Pocket Guide To Application Delivery Systems

Peter Sevcik and Rebecca Wetzel

Confused by the proliferation of products aimed at improving app performance? Here's an overview and taxonomy.

The roster of products that help applications perform well over private, virtual private and public wide area networks continues to grow, and so does the list of terms used to describe those products. Product introductions are routinely accompanied by new classification terms, and marketing brochures are replete with terms like application front end (AFE), Web application front end (WAFE), application delivery controller (ADC), wide area data services (WADS), WAN optimization controller (WOC), WAN acceleration device (WAD), and wide area file services (WAFS), just to name a few.

It's reached the point where you need a decoder ring to determine which products do what. Although differentiation can be positive, beyond a certain point differentiation becomes confusion—and we have passed that point.

To bring order to the nomenclature chaos, NetForecast has created the following taxonomy for application delivery systems (ADSs), as shown in Figure 1. At the apex of the taxonomy, we apply the term ADS to all the products—as well as the services—designed to help applications perform well over WANs.

We chose the word "application" because it is application performance after all, not network or other types of performance, that these offerings aim to improve. We use the word "delivery" because the offerings help the performance of applications delivered to the user over any network (including campus LANs, private WANs, virtual private networks (VPNs) and the public Internet). Finally, we use the word "system" because to work their magic, all ADS solutions must work as a pair of elements within the network or on the user's desktop. System also applies equally well to services and products.

Why Application Delivery Systems Are Needed

ADSs exist to counter forces that hamper application performance over WANs. Chief among these forces are: long distances, high turn counts, big payloads, insufficient bandwidth, network congestion and server bottlenecks.

The forces that hurt application performance are exacerbated by the unintended consequences of a number of common business initiatives, such as server centralization, application "Webification," increased inter-office collaboration and globalization. The resulting adverse effect on the user's experience can be so severe that users become frustrated and unproductive, putting these very business initiatives at risk.

Because business productivity depends on it, application performance measurement over a wide area network must reflect the user's experience—and the most useful measure of the user's experience is task response time. The business initiatives mentioned above adversely affect one or more factors that influence the user's task response time; the formula depicted in Figure 2 (p. 30) summarizes these performance-influencing factors and their consequences.

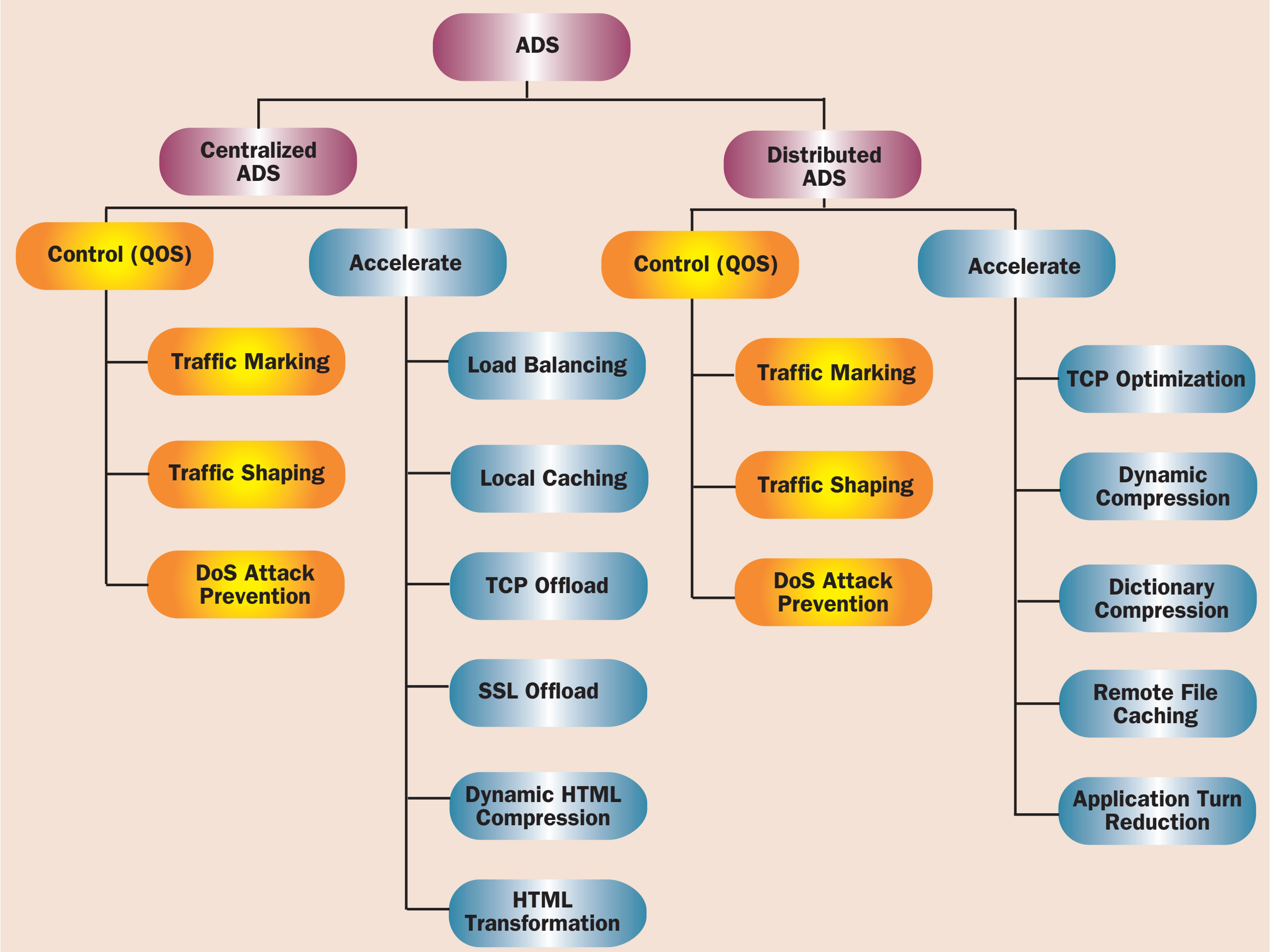
Only after identifying the factors responsible for increasing user response times is it possible to match the correct ADS solution type(s) to your particular set of performance problem causes.

Here are a few examples of the performance consequences of business initiatives:

- In server centralization, servers are moved from distributed offices to a central datacenter (usually for cost and control reasons), increasing the user-server distance and thus round trip times.
- Application "Webification" migrates applications to Web-based protocols and graphical user interfaces, generally increasing payload size and application turns in the process.
- Increased inter-office collaboration to tackle complex projects involving large files with dispersed rather than co-located employees often increases payload size, user-server distance, and server load.
- Finally, with globalization placing more employees and partners in more places, it is

Peter Sevcik is president of NetForecast and is a leading authority on Internet traffic, performance and technology. He can be reached at peter@netforecast.com. Rebecca Wetzel is an associate of NetForecast and a 20-year veteran of the data networking industry. She can be reached at rebecca@netforecast.com.

FIGURE 1 ADS Taxonomy



inevitable that some will be in areas served only by limited bandwidth connections.

Given the reality that business initiatives often challenge application performance, ADSs are helpful—and in many cases essential—to the success of the initiatives.

Centralized versus Distributed ADS Solutions

All ADS solutions use one of two approaches:

- A centralized or single-ended approach (also referred to as asymmetrical).
- A distributed or dual-ended approach (also referred to as symmetrical).

Centralized ADSs use a device in a datacenter near a server or server cluster. The device intercepts traffic passing to and from the server(s), and directs and/or modifies this traffic. Modifications to intercepted server traffic must be understood on the user's end, so the datacenter device must communicate with client software that makes sense of the modifications. The user's browser serves as the most ubiquitous standard client; therefore, at present centralized ADS solutions are limited to Web-based applications.

Distributed ADS solutions rely on a device in the datacenter and companion devices in remote offices. These devices are placed near WAN ingress/egress points where they can see, prioritize and modify traffic. Because distributed ADS solutions require access to the remote office, they are limited to private or virtual private networks (VPNs). In the case of telecommuting or mobile workers, distributed ADS vendors sometimes supply the "remote device" as software installed on the user's PC.

A critical difference between these two approaches is where and how they can be applied, as shown in Figure 3 (p. 31).

Another important aspect of the two approaches is that the centralized approach is inherently open and interoperable, while distributed solutions are closed and vendor specific. You can buy centralized ADS solutions from vendors A and B as long as they operate in front of different application servers. The users will continue to use the same browser to access all "enhanced" applications.

However, if you buy a distributed ADS solution from vendors D and E, they will both have to be

FIGURE 2 Formula For Performance-Influencing Factors

$$R \approx \frac{\text{Payload}}{\text{Bandwidth}} + \text{AppTurns(RTT)} + C_s + C_c$$

Where:

Bandwidth is the minimal bandwidth (bits per second) across all the network links between the user and the application server. The slowest link is typically the user's access line to the network. Useable bandwidth of the link may be reduced by the effects of conflicting traffic (congestion) and protocol efficiency (e.g., TCP window).

Cc (Compute Client) is the total processing time (seconds) required by the client device.

Cs (Compute Server) is the total processing time (seconds) required by the server(s).

Payload is information content (bytes) that must be delivered to/from the user's device.

R is the response time, defined as the elapsed time (seconds) between a user action (e.g. mouse click, enter, return) and the system response (client, network, server), so the user can proceed with the process. The aggregation of these individual task completion waiting periods defines the "responsiveness" of the application to the user.

RTT is the round-trip-time (seconds) between the user and the application server.

AppTurns are the application client-server software interactions (turn count) needed to generate a user-level system response or task (see above). Turns are separate from and in addition to TCP two-way interactions (e.g., open, close, ACKs). The user is not aware of turns.

installed in all locations. Furthermore, some features of D may adversely affect the work of E. Operating two different distributed solutions is tricky, and often they work as "ships in the night," ignoring each other.

The bottom line is that you can be a multi-vendor centralized ADS shop, but you will typically be forced to adopt a single-vendor distributed ADS solution.

Control And Acceleration Functions

Both centralized and distributed ADS solutions have two primary performance functions—they can control application performance over a WAN and/or they can accelerate it. Increasingly, these two performance functions are being combined into single solutions.

Control solutions protect application performance from degrading, by using techniques such as traffic marking, traffic shaping and denial-of-service attack prevention to help ensure quality of service (QOS). Control devices maintain existing performance under adverse network conditions by managing and allocating access to bandwidth or server resources by application or user.

They also protect against malicious users by offloading illegitimate or non-critical traffic from the server. A goal of ADS control solutions is to manage network and/or server resources for optimal business value.

It is important to note that because ADS control solutions are designed to protect but not improve performance, they do not speed response time when there is no congestion.

In contrast, acceleration solutions speed applications by reducing payload, shortening round-trip times, using bandwidth more effectively, reducing turns and/or offloading the server, thus improving application performance for all users all the time. Acceleration techniques change how an application behaves over a WAN to make it faster. Acceleration solutions speed up applications even when there is no congestion on the network, and some also offload some critical traffic from the datacenter.

A word of caution is needed about using acceleration techniques without also deploying control solutions. Acceleration in the absence of control is not recommended because performance for accelerated applications can still deteriorate badly under adverse network conditions.

Control Techniques

The following control techniques are used by both centralized and distributed ADS solutions. Many implementations are standards-based so they can interoperate with other elements of the system, such as routers or MPLS services. However, distributed control solutions often employ proprietary policy and management techniques that force single vendor deployments.

Traffic Marking provides information to downstream devices regarding how to handle different application traffic types. There are several marking standards, including IEEE 802.1p/802.1q, Type of Service (TOS), and Differentiated Services (DiffServ) Codepoints.

Traffic Shaping applies policies and priorities to different traffic types to ensure that the performance of critical applications is protected during network congestion. Several packet prioritization and queue management techniques can be brought into play, including Weighted Fair Queuing (WFQ), Class Based Queuing (CBQ), Random Early Discard (RED) and DiffServ, as well as a host of proprietary techniques.

Denial of service (DoS) Attack Prevention detects and blocks malicious attempts to tie up server or bandwidth resources, to make sure those resources are available for legitimate users.

Acceleration Techniques

Acceleration techniques differ between Centralized and Distributed ADS solutions. Following are descriptions for the major acceleration techniques used for each approach:

Centralized ADS Acceleration Techniques

Load Balancing distributes requests to different nodes within a cluster of servers, thus optimizing system performance and increasing availability and scalability. This addresses the problems of insufficient server resources and server congestion.

Local Web Caching, also known as reverse proxy caching, supports HTML content. The cache sits in the datacenter near the server. Client requests for Web content are transparently routed to a proxy server, which returns requested objects either from its cache or after fetching the objects from the content server. Local caching reduces the load on the Web server, thus speeding the server compute time.

TCP Offload funnels traffic from many connections into a single persistent TCP connection in the server. This saves server CPU processing or context switching time.

SSL Offload terminates each user's SSL session in an appliance and provides the data to the server in the clear. This saves server CPU processing and context switching, and saves running the SSL encryption algorithm. It also simplifies global key management across many servers.

Dynamic HTML Compression accelerates traffic by reducing the payload using an open compression standard called GZIP. Dynamic HTML compression is similar to "zipping" a file. It provides a powerful benefit to performance because, unlike images (e.g., GIF, JPEG files) that are already compressed, HTML is just ASCII text, which is highly compressible.

HTML Transformation addresses the fact that most websites are built without concern for performance and therefore perform poorly over a WAN. HTML transformation dynamically corrects for poor design by instructing the browser to retrieve the content in a new way. For example, many individual requests for page elements are integrated into a single request, lowering turn count. HTML content is also compressed, lowering payload, and HTML transformation makes better use of the user's desktop cache.

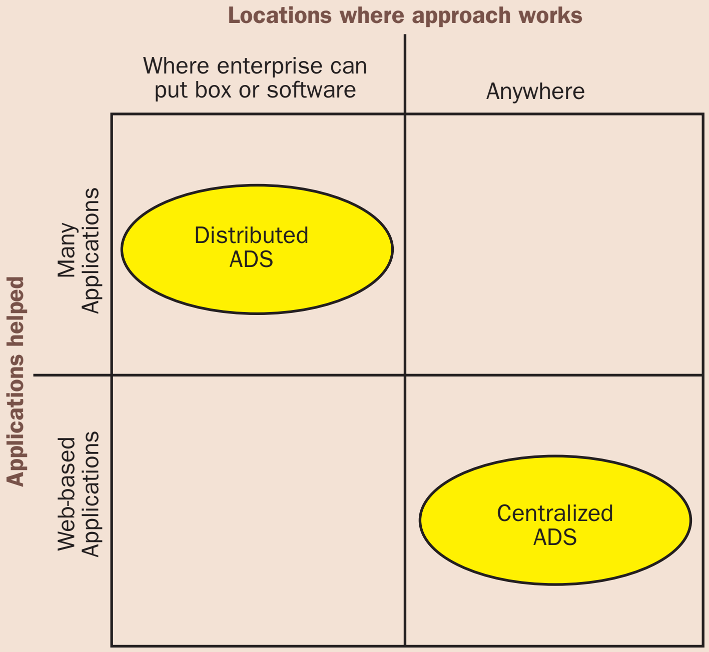
Distributed ADS Acceleration Techniques

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

The effect of TCP optimization is to increase throughput by matching the transmission rate to a constrained bandwidth access line, or overriding TCP window limits on high bandwidth but long latency paths. Most ADS devices implement proprietary variations or extensions to the "standard" techniques described above, making them incompatible with other vendor implementations.

Dynamic Compression is applied to data "on the fly" to reduce payload. Dynamic compression generally includes packet-level payload compression, compression of TCP and compression of elements larger than a packet, such as a window's

FIGURE 3 Optimal Use Of Different Approaches



Remote file caching can be done via either a "push" or a "pull" approach

FIGURE 4 Centralized ADS Techniques

	Control		Accelerate						
Protocols Affected	Traffic Marking	Traffic Shaping	DoS Attack Prevention	Load Balancing	Local Caching	TCP Offload	SSL Offload	Dynamic Compression	HTML Transformation
NTFS, DFS, NFS, etc.									
MAPI									
CIFS									
HTML	○	○	●	●	●	○	○	●	●
HTTPS (SSL)	○	○	●	●		○	●		
HTTP	○	○	●	●		○			
TCP	●	●	●	●		●			
UDP real-time VOIP or IPVideo									
UDP Streaming audio or video									
IP	●[1]	●[1]	●[1]						

Directly helps	●
Indirectly helps higher protocol	○

Notes:
1, only controls the traffic to/from the server(s) associated with ADS

FIGURE 5 Distributed ADS Techniques

	Control		Accelerate					
Protocols Affected	Traffic Marking	Traffic Shaping	DoS Attack Prevention	TCP Optimization	Dynamic Compression	Dictionary Compression	Remote Caching	App Turn Reduction
NTFS, DFS, NFS, etc.	○	○	○	○	○	○	●	
MAPI	○	○	○	○	○	○		●
CIFS	○	○	○	○	○	○	●	●
HTML	○	○	○	○	○	○	●	●
HTTPS (SSL)	○	○	○	○	○[2]	○[2]		●[2]
HTTP	○	○	○	○	○	○		●
TCP	●	●	●	●	●	●		
UDP real-time VOIP or IPVideo	●	●	●					
UDP Streaming audio or video	●	●	●				●	
IP	●	●	●					

Directly helps	●
Indirectly helps higher protocol	○

Notes:
2, only works if ADS can decrypt-recrypt the payload

worth of data. The techniques used are often proprietary variations on the GZIP method.

Dictionary Compression can be viewed as caching on an arbitrary data segment size, and the effect is to reduce payload. The system watches bytes go by and determines if a chunk of data referred to as a segment can be tagged. The segments have no relationship to a file or file name. Some, few or many segments can equal a file, and some, few or many files can equal a segment.

The first time data comes through the source node, the system detects patterns (segments) in the data (payload), and the segments are tagged with reference numbers of typically 16 to 34 bytes depending on the implementation. Some systems use a hierarchical reference number. The system then transmits the segment and reference number to the destination node.

The second time the data passes through the source node, the system determines if the data had been sent before. If it has not, the system sends the original data plus a new reference number for that data. If it has, the system sends only the reference number.

At the destination, the system stores the segments with the reference numbers in a protocol- and application-independent form. After receiving a reference number it recognizes from the source, the system injects the segment into the traffic stream.

Remote File Caching is the oldest acceleration technique, and its effect is to offload the server, reduce round-trip time and reduce payload. It often operates on servers as well as desktops, and is typically a "pull" solution, with the cache populated with files that traverse the appliance. When it detects a unique file name, it stores the file and name (with some systems adding a hash of the file to determine if the file has changed). When the source node sees that the origin is starting to send the same file again, it notifies the destination to deliver the file named "X" that it already has. Some systems check first to see if the file has changed before notifying the destination server.

Many systems also use a "push" approach (often called virtual file storage). In this case, the system may, for example, send all the files associated with an office to that office ahead of time (typically overnight). Some solutions also have sophisticated distributed file management systems. Vendor implementations are limited to a specific file family such as Windows NTFS, Windows DFS, Sun NFS, Linux FHS, etc. Any given implementation will only deal with one of these file families, which is usually fine because the enterprise has standardized on one type of client. Some file families (e.g., Windows) have more than one implementation (e.g., NTFS or DFS) but again the individual enterprise has standardized on one of these implementations.

Application Turn Reduction limits the application turn count by gathering most content into a single transaction over long network distances. The effect in the performance equation is to reduce the turn count. The system processes the application logic by intercepting the original client-server transmissions, interpreting the original payload locally, determining what the client and server are trying to do, doing it locally and thus more quickly on the LAN and retransmitting all the content in a single block. This typically reduces many WAN turns to one, and this single block is usually speeded to its destination using optimized TCP or a proprietary transport protocol.

Application turn reduction predicts transactions based on past behavior, reconstructing the application-level interactions on both the client and server ends, and it preserves client-server protocol semantics.

This technique is only applicable to protocols or applications that the vendor has decoded such that the devices understand the application logic. Application turn reduction should not be confused with TCP turn reduction, which occurs as part of TCP optimization.

Distributed ADS techniques cover a broad range of products

Which ADS Techniques Help Which Protocols

Centralized and distributed ADS techniques apply to different protocol sets. Figures 4 and 5 show that the centralized ADS techniques are designed to help Web-based protocols, whereas distributed ADS techniques cover a broader range of protocols.

Conclusions

This pocket guide provides an overview of the types of solutions designed to help applications perform well over WANs, and it specifies what protocols each solution type can support. For a more comprehensive view we suggest you read NetForecast's complete Field Guide to Application Delivery Systems, available at no charge at www.netforecast.com□