Field Guide to Application Delivery Systems

By Peter Sevcik and Rebecca Wetzel
September 2006

Navigating the labyrinth of products designed to help applications perform well over private, virtual private, and public wide area networks (WANs) becomes more challenging as product choices proliferate. Product introductions routinely add new nomenclature to an already confusing mix, making it hard to identify which solutions solve what problems.

Citrix Systems, Juniper Networks, Packeteer, Radware, and Riverbed Technology are pleased to present this field guide to help alleviate the confusion. The guide defines the major causes of WAN performance problems, describes the prevailing solution techniques using common nomenclature, and maps the techniques to problems so readers can select the right solution set.

To identify the Application Delivery System (ADS) techniques supported by specific vendors, refer to the *ADS Vendor Matrix* at www.netforecast.com.

Report Sponsors

NetForecast Report 5085

Table of Contents

Application Delivery System Taxonomy

To bring order to the perplexing array of product offerings, NetForecast has created a taxonomy for Application Delivery Systems (ADSs) shown in Figure 1. At the apex of the taxonomy we apply the term ADS to all the products and services designed to help applications perform well over WANs.

Here is the logic behind selecting the term ADS. We chose the word ***application*** because it is application performance after all, not network or other types of performance that these offerings help.

We use the word ***delivery*** because the offerings help performance of applications “delivered” to the user over any network (including campus LANs, private WANs, virtual private networks (VPNs), and the public Internet). The user transparently communicates with an ADS delivery element that serves as an intermediary for the origin application server and back-end system.

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.

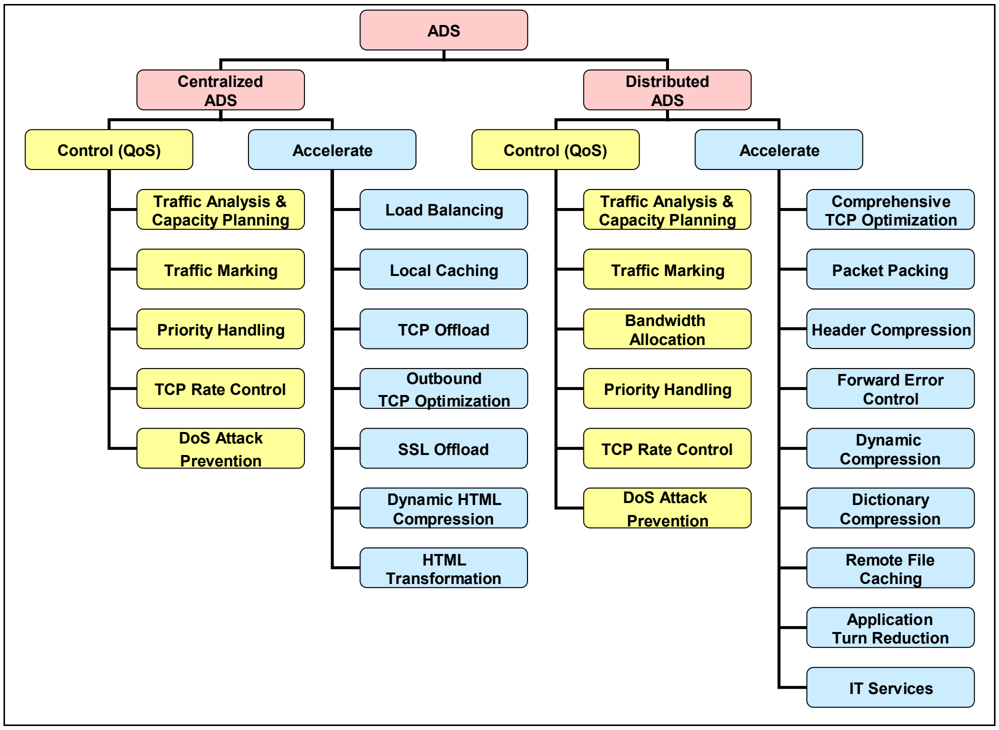


Figure 1 – Application Delivery System Taxonomy

What Causes Poor WAN Application Performance?

Before jumping in to describe techniques that make applications perform better over a WAN, it is important to understand what can cause WAN-connected applications to perform poorly in the first place. There are six key causes of poor performance: long distances, high turn counts, big payloads, insufficient bandwidth, network congestion, and server bottlenecks.

Long Distances

Distance hurts performance because the further a user is from the application server, the longer it takes packets to traverse the distance. This is sometimes referred to as the "speed of light problem". The time added due to distance (a.k.a. latency) can add up, especially if an application requires many round trips to complete a task.

High Application Turn Counts

Latency due to distance may go unnoticed for applications with few client-server software interactions (or turns), but it poses a problem for applications requiring many interactions—such as those based on "chatty" application protocols like Microsoft's Common Internet File System (CIFS) file sharing protocol, and Outlook that uses Microsoft's Messaging Applications Programming Interface (MAPI) email protocol for Exchange. Each additional turn requires packets to traverse the user-to-server distance twice, once in each direction.

Big Payloads

Big payloads such as those due to large file transfers and complex Web pages can slow end-user response times as they consume bandwidth over constrained links.

Insufficient Bandwidth

Insufficient bandwidth along the path between the user and the application server slows user response times. Bandwidth is typically in shortest supply on the user's network access line, making this "last mile" connection the most frequent culprit for insufficient bandwidth.

Network Congestion

Network congestion can stem from a variety of sources to consume available bandwidth and slow user response times. Common congestion sources are high bandwidth applications such as streaming audio and video, or "flash crowds" caused by surges of users simultaneously vying for the same limited bandwidth resources.

Server Bottlenecks

Insufficient server computing resources and/or insufficient server connections also can lengthen application response times.

The Big Performance Picture

Poor application performance renders users unproductive—therefore, application performance measurement over a WAN must reflect the user's experience. The most useful measure of the user's experience is task response time.

Performance Cause and Effect

Each problem described above adversely affects one or more of the factors that influence task response time. The following formula summarizes these performance-influencing factors. Only by identifying the factors responsible for increasing user response times to unpalatable levels can you be sure to match the right solution with the right problem.

$$R \approx \frac{Payload}{Bandwidth} + AppTurns(RTT) + Cs + Cc$$

Where:

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

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 link bandwidth 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, which is 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 application "responsiveness" perceived by the user.

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

Application Profiles Affect Performance

It is a common misconception that application characteristics do not contribute as greatly to long user response times as network characteristics. Examining the profiles for many well known "modern" applications shown in Figure 2 puts that misconception to rest. Many widely-used applications require hundreds of turns and kilobytes of payload to complete each user task—and application profiles are worsening over time.

Note that the chart uses a log-log scale so every bar shows a ten-fold increase.

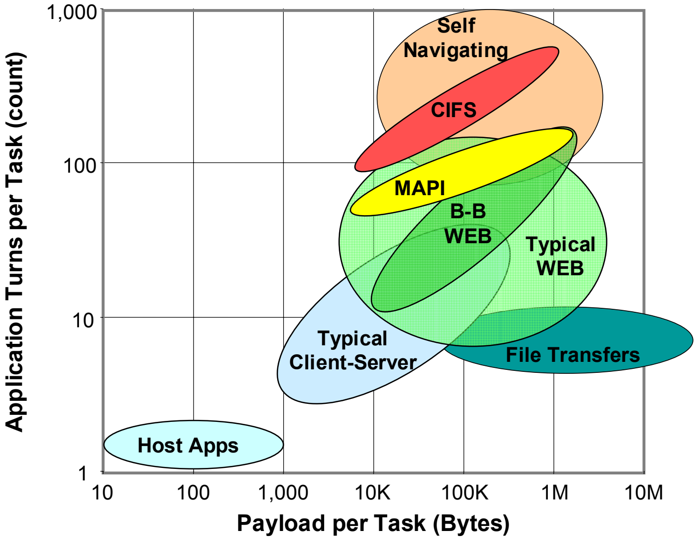


Figure 2 – Application Profiles

NetForecast tracks the profiles of typical business-to-business Web sites using the Keynote Business 40 Index as the proxy for such sites. The Keynote Business 40 Index is a representative group of business Web sites selected by Keynote Systems. Over the past decade turn counts for these Web sites have grown 12% per year and payload has grown 20% per year. Today the average Keynote Business 40 home page requires 60 turns and 300 kilobytes to load.

Business Initiatives and Performance

A number of common business initiatives are thrusting application performance into the spotlight because applications must perform well over the WAN for the initiatives to flourish. Four such initiatives: server centralization, application "webification", increased inter-office collaboration, and globalization advance business interests, but pose daunting application performance challenges.

Server Centralization

The need to manage information more efficiently and to conform with such regulations as the US Sarbanes-Oxley Act and Basel II, are driving servers from distributed to centralized locations. Migrating server access from a local area network to a WAN poses performance challenges because it increases user-to-server distance, it lowers available bandwidth compared to a LAN, and it increases WAN traffic.

Application "Webification"

Business pressures are driving enterprises to migrate applications to Web-based graphical user interfaces and Web-specific protocols like HTTP, HTML, XML and SOAP to make the applications accessible using open standards. An example of this trend is widespread migration from SAP's proprietary R/3 graphical user interface to the Web-enabled mySAP interface. This application "webification" often has two adverse performance effects—as it does for SAP—it enlarges payload size, and it increases the number of application turns.

Increased Inter-office Collaboration

Competitive forces are driving enterprises to make smarter use of distributed workforces by promoting collaboration across offices. Design firms for example are increasingly assigning employees in different offices around the world to the same design projects, whereas in the past only co-located employees collaborated so closely. Such inter-office collaboration increases distance to users, it often increases payload size across the WAN with the exchange of large files, and it can burden application servers.

Globalization

The need to keep pace with the relentless interconnection, expansion, and interdependence of global markets is driving enterprises to become increasingly global. Not only does globalization challenge application performance by increasing user-to-server distance, it also shifts traffic from high-functioning corporate networks to the less dependable Internet, and only limited bandwidth may available for some users.

The following figure shows how performance problems caused by business initiatives most directly influence factors in the performance formula to raise response times for WAN-connected applications.

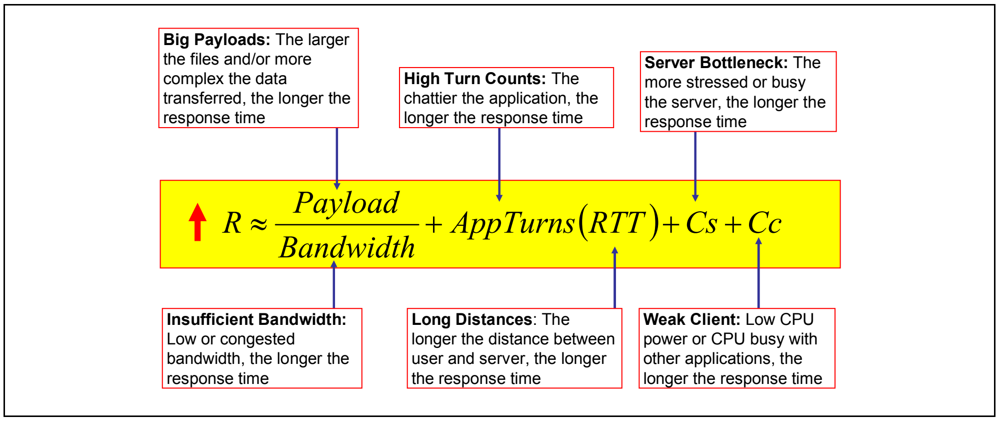

Big Payloads: The larger the files and/or more complex the data transferred, the longer the response time
High Turn Counts: The chattier the application, the longer the response time
Server Bottleneck: The more stressed or busy the server, the longer the response time
$R \approx \frac{Payload}{Bandwidth} + AppTurns(RTT) + Cs + Cc$
Insufficient Bandwidth: Low or congested bandwidth, the longer the response time
Long Distances: The longer the distance between user and server, the longer the response time
Weak Client: Low CPU power or CPU busy with other applications, the longer the response time

Figure 3 – Response Time Causes and Effects

WAN Application Performance Solution Overview

WAN application performance solutions employ two basic approaches, a single-ended approach, and a dual-ended approach—and solutions perform two functions, they control and/or accelerate application performance. In this guide we refer to single-ended solutions as Centralized Application Delivery Systems (Centralized ADSs), and dual-ended systems as Distributed Application Delivery Systems (Distributed ADSs). [Note that some Distributed ADS systems can work alone, but that is the exception not the rule. Read on for more explanation of this]

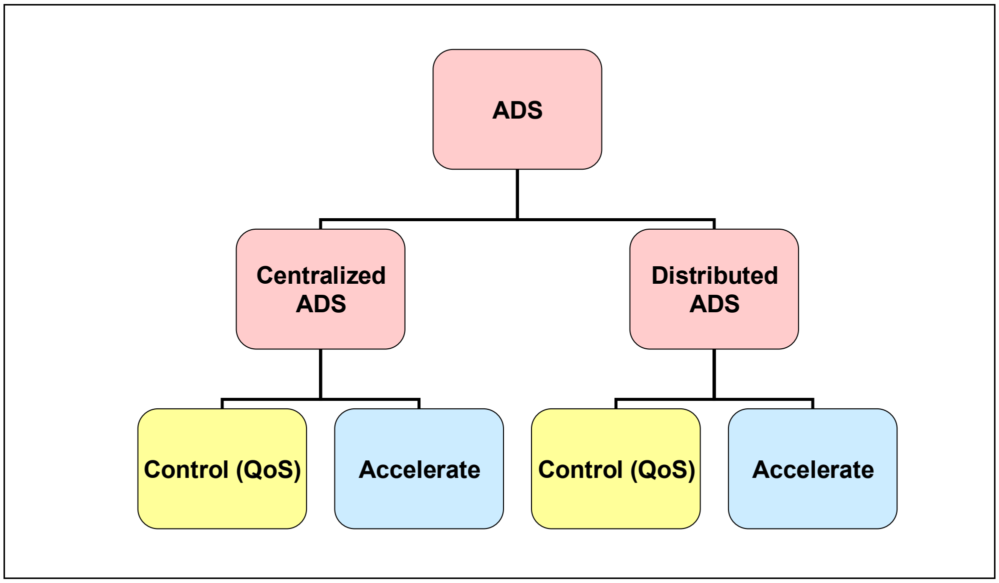


Figure 4 – ADS Approaches and Functions

Centralized versus Distributed ADS Solutions

Centralized ADS solutions employ a device in a data center near a server or server cluster. The device intercepts traffic passing to and from the server(s), and directs and/or modifies the intercepted traffic. Modifications to intercepted server traffic must be understood on the user's end, so the data center 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 typically deployed to deliver to Web-based applications.

Some Centralized ADS vendors provide proprietary software clients that can further accelerate Web applications as well as optimize non-Web applications. These clients provide a cost-effective alternative to Distributed ADS solutions in home offices or "micro branches".

Distributed ADS solutions rely on a device in the data center 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. 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 5.

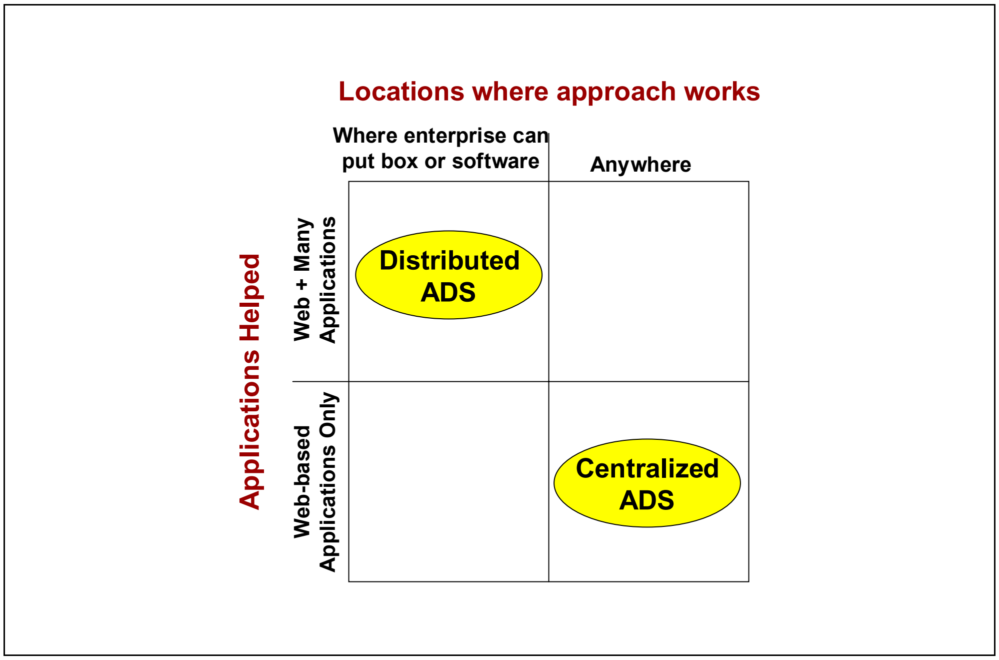

Locations where approach works
Where enterprise can put box or software
Anywhere
Applications Helped
Web + Many Applications
Web-based Applications Only
Distributed ADS
Centralized ADS

Figure 5 – The Centralized-Distributed Divide

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 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 not interoperate. Thus to experience the full benefit of solutions D and E, you must install both in all locations, which can be costly. Furthermore, some features of D may adversely affect the work of E. Operating two different distributed solutions is tricky, and at best 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.

Centralized/Distributed Scope and Span

The technical functions that Centralized and Distributed ADS solutions perform are often very similar—even identical. But as Figure 6 shows there are major differences in the types of traffic that benefit and the span (or reach) of the two solution types.

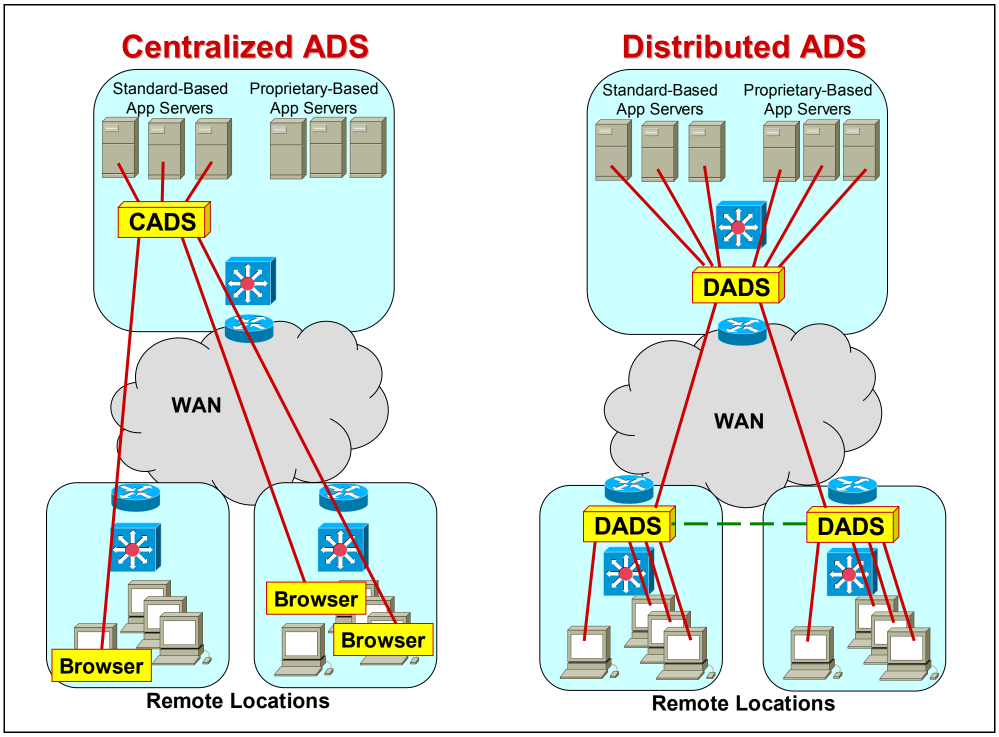

Centralized ADS
Standard-Based App Servers
Proprietary-Based App Servers
CADS
WAN
Browser
Browser
Browser
Remote Locations
Distributed ADS
Standard-Based App Servers
Proprietary-Based App Servers
DADS
WAN
DADS
DADS
Remote Locations

Figure 6 – Scope and Span of Centralized versus Distributed Solutions

The red lines in Figure 6 show the span of influence for Centralized and Distributed ADS techniques. In both approaches control and acceleration techniques can only be applied to traffic the devices directly process (in-line) or monitor (through a switch-mirrored port).

Centralized ADS solutions support the specific servers they front, and can only influence applications based on open standards such as Web (HTTP, HTML, Java), XML—and more recently VoIP (SIP) and Video over IP (H.323). Despite the growth of these standards-based applications, a recent NetForecast survey indicates that Web-based traffic represents only about one quarter and real-time traffic one fifth of a typical enterprise's network load[1]. This leaves just over half the traffic in the typical enterprise untouchable by Centralized ADS solutions.

Centralized ADS solutions exert influence over traffic traveling outbound across the WAN to Web browsers, and inbound across a LAN to the servers. Centralized ADS solutions cannot influence inbound WAN traffic because that is the realm of the browser.

Distributed ADS solutions, on the other hand, exert influence over outbound as well as inbound traffic traversing the WAN-LAN boundary at the data center and at remote sites. Therefore, these solutions can affect all WAN traffic (including Web, traditional client-

[1] "IT Managers Weigh In On Application Performance," Peter Sevcik and Rebecca Wetzel, *Business Communications Review*, April 2006

server, UDP, and VoIP), but they cannot influence traffic that remains within the data center or within remote sites.

The dual-ended nature of most Distributed ADS implementations enables stronger versions of the same performance techniques used in Centralized ADS solutions. For example, distributed TCP optimization can leverage six separate TCP flows (two between the server and Distributed ADS, two between the Distributed ADS devices, and two between the remote device and the client), while centralized TCP optimization can only influence three TCP flows (two between the server and the Centralized ADS device, and one outbound from device to the browser).

Finally, as the dashed green line in Figure 6 illustrates, some Distributed ADS solutions can leverage peer-to-peer information exchange among remote devices and/or share a database of knowledge at the data center.

Distributed ADS solutions are typically dual-ended or synchronous, which means they operate as a pair as shown in Figure 6. However, some Distributed ADS techniques—primarily control functions—can also operate alone. For example, a single distributed device is often used to manage college campus traffic to and from the Internet to prevent music and video downloads from swamping other applications. Although single-ended distributed implementations exist, this report focuses on the more ubiquitous dual-ended solutions.

In contrast to Distributed ADS solutions, all Centralized ADS solutions are single-ended or asynchronous. This means they operate as standalone devices that do not rely on communications with peers.

Control

Modifies how packets behave, thus changing the way the network looks to the application.

Control

ADS control solutions protect application performance from degrading. Control devices maintain existing performance under adverse network conditions by fixing, mitigating, or avoiding performance incidents resulting from insufficient bandwidth or server resource constraints. 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 during periods of no congestion.

Acceleration

Modifies how an application behaves, thus changing the way the application looks to the network.

Acceleration

ADS acceleration solutions speed applications and thus improve application performance for all users all of the time. Acceleration techniques change how an application behaves over a WAN to make it faster. The devices accelerate applications even when there is no congestion. Some acceleration solutions also offload some critical traffic from the data center.

A word of caution is warranted about using acceleration techniques without also deploying control solutions. Applying acceleration in the absence of control is not recommended because despite the benefits of acceleration, performance for accelerated applications can still deteriorate badly under adverse network conditions. Let’s consider an example to illustrate why control should accompany acceleration.

The response time for a mission-critical application is unacceptable at a branch office experiencing a mere 30 percent remote access line utilization. Control techniques cannot improve the poor performance because there is no congestion, so prioritizing one application over another doesn’t help. Deploying an acceleration technique solves the performance problem and users are happy until a server at the remote site fires up a back-up process during the work day boosting line utilization to 80 percent. The mission-critical application slows to a crawl despite the acceleration technique and users become irate. Had control techniques been in place, users would likely never know there was a congestion problem because control ensures the “headroom” that acceleration requires.

Here is a good way to think about how control and acceleration work. Control modifies how packets behave and thus changes the way the network looks to the application, while acceleration modifies how applications behave and thus changes the way the application looks to the network.

Centralized ADS Techniques Defined

Here is a lineup of the primary control and acceleration techniques employed in Centralized ADS solutions, and a brief definition for each technique.

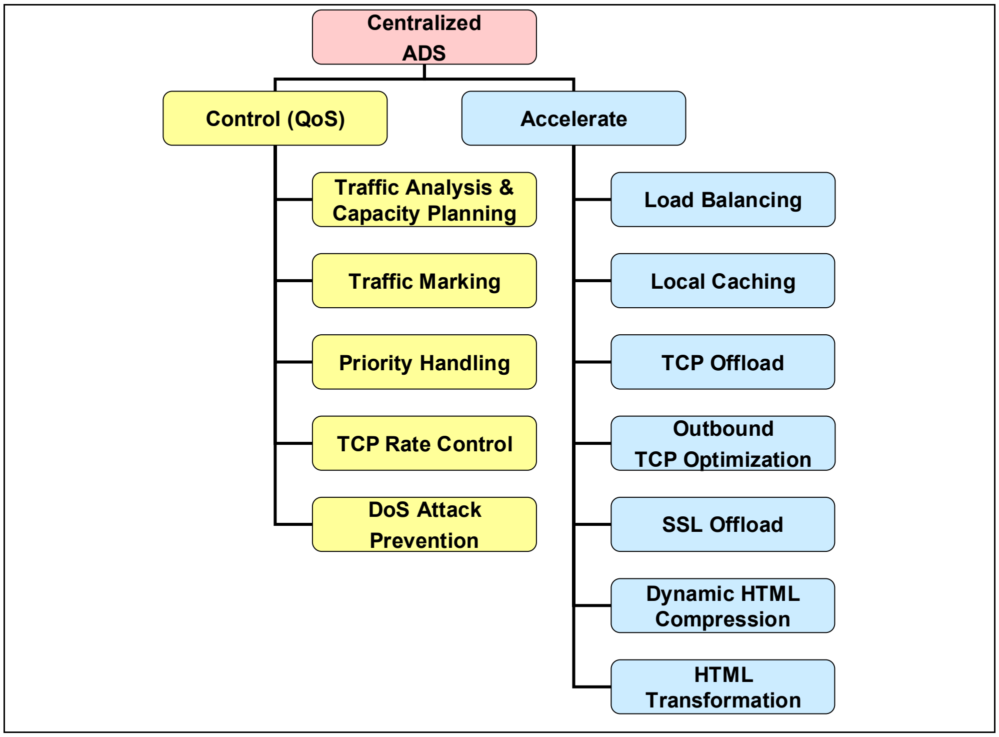


Figure 7 – Centralized ADS Solutions

Control

Traffic Analysis and Capacity Planning monitor and report on the Web server and server-side connection health to help determine quality of service policies, properly size resources, and identify and correct problems that affect performance.

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. The markings are interpreted by down-stream devices and systems.

Priority Handling delivers traffic in a specified order of priority depending on the application. Centralized ADS priority handling depends on the relative importance assigned to traffic classes, not the traffic marking described above. Typical importance classes are the application, the user group, and the session context (e.g. in a retail setting customer checkout would be assigned more importance than browsing). The equivalent of “traffic marking” is often performed by Web session cookies. The cookies can be linked to the user and changed based on context.

TCP Rate Control applies proprietary techniques that buffer and then pace packets into downstream systems. Centralized TCP rate control is applied to the outbound (Centralized ADS-to-Browser) TCP flow. The Centralized ADS device can gather outbound TCP payload from the server, buffer the payload, and then pace the rate at which it transmits the same packets it received from the server. The TCP connection need not be terminated in the ADS device.

The rate control algorithm determines the highest effective achievable bandwidth, and paces the traffic at a rate just below the maximum effective bandwidth. This eliminates the too-fast-too-slow saw tooth pattern of typical TCP connections that have constricted bandwidth along the path. The technique is most effective during the transfer of large files.

DoS Attack Prevention detects and blocks malicious attempts to tie up server resources so those resources are available for legitimate users.

Acceleration

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 multiplexes many short-lived client connections into a much smaller number of long-lived, persistent connections to the servers. TCP offload terminates the TCP connections from both the server and the clients (TCP proxy), thus saving server CPU processing or context switching time. It also accelerates connection setup (three-way TCP handshake) between the client and the Centralized ADS device rather than between the client and the often busy Web server.

Outbound TCP Optimization can improve TCP connection performance by providing consistent and up-to-date outbound TCP connection implementation between the data center servers and browsers. Since most traffic from a Web server is outbound, it is important that the outbound connections work as well as possible. Outbound TCP optimization benefits performance most when servers and browsers are running multiple TCP implementations.

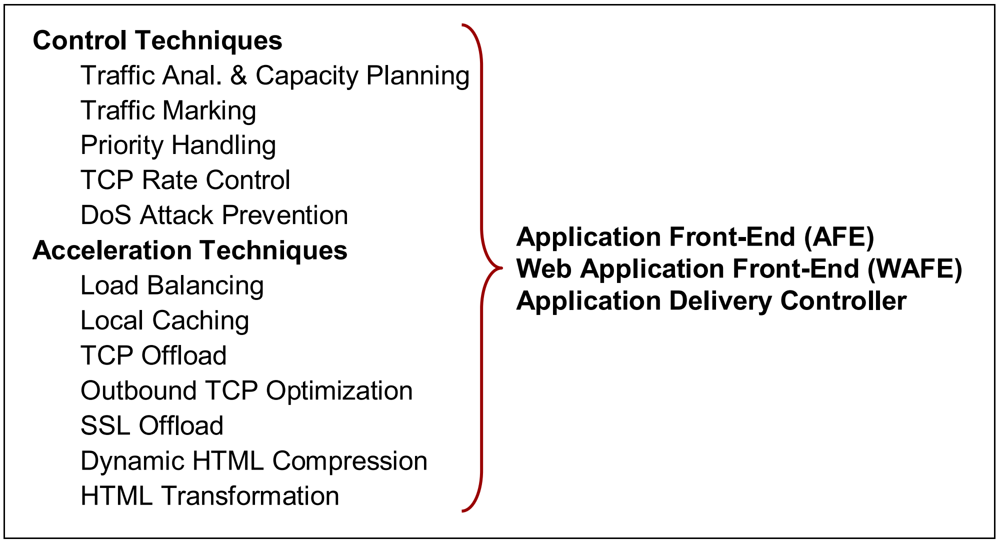
Outbound TCP optimization allows WAN TCP parameters to be managed in a consistent way, and it provides a means to ensure that the outbound TCP implementation incorporates all performance enhancements standardized by the IETF in its Request for Comment (RFC) list shown in Appendix A. Outbound TCP optimization requires TCP proxy capabilities, thus TCP offload must also be in place for outbound TCP optimization to work.

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 Web sites 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.

Figure 8 shows other common names applied to Centralized ADS solutions.



Control Techniques
- Traffic Anal. & Capacity Planning
- Traffic Marking
- Priority Handling
- TCP Rate Control
- DoS Attack Prevention

Acceleration Techniques
- Load Balancing
- Local Caching
- TCP Offload
- Outbound TCP Optimization
- SSL Offload
- Dynamic HTML Compression
- HTML Transformation

Application Front-End (AFE)
Web Application Front-End (WAFE)
Application Delivery Controller

Figure 8 – Common Names for Centralized ADS Solutions

Distributed ADS Techniques Defined

Following is a list of the primary control and acceleration techniques employed in Distributed ADS solutions, and definitions for each. Figure 8 shows how they fit into the general taxonomy.

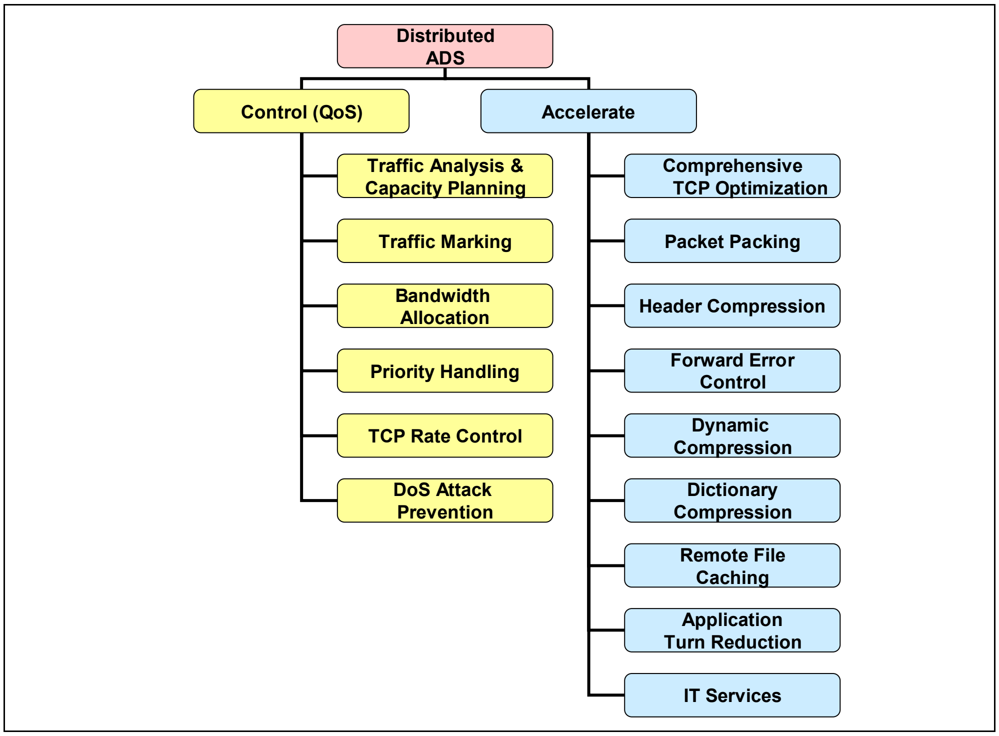


Figure 9 – Distributed ADS Solutions

Control

Traffic Analysis and Capacity Planning monitor and report on application server and WAN health to help determine quality of service policies, properly size resources, and correct problems that affect performance. The focus is typically on WAN bandwidth available for the data center and each remote location.

Traffic Marking provides information to downstream devices regarding how to handle different types of application traffic. Distributed ADS devices generally use IEEE 802.1p / 802.1q, Type of Service (TOS), and DiffServ Codepoints traffic marking standards. In many cases, these distributed devices also read the marking that was supplied either by another distributed device or another element in the overall system.

Bandwidth Allocation uses pre-set policies to limit the bandwidth that any single user/application or group of users/applications may consume in the face of competing traffic. The traffic categories are either based upon traffic marking, IP address, or TCP socket numbers. Often the policy permits a traffic category to have as much bandwidth as it needs if there is no demand from a higher category. But in periods of heavy traffic, the allocations limit each category to a predetermined bandwidth at the control point where the ADS device is operating (typically near a WAN link).

Priority Handling delivers traffic in a specified order of priority depending on the traffic marking. The devices apply policies and priorities to different traffic types to ensure that the performance of critical applications is protected during periods of network congestion. Several packet prioritization and queue management techniques may be used, including WFQ, CBQ, RED, DiffServ, or HFSC (Hierarchical Fair Service

Curves). However, many vendors of these systems have added proprietary extensions to these standard techniques in order to provide slightly better priority handling relative to the standards.

TCP Rate Control applies proprietary techniques that buffer and then pace packets into downstream systems to smooth the bursty nature of data traffic. In a distributed solution, rate control is applied physically near both the server and client TCP implementations (on the LAN). The rate control algorithm determines the highest effective achievable bandwidth, and paces the traffic at a rate just below the maximum effective bandwidth. The send and receive TCP implementations quickly settle into a window and ACK pattern that supports the prescribed rate. This eliminates the too-fast-too-slow saw tooth pattern of typical TCP connections that have constricted bandwidth along the path. The technique is more effective during the transfer of large files.

Most distributed TCP rate control implementations operate independently in both directions, but some Distributed ADS vendors add information to the algorithms such as precise round-trip timing, to make their TCP rate control implementations more effective.

DoS Attack Prevention blocks malicious attempts to tie up client, network, or server resources so the resources are available for legitimate users. Since a distributed ADS device is often placed in a critical WAN ingress/egress location, it sees all of the traffic that is entering the enterprise location. Therefore, the DoS prevention can be very comprehensive. In addition, many devices look for signature patterns of worms, viruses, zombies, etc and can proactively discard such traffic or in some cases quarantine the offending desktop or server. DoS attack prevention in Distributed ADS devices can operate in stand-alone mode or in concert with peers. Peer-to-peer exchange of information about malicious traffic can serve as an early warning and can benefit from geographic context (e.g., a worm just attacked location A and is coming your way).

Acceleration

Comprehensive TCP Optimization is markedly different from the outbound TCP optimization discussed in the Centralized ADS technique section. Because there are two devices in the path between the server and client, they can terminate both connections locally on the LAN. Replacing one TCP connection with three distinct connections that are under the control of the ADS devices decouples the behavior (or misbehavior) of the client and server TCP implementations.

The LAN connections at the server and at the client ends can be enhanced with most of the benefits attributed to centralized TCP offload and outbound TCP optimization. The remote ADS-to-client TCP implementation need not incorporate all of the RFCs listed in Appendix A since most of them do not apply to a LAN connection. The TCP connection operating over the WAN can take on all of the performance enhancements described in Appendix A as well as implement proprietary enhancements.

Comprehensive TCP optimization increases throughput by matching the transmission rate to a constrained bandwidth access line, and/or overriding TCP window limits on high bandwidth but long latency paths. Most Distributed ADS devices implement proprietary variations or extensions to the “standard” techniques described above, 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. This connection can add more tricks to selective acknowledgement (SACK), or increase TCP window size (putting more data “in flight” on long latency paths) much more aggressively than standards permit.

Packet Packing selectively bundles or concatenates separate packets into a single larger packet, eliminating header overhead associated with separate packets. The techniques is typically applied to VoIP, however, value is only achieved when there are many simultaneous VoIP calls between two locations.

Header Compression removes redundant data from TCP/IP or UDP/IP headers that exists between two well-known locations. If there is a lot of traffic between two specific hosts on an IP network, then many of the fields in the headers are the same or predictable. Header compression takes advantage of this knowledge and compacts the headers such that they can still travel through a routed network. This technique is typically applied to VoIP traffic.

Forward Error Control allows errors to be corrected in data received, which reduces the need to retransmit data when packets are lost. Forward error control is helpful in conditions of high network congestion, and it is typically applied to very time sensitive data that will not benefit from retransmission after loss like voice or video. It requires the addition of forward error correction bits, which add some overhead.

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 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 that are about 16 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. This approach only works when the file name remains identical. If the file name changes, the user will get a "miss" and the renamed file must be retrieved from the origin file 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 generally limited to a specific file family such as Windows NTFS, Windows DFS, Sun NFS, Linux FHS, etc.

Application Turn Reduction reduces 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 application 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 re-transmitting 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 is implemented with lightweight clones of the original server and client application software. This requires that the vendor in essence reverse engineer a sufficient amount of the application transactional logic so that the reduction can both take place and not hinder the proper operation of the application. It is often tricky to intercede and then reconstruct the application-level interactions on both the client and server ends so as to preserve client-server protocol semantics.

Therefore, this technique only applies to protocols or applications that the vendor has decoded so 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.

IT Services redirect Microsoft IT functions to be performed locally rather than centrally. The solution is only needed when the typical local Microsoft servers at a remote office are not present, usually due to server centralization. IT services lower WAN bandwidth usage and speed response time.

IT services are fundamentally a file caching solution for Microsoft systems with added Microsoft features such as active directory, print server, DNS, systems management server (SMS) services (SMS redistribution), support for end-to-end security schemes like signed SMB, and file locking and consistency checks.

A local box (software on a server or an appliance) is on the LAN like a local file server, and users either map their network drives directly to the local appliance, or use a file-level redirection solution like Microsoft DFS-N (Distributed File System Namespaces).

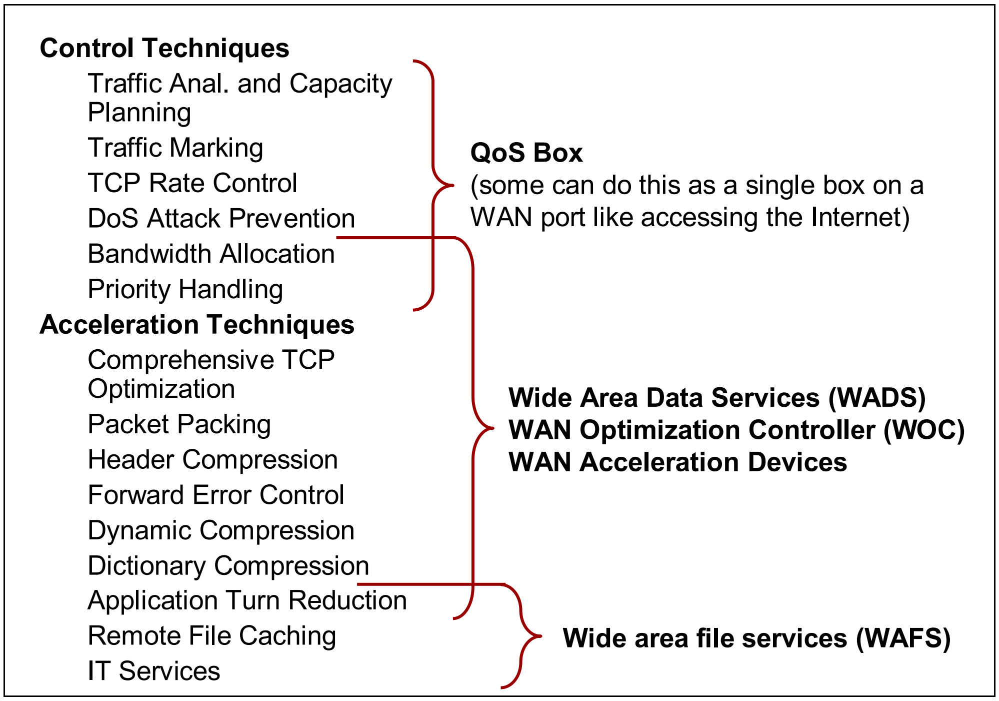


Figure 10 – Common Names for Distributed ADS Solutions

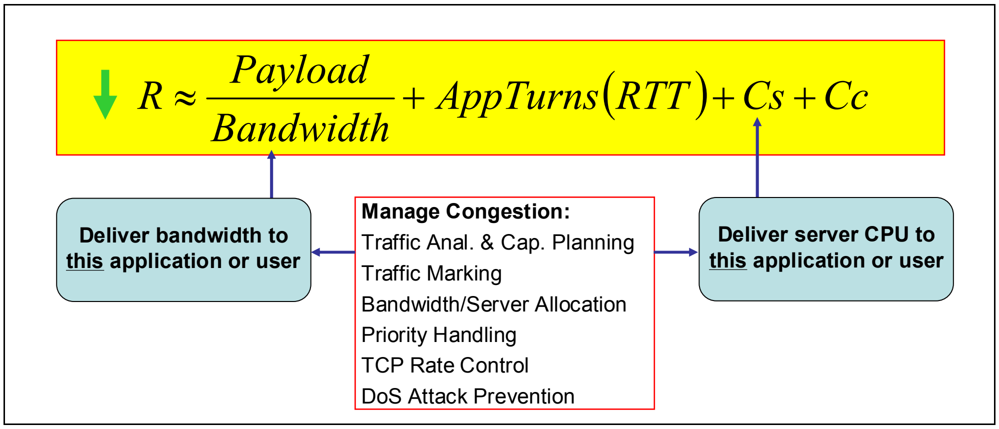
Choosing the Right Solution Set

Now that you are familiar with the ADS solution landscape, it is time to match solution types with your unique needs. In this section we describe the circumstances which call for control techniques, and those which call for acceleration techniques—and we match individual techniques with the common problems they alleviate and the protocols they help.

When to Use ADS Control Techniques

When you need to protect your users' application performance experience from insufficient bandwidth and/or server resources, you need to look at control solutions. The subset of control techniques that will help in your situation depends on the particular performance problem you face, and the particular protocol(s) that you need to protect.

Figure 11 shows that control devices protect performance by managing and allocating access to bandwidth or server resources by application or user.



$R \approx \frac{Payload}{Bandwidth} + AppTurns(RTT) + Cs + Cc$
Deliver bandwidth to this application or user
Manage Congestion:
Traffic Anal. & Cap. Planning
Traffic Marking
Bandwidth/Server Allocation
Priority Handling
TCP Rate Control
DoS Attack Prevention
Deliver server CPU to this application or user

Figure 11 – Control "Levers"

Protects Business Continuity: Control functions help ensure business continuity by keeping traffic moving efficiently during periods of network stress. This is vital because network stress is a given, be it from traffic shifts, improper network use, attacks, or a host of other sources.

Manages Application Traffic: Control devices bridge gaps between supply and demand for network and application resources. They also balance network resource use and application performance levels with business needs so that the most vital applications are allocated more network resources and therefore are likely to experience better performance during periods of network stress than those of least importance. The benefits of application traffic management apply broadly across networks and applications.

Enables Predictable Performance: Control devices provide knowledge about the traffic traversing the WAN, and this information allows you to pinpoint application performance problems, control application performance, and provides ongoing visibility into changes in performance.

When to Use ADS Acceleration Techniques

When you need to speed the user experience at all times, you need to determine what elements in the performance equation you can influence to give you the best results. That will determine which subset of techniques to consider. Figure 12 shows that acceleration devices improve performance by reducing payload, shortening round trip times, using bandwidth more effectively, reducing turns, and/or offloading the server.

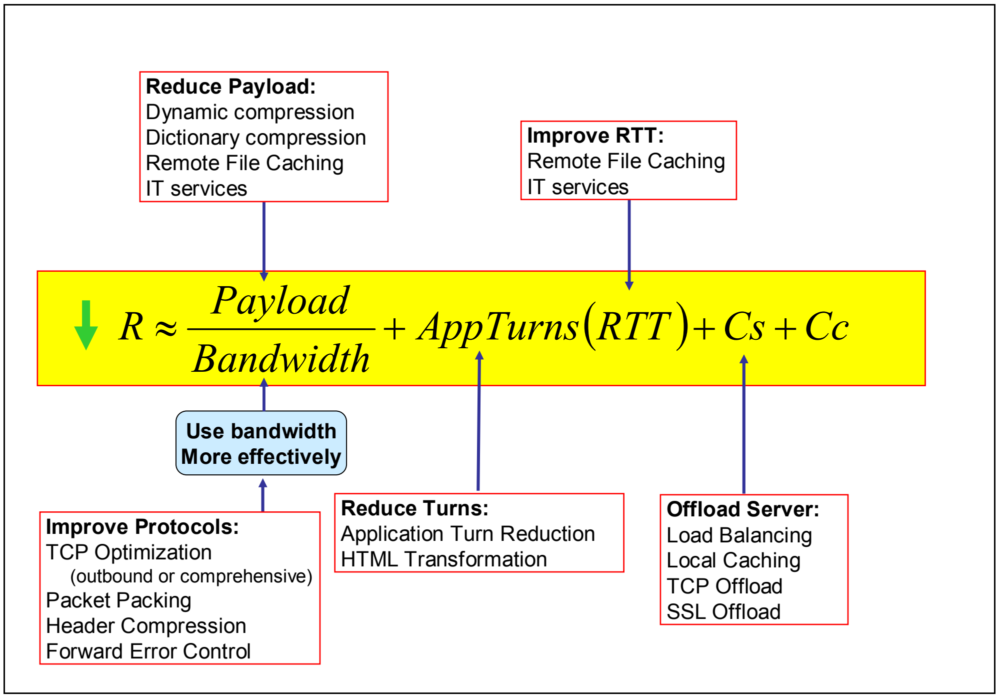


Figure 12 – Acceleration "Levers"

Enhances User Productivity: Acceleration solutions provide better performance, improving the productivity of all application users, no matter where they are. Note that acceleration improves the performance of distant users most of all, allowing them enjoy a more "local-like" experience.

Enables Remote Use of Centralized Resources: Acceleration solutions are essential to successful server centralization, and to realizing the savings and efficiencies that server centralization brings.

Facilitates Remote Collaboration: Application acceleration enables teams to collaborate across long network distances, and to share work among distributed offices despite challenging network bandwidth constraints.

Extends Reach: Accelerating applications extends an enterprise's ability to work effectively with employees, partners, and customers in previously unreachable areas.

Which ADS Approach and Techniques to Use

The following Tables summarize the effects these techniques have on improving application delivery. The tables are grouped into centralized solutions followed by distributed solutions.

Centralized ADS Tables

Tables 1 and 2 show the performance problems that each of the Centralized ADS control and acceleration solutions addresses and the extent to which it can help. Tables 3 and 4 show the protocols that each control and acceleration solution is designed to help, and also the extent to which it can help each protocol.

Table 1 - Problems Centralized ADS Control Solutions Address

Problem Improved	Traffic Anal & Cap Planning	Traffic Marking	Priority Handling	TCP Rate Control	DoS Attack Prevention
Long Distances (speed of light problem)					
High Turn Counts (chatty applications)					
Big Payloads (e.g., large file transfers)					
Insufficient Bandwidth (e.g., constrained last mile connection)		●	●	●	
Bandwidth Congestion (e.g., bandwidth hogs, flash crowds)	●	●	●		
Insufficient Server Capacity (e.g., moves, adds, changes)	●				
Server Congestion (e.g., DoS attacks)	●				■

Table 2 - Problems Centralized ADS Acceleration Solutions Address

Problem Improved	Load Balancing	Local Caching	TCP Offload	Outbound TCP Optimization	SSL Offload	Dynamic Compression	HTML Transformation
Long Distances (speed of light problem)							■
High Turn Counts (chatty applications)							■
Big Payloads (e.g., large file transfers)				●		■	
Insufficient Bandwidth (e.g., constrained last mile connection)				●		●	■
Bandwidth Congestion (e.g., bandwidth hogs, flash crowds)						●	■
Insufficient Server Capacity (e.g., moves, adds, changes)	■	●	●		●		
Server Congestion (e.g., DoS attacks)							

Legend

Dramatic Improvement	■
Minor Improvement	●

Table 3 - Protocols Centralized ADS Control Solutions Address

Protocols Affected	Traffic Anal & Cap Planning	Traffic Marking	Priority Handling	TCP Rate Control	DoS Attack Prevention
Remote Desktop Services					
NTFS, DFS, NFS, etc.					
FTP (distribution, back-up)					
MAPI					
CIFS					
XML					
HTML	●	●	●	●	■
HTTPS (SSL)	●	●	●	●	■
HTTP	●	●	●	●	■
TCP	●	●	●	●	■
VoIP or IPVideo session (TCP)	●	●	●		
VoIP or IPVideo bearer (UDP)		●			
Streaming audio or video (UDP)					
UDP					
IP		●	●		●

Table 4 - Protocols Centralized ADS Acceleration Solutions Address

Protocols Affected	Load Balancing	Local Caching	TCP Offload	Outbound TCP Optimization	SSL Offload	Dynamic Compression	HTML Transformation
Remote Desktop Services							
NTFS, DFS, NFS, etc.							
Remote file management (FTP, back-up)							
MAPI							
CIFS							
XML	■						
HTML	■	●	●	●	●	●	■
HTTPS (SSL)	■		●	●	●		
HTTP	■		●	●			
TCP	■		●	●			
VoIP or IPVideo session (TCP)	■		●	●			
VoIP or IPVideo bearer (UDP)	■						
Streaming audio or video (UDP)	■	■					
UDP							
IP							

Distributed ADS Tables

Tables 5 and 6 show the performance problems that each of the Distributed ADS control and acceleration solution addresses and the extent to which it can help. Tables 7 and 8 show the protocols that each control and acceleration techniques is designed to help, and the extent to which it can help each protocol.

Table 5 - Problems Distributed ADS Control Solutions Address

Problem Improved	Traffic Anal & Cap Planning	Traffic Marking	Bandwidth Allocation	Priority Handling	TCP Rate Control	DoS Attack Prevention
Long Distances (speed of light problem)						
High Turn Counts (chatty applications)						
Big Payloads (e.g., large file transfers)						
Insufficient Bandwidth (e.g., constrained last mile connection)	●	●	●	■	●	
Bandwidth Congestion (e.g., bandwidth hogs, flash crowds)	●	●	●	■		■
Insufficient Server Capacity (e.g., moves, adds, changes)	●					
Server Congestion (e.g., DoS attacks)	●					■

Table 6 - Problems Distributed ADS Acceleration Solutions Address

Problem Improved	Comprehensive TCP Optimization	Packet Packing	Header Compression	Forward Error Control	Dynamic Compression	Dictionary Compression	Remote File Caching	Application Turn Reduction	IT Services
Long Distances (speed of light problem)	●							■	■
High Turn Counts (chatty applications)								■	
Big Payloads (e.g., large file transfers)	■				●	■	■		
Insufficient Bandwidth (e.g., constrained last mile connection)	●	●	●	●	●	■	■		
Bandwidth Congestion (e.g., bandwidth hogs, flash crowds)	●				●	■	■		
Insufficient Server Capacity (e.g., moves, adds, changes)									
Server Congestion (e.g., DoS attacks)									

Legend

Dramatic Improvement	■
Minor Improvement	●

Table 7 - Protocols Distributed ADS Control Solutions Address

Protocols Affected	Traffic Anal & Cap Planning	Traffic Marking	Bandwidth Allocation	Priority Handling	TCP Rate Control	DoS Attack Prevention
Remote Desktop Services						
NTFS, DFS, NFS, etc.	●	●	■	●	●	■
Remote file management (FTP, back-up)	●	●	■	●	●	■
MAPI	●	●	■	●	●	■
CIFS	●	●	■	●	●	■
XML	●	●	■	●	●	■
HTML	●	●	■	●	●	■
HTTPS (SSL)	●	●	■	●	●	■
HTTP	●	●	■	●	●	■
TCP	●	●	■	●	●	■
VoIP or IPVideo session (TCP)	●	●	■	●	●	■
VoIP or IPVideo bearer (UDP)	●	●	■	■		●
Streaming audio or video (UDP)	●	●	■	■	■	●
UDP	●	●	■	●		●
IP	●	●	■	●		●

Table 8 - Protocols Distributed ADS Acceleration Solutions Address

Protocols Affected	Comprehensive TCP Optimization	Packet Packing	Header Compression	Forward Error Control	Dynamic Compression	Dictionary Compression	Remote File Caching	Application Turn Reduction	IT Services
Remote Desktop Services	●								■
NTFS, DFS, NFS, etc.	●				●	■	■		■
Remote file management (FTP, back-up)	●				●	■	■	■	
MAPI	●				●	■	■	■	
CIFS	●				●	■	■	■	
XML	●				●	■	■	■	
HTML	●				●	■	●	■	
HTTPS (SSL)	●				●*	■*		■*	
HTTP	●				●	■		■	
TCP	●				●	■			
VoIP or IPVideo session (TCP)	●								
VoIP or IPVideo bearer (UDP)		●	●	●					
Streaming audio or video (UDP)				●			■		
UDP									
IP									

* Only works if ADS can decrypt-accelerate-recrypt the payload

ADS Vendor Matrix

Look up what each vendor actually supports and what they call it at: www.netforecast.com

The ADS Vendor Matrix will be periodically updated as new vendor information is available.

NetForecast helps enterprises and vendors understand and improve the performance of networked applications.

Additional information is available at: www.netforecast.com

Conclusion

This field guide provides information needed to make sense of the types of solutions designed to help applications perform well over WANs. It summarizes the problems each solution solves, for what protocols the solution solve the problems, and where the solutions sit in the network. For more information about particular vendor solutions and how to map to performance problems and protocols please visit www.netforecast.com.

About the Authors

Peter Sevcik is President of NetForecast and is a leading authority on Internet traffic, performance, and technology. Peter has contributed to the design of more than 100 networks, including the Internet, and holds the patent on application response-time prediction. 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 with unparalleled inside knowledge of the Internet service and product markets. She works with network product vendors and service providers to develop and implement product strategies. She can be reached at rebecca@netforecast.com.

Appendix A – TCP Performance Optimization RFCs

RFC	Title	Date
1323	TCP Extensions for High Performance	May 1992
1337	TIME-WAIT Assassination Hazards in TCP	May 1992
1948	Defending Against Sequence Number Attacks	May 1996
2018	TCP Selective Acknowledgment (SACK) Options	Oct 1996
2581	TCP Congestion Control	Apr 1999
2883	An Extension to the Selective Acknowledgement (SACK) Option for TCP	Jul 2000
3042	Enhancing TCP's Loss Recovery Using Limited Transmit	Jan 2001
3168	The Addition of Explicit Congestion Notification (ECN) to IP	Sep 2001
3390	Increasing TCP's Initial Window	Oct 2002
3465	TCP Congestion Control with Appropriate Byte Counting (ABC)	Feb 2003
3742	Limited Slow-Start for TCP with Large Congestion Windows	Mar 2004
3782	The NewReno Modification to TCP's Fast Recovery Algorithm	Apr 2004
4015	The Eifel Response Algorithm for TCP	Feb 2005
4138	Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP)	Aug 2005
4164	RObust Header Compression (ROHC): Context Replication for ROHC Profiles	Aug 2005