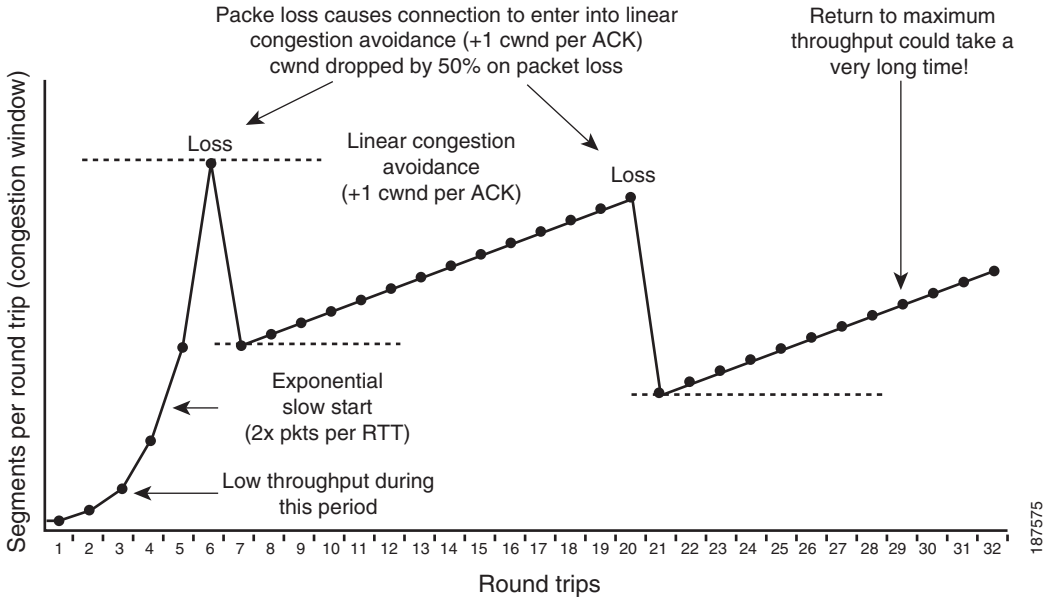


Because the capacity of the network path is unknown, starting at too high a rate (a large MWS) can cause severe congestion and packet loss not only for a specific session, but also for all sessions sharing the same path. For this reason, traditional TCP employs a technique known as slow-start that initiates the session with a small window that expands as good network behavior is observed. However, for higher latency links the RTT can slow such expansion so much that the bandwidth potential of the link is never realized. RFC 3390 supports larger initial windows so that a higher data rate is set when a connection is initiated.

When packets are lost, presumably because of congestion, traditional TCP aggressively reduces the MWS and slowly increases the data rate. This is known as congestion avoidance. For higher latency paths, it can take quite a long time to build to the proper data rate while causing minimal impact to other flows on the network path.

A multipacket loss event in traditional TCP can also result in unnecessary data retransmission or multiple RTT of delay). This is because in traditional TCP, only properly ordered packets are acknowledged. RFC 2018 introduced a mechanism to perform selective acknowledgements, and other mechanisms for advanced congestion avoidance are available. Figure 6-6 illustrates this.

Figure 6-6 Cumulative Traditional TCP Stack Delays and Underutilized Links



These are only some of the added capabilities of advanced TCP stacks that are often not available on a particular operating system version, disabled by default, or not implemented on any client-facing operating system. As TCP sessions traverse the network, they can be intercepted at an aggregation point and ”upgraded” with additional properties, regardless of the capabilities of actual end host. Cisco WAAS includes an intercepting TCP stack enhancer, called TCP Flow Optimization (TFO), which can apply the described TCP stack improvements to network flows.

6.2.2 Layer 4 Payload Compression

Data compression on limited bandwidth WAN links has been available for several years. Compression not only increases effective data transfer rates, but also reduces buffering, which can effectively reduce latency.

Data compression has been implemented at the lower data communications layers for a long time. For example, V.42bis on dialup modems and various schemes (RFC 1974, RFC 1967, and so on) used the Point-to-Point Protocol (PPP) Compression Control Protocol (CCP) to negotiate compression protocols.

In Frame Relay (FR), FRF.9 can compress data on a per-FR private virtual circuit (PVC). However, as these compression schemes are tied to the lower network layers, they are useful only for single L3 hops. Additionally, such compression technologies were limited by the size of the dictionary table used to store compressing patterns. The dictionary size had to be balanced against the practice of communicating over a transport system that did not reliably deliver data in order. A smaller dictionary size (usually limited to a packet) reduced the range of the data stream that could be referenced.

At the higher IP layers, there have been nondata payload compression schemes that act on TCP/IP headers, for example, RFC 1144, and compression schemes that act on specific application traffic, such as Real-time Transport Protocol (RTP), for example, RFC 2508). In practice, the techniques in these examples are effective only on extremely low bandwidth paths, such as T1and slower, where bandwidth savings constitute higher percentage of the overall WAN speed. Both RFC 1144 and RFC 2508 are limited to single hop L3.

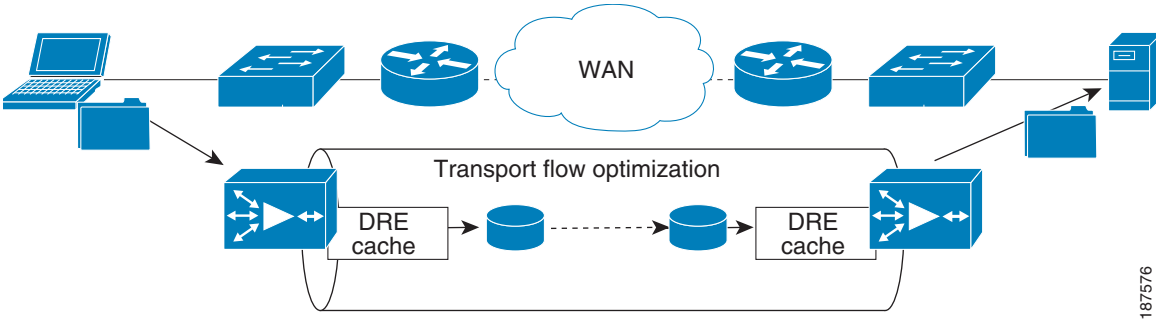
The IPComp protocol (RFC 3173) added a capability to compress the IP payload between two hosts or routers. Unlike the previously described compression methods, IPComp can take place over multiple L3 hops, so IPComp can be practically deployed in MPLS VPNs and in VPNs created over the Internet. However, IPComp still works over an unreliable transport and maintaining a large dictionary is a significant task. For combined IPSec and IPComp, the compression dictionary is limited to single packets and severely limits potential compression. In addition, in many deployment scenarios, traditional IPComp would require the creation of tunnels.

In recent years, TCP-based network compression has been developed to address many of these issues. TCP-based network compression combines large RAM based pattern matching dictionaries and large disks (several hundred gigabytes). Because TCP provides reliability, the compression dictionary and history are not limited to a packet and extremely high compression ratios can be achieved. This form of compression is different from object-caching mechanism (for example, HTTP caching). TCP-based network compression can work effectively when changes occur in the object.

For example, with TCP stream compression or data reduction, a change in a Word document would transfer only the changed content with compressed references to the rest of the content. In HTTP object-caching, the object must first be recognized (usually based on a name), and the entire object is transferred. Because TCP-based network compression is performed at L4, compression can occur over multiple router hops. This technology can be deployed in many kinds of networks, including MPLS VPN, Internet, and so on.

In a Cisco implementation, the WAAS product line, TCP data reduction technology is available in the Data Redundancy Elimination (DRE) function, which stores variable-length patterns on the WAAS hard drive. LZ compression is performed, after DRE (across multiple packets), for new patterns that DRE has not yet learned, or for micropatterns, as illustrated in Figure 6-7.

Figure 6-7 A WAAS Device Performing DRE and LZ Compression



6.3 Layer 7 Optimizations

The L4 based, TCP-based compression method described in the previous section does not examine the actual traffic payload. The method simply observes patterns in the data stream and performs a compression algorithm. In certain cases, however, when the context of the data and the protocol is well-understood, much higher bandwidth, message, and latency (because of fewer RTTs) reductions can be achieved.

Many network applications were first developed for, and deployed in, only high bandwidth, low-latency networks, such as a high-speed LAN. Networking assumptions that were built into these applications usually do not hold true in WAN deployments. L7 optimizations attempt to improve the performance of such network applications on the WAN. L7 optimizations use a variety of techniques, such as aggregating messages, caching, and advanced compression algorithms.

6.3.1 HTTP Compression

The HTTP/1.1 specification (RFC 2616) allows the use of various types of compression on HTTP payloads. This enables servers to directly send compressed data to a web browser. (Almost all browsers support such encoding.) A decompressing network node at the network edge is not required... While many web servers can perform this compression, web servers are usually better used for dynamic web page creation and interactions with backend servers.

The more mundane task of compression can be offloaded to network devices that can perform the compression in hardware without adding much additional latency. As in the TCP case, a compression dictionary and history can be highly effective within the scope of an HTTP session because references can be passed across multiple packets in the stream.

However, unlike DRE technology, the compression history in HTTP/1.1 applies only to one HTTP session. Downloads by other PCs at the same site do not see any improvement because the compressed content can only be used once. In the DRE case, traffic for multiple computers traffic passes through an L4 optimizer, which keeps a copy of the data patterns for future reuse and referencing. Therefore, HTTP compression is generally less effective than DRE.

6.3.2 Application Acceleration

There are many cases where the implementation of a network application or protocol does not work well over the bandwidth limited and higher latency WAN environments. Application acceleration is designed to understand a network application and reinvent its protocol, optimized for the WAN. This protocol reinvention can be very simple, such as simply reordering message payloads, or much more complicated.

A good candidate for application optimization is Common Internet File System (CIFS), which Microsoft developed for Windows network file services as a variation of the Server Message Block (SMB) protocol. Over time, many versions of CIFS have been developed. Each new version of the Windows operating system includes a slightly expanded CIFS vocabulary. Currently, there are over 120 commands across the various Windows versions. When many CIFS operations are initiated, several handshaking operations must be performed, such as file permissions checking, file locking, and so on. Because these operations depend on each other, they are done serially. Each operation incurs delay across the WAN.

The CIFS application acceleration technology implemented in WAAS tries to group CIFS messages and proxy them on the optimization node on the WAN egress. This reduces the effects of the round-trip times. Additionally, the benefits of DRE, LZ compression, and other TCP stack optimizations are performed on the CIFS transactions.

6.3.3 Prepositioning

Prepositioning content where it is more convenient to the user (from a networking perspective is a very common form of optimization. Several different technologies can perform prepositioning, but the benefit comes from topological closeness. The ability to host and serve the content with minimal user interaction is also important. Ensuring that prepositioned content remains “fresh,” that is, that prepositioned content matches the content at the originating server is essential.

The L3 End Point Optimization (EPO) technology makes very frequent use of prepositioning. EPO uses DNS to send a user to an HTTP cache that is topologically closer to the user, and that is already prepositioned with the content.

Windows file services can use local file-shares that are already prepositioned with documents, images and so on. WAAS provides prepositioning services as part of its CIFS framework.

Many audio/video files are extremely large. The Cisco Application and Content Networking System (ACNS) product supports prepositioning videos and audio files, such as training material, during off-peak times, and then using the content during peak times. This reduces the WAN load and provides users with nearly instantaneous access to the content.

6.3.4 Stream Splitting Technologies

Video and audio streaming have interesting properties with respect to optimization in that multiple parties are many times interested at the same time in the same set of data. Additionally, some applications such as stock tickers and pre-positioning are trying to deliver the same data set to a distributed population.

6.3.5 Multicast

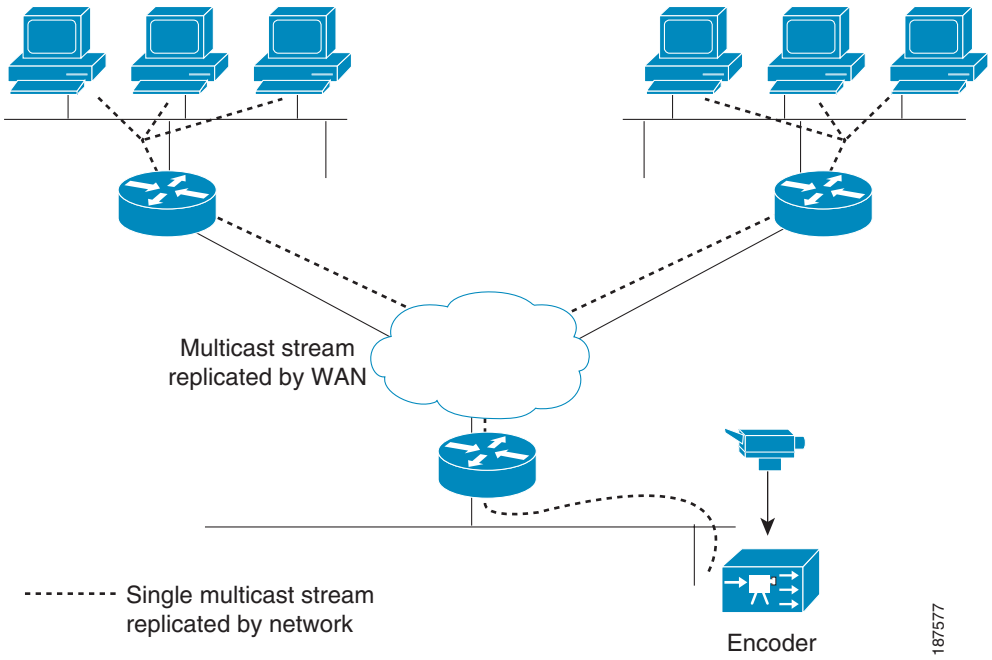
None of the technologies described so far require changing WAN functionality or features. In this sense, they work despite inefficiencies in the WAN network. Multicast technology is akin to a controlled broadcast in that data is transmitted once over a network, specifically to interested parties.

Looking at the L4 optimizations described in section 6.2 Layer 4 Optimizations, and at Figure 6-8, you can see that unicast streams are optimized at the branch level. That is, for multiple receivers at a branch, only one stream is sent to the branch. However, from the perspective of the stream source (the hub site), multiple identical copies are sent to each branch. If hundreds of branches are interested in the stream, there will be hundreds of duplicate data streams that will each need bandwidth on the WAN link at the hub.

Multicast can also be used with prepositioning. As described earlier in this chapter, prepositioning places the same data at points closer to its users. One way to update distributed preposition destinations is to use individual unicast streams and update each site one by one. If this is done concurrently, the individual streams use a lot of bandwidth on WAN link of the origin server. If done serially, the prepositioned update can take a long time to complete. With a multicast enabled WAN, however, the origin server can update the prepositioned sites concurrently, and WAN bandwidth is used more efficiently.

A multicast enabled WAN enables the highly efficient duplication of the data to multiple sites. If the WAN link from the hub and the WAN itself is multicast-enabled, only a single stream of data must be sent from the hub site. The WAN duplicates the flow as necessary along the path to the sites. This not only reduces the outgoing bandwidth use at the hub, it also optimizes bandwidth use in the WAN.

Figure 6-8 Multicast-Enabled WAN



Multicast can also be useful when the WAN is not fully multicast enabled, but is capable of multicast transport. Generally, in a WAN that is only capable of multicast transport, the WAN edge router must replicate the packets. To create such a network, generic routing encapsulation (GRE) tunnels can be deployed between the hub and branch sites, or, in a FR WAN, for example, each branch site can have a PVC to the hub.

This also means that neither the WAN nor the hub will see bandwidth savings for multisite traffic. However, having the WAN edge router perform the replication removes this burden from the server. The server does not need to expend resources for duplicating packets, and those server resources are free for other tasks.

6.3.6 Multicast Translation and Unicast Stream Splitting

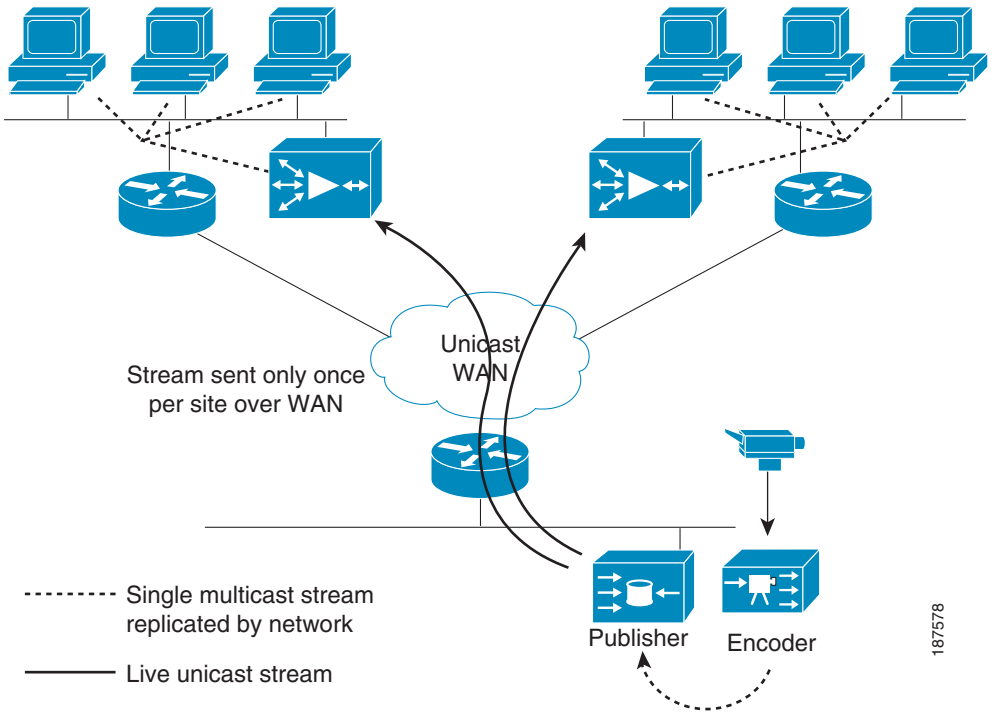
The one-to-many or many-to-many type of data distribution model maps perfectly to multicast. However, multicast is not enabled in many parts of enterprise networks, nor is multicast generally enabled in the Internet. This lack of multicast capability causes users in the same branch to receive the same audio or video stream from the streaming server, with each stream requiring its own bandwidth on the local branch, and on the WAN.

In this optimization case, the video/audio optimization device (Cisco ACNS) intercepts the streaming data request and only one request is sent over the WAN. The origin server therefore transmits only one stream to the site. After the data arrives from the WAN to the optimization device, the device can either multicast the stream locally to the interested parties, or replicate the unicast single stream of data into multiple unicast streams, once for each interested party.

This technique dramatically reduces the WAN bandwidth used by streaming video and audio. In fact, this technique might be required to enable multiple users to receive such streams; multiple receivers can easily use all the available bandwidth on the WAN access link.

This technology is also very useful in reducing the amount of streaming audio, such as Internet radio, that is received over an enterprise Internet access connection. It is possible; for example, that several hundred users are listening to the same Internet radio station and that each listener is eating up bandwidth on the Internet access link.

Figure 6-9 Optimizing Unicast Streams over the WAN



6.4 References

1. RFC 1035, *Domain Names - Implementation and Specification*, P. Mockapetris, 1987.
2. RFC 1144, *Compressing TCP/IP Headers*, V. Jacobson, 1999.
3. RFC 1546, *Host Anycasting Service*, C. Partridge, T. Mendez , and W. Milliken, 1993.
4. RFC 1950, *ZLIB Compressed Data Format Specification v. 3.3*, P. Deutsch and J-L. Gailly, 1996.
5. RFC 1962, *The PPP Compression Control Protocol (CCP)*, D. Rand, 1996.
6. RFC 1974, *PPP Stac LZS Compression Protocol*, R. Friend and W. Simpson, 1996.
7. RFC 2118, *Microsoft Point-To-Point Compression (MPPC) Protocol*, P. Singh, 1997.
8. RFC 2508, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*, S. Casner and V. Jacobson, 1999.
9. RFC 2616, *Hypertext Transfer Protocol - HTTP/1.1*, R. Fielding et al., 1999.
10. RFC 2782, A DNS RR for specifying the location of services (DNS SRV), A. Gulbrandsen, P. Vixie, and L. Esibov, 2000.
11. RFC 3173, IP Payload Compression Protocol (IPComp), A. Shacham et al., 2001.