Networks Need A New Application Delivery Architecture

Robin Layland

Building a more responsive, secure infrastructure will result in higher user satisfaction.

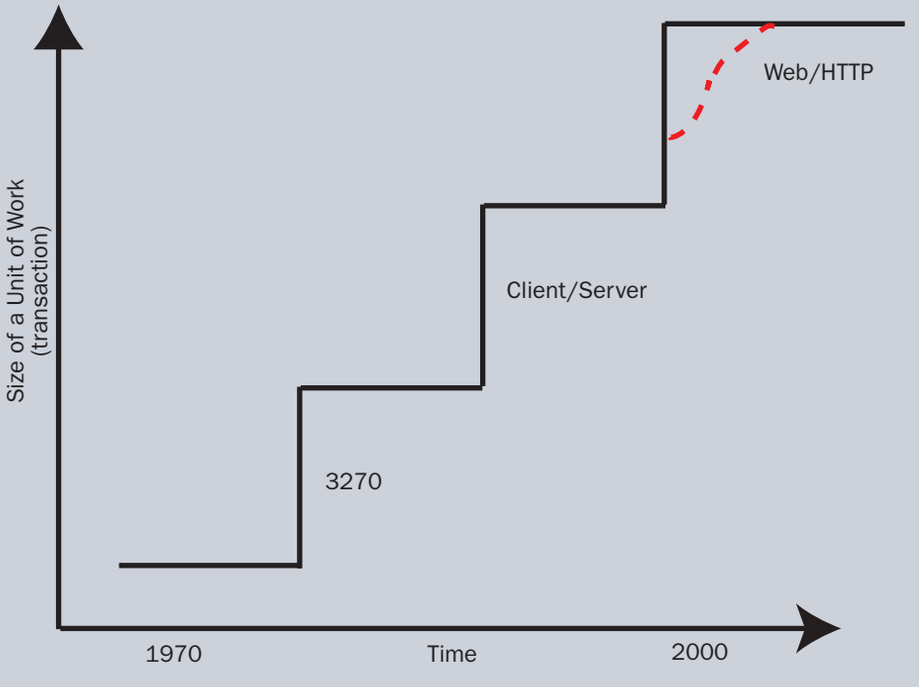
What we hired the network to do in the past—reliably switch and route packets at high speeds—has basically been achieved. Speeds of 1- and 10-Gigabits far outstrip the needs of most every user. Future challenges, such as video and TV, will require more speed, but for enterprises, these challenges are way in the future, and 40- or 100-Gigabit Ethernet will be there to solve the problem. The key is that these increases in speed will not change the network architecture; only the speeds will change.

Even security problems at the network layer have been addressed. Network layer firewalls do an excellent job of opening and closing ports and recognizing network level attacks such as denial of service (DoS). So does all of this mean our jobs are over and it's time to find a new career?

Fortunately, the answer is no. Users and customers have new demands beyond the original challenge of making packets secure and moving them faster, and the network is in the ideal position to solve these new demands.

The new jobs are all application focused, requiring networkers to move up the stack. The jobs include making applications perform better—not just making bits go faster. We also have to make them safer through better security that understands what is traversing the network at the application layer, and we have to support virtualization and consolidation.

FIGURE 1 Growth In The Size Of A Transaction



Robin Layland is president of Layland Consulting, a firm that specializes in network architecture and new technology including application acceleration, security and WiFi. He has more than 25 years' experience including technical and management positions at leading enterprises and working with the vendor community. Robin can be reached at robin@layland.com.

Networkers must implement a new application-focused architecture that requires learning new skills and deploying new equipment that performs new tricks. The new architecture is called an Application Delivery Architecture. Networkers that take up this new challenge will have a bright and interesting future.

Problem Number One

Why has the new challenge emerged now? Three trends converged to cause the need for an application delivery architecture. The first is a change in how applications are delivered and developed.

A brief history, shown in Figure 1, demonstrates what is happening. When IBM's SNA dominated, applications were developed to the 3270 standard and transactions generally ranged from 1,000 to 2,500 bytes. Client/server was the next change in how applications were developed and delivered, and again the size of a transaction increased and networks grew faster to meet the challenge.

Now application developers are moving to Web interfaces—HTTP and XML—for delivery. The size of a unit of work has exploded, but users still expect it to be delivered as fast as, or faster than, the old client/server applications.

The traditional answer was to "throw bandwidth at it," which works in many cases. But for several reasons, this approach won't work as well as it did in the past. First, the increase in transaction size is many times larger than the older client/server transactions, and few networks are able to increase their bandwidth by several factors.

The next reason is the way Web protocols work. An example shows the problem: A Web transaction asks for an object; the server realizes the client needs to be authenticated or authorized, and therefore must send an authorization request back to the client, which then obtains the authorization, forwards it to the server and finally gets the object. Only then does it ask for the next object.

In that scenario, each object and authorization required its own TCP connection. Given the connection start-up time, TCP's slow-start flow control, the transmission times and the sequencing problem, adding bandwidth may only have a small impact. More advanced techniques than just increasing bandwidth are needed to solve this problem.

Additionally, network managers can't throw bandwidth at all the problems that may exist. Workers and business partners are accessing applications over the Internet, and the network manager doesn't control the Internet bandwidth end to end.

Many network managers comment that they have migrated to Web-based applications and have not experienced the problem noted above. That does not mean the problems are not coming.

The reason is that the transition is not really like what is shown in Figure 1, a step function. Initial migrations to HTTP don't use the full power of the protocol. Instead they wrap the older client/server applications in the Web wrapper. This was initially the case with client/server applications—they took the 3270 screens and put them in a client/server wrapper. It wasn't until application developers rewrote or rolled out new applications that they started to use the functions and power in the new protocols.

This is happening again with the migration to Web-based applications. Major applications such as SAP, Oracle and others start using the Web but really they are sending down applets that allow them to "drop back" to their old client/server ways. In fact, it is causing problems for early acceleration techniques, since these techniques are built to accelerate HTTP and are not prepared to accelerate the older client/server flow.

What really happens is more like an S shaped curve, shown in red in Figure 1, than a step function. The move to Web-based applications causes a little bump up in the size of transaction at first. As new applications and older applications are fully adapted to the Web, the size of a transaction will increase as they fully integrate the power of the Web and XML.

Problem Number Two

The second problem facing network managers is datacenter consolidation or centralization. Servers are being moved from remote sites to the datacenter, meaning they have to be accessed across the lower-bandwidth WAN.

The third problem requiring a new architecture is security. Networks have played an active role in security, initially putting up firewall barriers at the perimeter. The goal was to block all traffic not authenticated, or to bar all but a few approved applications. But with the growth of Web-based applications for both external and internal users, large holes need to be opened in this perimeter barrier, because it is not practical to put every Web application outside the firewall, within the DMZ.

Additionally the types of attacks are changing. Originally, most attacks were at the network layer. Denial of service (DoS) attacks were based on starting a large number of TCP connections, something the network could detect and stop. The problem is that the attacks are getting more sophisticated and are being launched at the application layer.

A DoS attack was just flooding a server with TCP connection requests (SYN), but now it may involve flooding an application with "legitimate" requests. The result is the same—the application or server is prevented from doing its function. The attack may look like a legitimate transaction—only if you look in the application data and see that the request is for a very large number of credit card numbers can security realize it is outside the norm of a client to ask for.

"Throwing bandwidth at the problem" won't solve TCP issues

Acceleration's goal is faster page displays, not just faster packet transport

Additionally, malicious code, viruses and worms can be hidden in the application payload. Security increasingly means that the network has to understand what is happening in the application layer.

It is the goal of an Application Delivery Controller (ADC) and Application Delivery client software to implement an application delivery architecture and address all of these problems.

Anatomy Of An Application Delivery Architecture

An ADC (Figure 2) is an evolution of an earlier generation of equipment referred to as either a Server Load Balancer (SLB) or Layer 4-7 Switch. The primary functions of a SLB were to determine the best location to send the data both locally and globally, and to monitor the health of the server to ensure high reliability and availability.

These are still important functions, but the evolution to an ADC has expanded these functions to cover XML. Adding XML functionality means the SLB must delve further into the application data, requiring the ADC to assemble data that is spread over several packets, combining the data to recreate all or part of the XML message, allowing it to determine where to forward the message.

An ADC differs from the first-generation SLB in two primary areas—acceleration and security. The reason the ADC combines security, acceleration and load balancing is that the underlying technology to perform these functions is the same. The two key technologies are:

- Proxy
- Full message inspection

The *proxy* function is required because the ADC needs to be able to terminate the connection, so as to decrypt the data for the purpose of fully examining the contents and taking control of the protocol flows. For example, if an ADC is going to secure and accelerate Microsoft file servers it must understand the Microsoft CIFS protocol. This understanding allows it to respond locally to the file server, reading data blocks from the file server before the client asks for them so that it can always have a block of data ready to send to the client. This means the ADC is breaking the application layer protocol between the client and server and responding as if it were the client.

And CIFS is just one example; every application protocol that the ADC secures and accelerates must be understood by the ADC. That means the ADC may need to understand HTTP; HTTPS; XML; SOAP; IM including the different versions from AOL, Yahoo and Microsoft; RTSP; MMS; CIFS; MAPI and P2P.

There is some disagreement as to whether the proxy needs to be a "full" proxy. In the case of the file protocols, a full proxy can mean that the proxy maintains its own image of the disk and acts like a file server. An alternative approach has the proxy intercepting the protocol command and passing some directly through and responding locally to other commands. This approach, which does not require the proxy to maintain an image of the disk, is referred to as a transparent proxy.

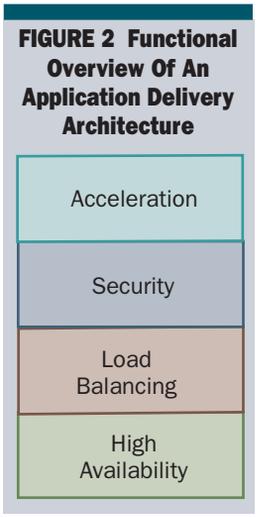


FIGURE 2 Functional Overview Of An Application Delivery Architecture

The second unifying technology is *full inspection*. Full inspection is an extension of deep-packet inspection. First-generation load balancers only needed to look as deep as the application header to understand where to send the message. But an ADC needs to be able to examine the entire packet or even to reconstruct a message that spans several packets to understand how to handle the message, look for malicious code or perform optimal caching.

There is an additional reason why security and acceleration need to be combined. For an ADC to perform acceleration, it needs to see the packet in clear text, which means decrypting the packet. If the security device is separate from the acceleration device then the accelerator must perform the VPN function of decrypting the message and then re-encrypting it. This duplicates the work the security device did and could slow the message down.

If the acceleration equipment is before the security device there are additional problems. When the accelerator applies advanced compression to the message it may substitute a single indicator for a large block of data. If malicious code or spyware is hiding in the block of data the security device will not be able to detect it.

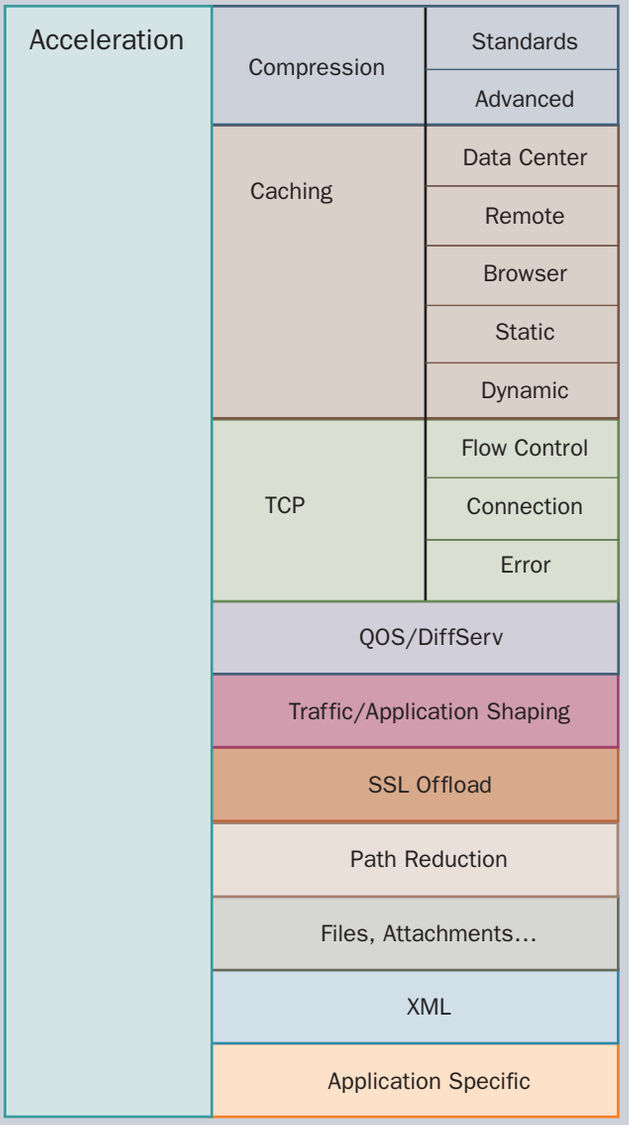
Combining the two functions within the same architected solutions provides the best design. Vendors need to either combine the functions or provide interfaces that allow their acceleration and security devices to work together.

Acceleration

The acceleration question is not how fast the network can move a packet; instead the question is how long does the page take to display or receive the entire file. No single acceleration technique can deliver transactions faster; instead the ADC applies a collection of techniques depending on the situation. The ADC has a wide range of acceleration solutions along with the intelligence to understand when to best apply them. The right part of Figure 3 shows many of the techniques in an ADC's arsenal. They include:

- **Compress**: ADC at a minimum can apply standard-based GZIP compression for HTTP transactions. More mature ADCs use advanced

FIGURE 3 Acceleration Techniques



compression techniques that recognize large patterns in the data, reducing the amount of data sent by 10 to 100 times.

■ **Caching**: Web pages are composed of a series of objects. Many of these objects are the same from page to page. The ADC can automatically recognize this and either send it from its own cache to the browser—from either a datacenter or a remote ADC—or it can direct the browser to use a previous version it already has in its own cache. The technique can be applied to static or dynamic objects.

■ **TCP**: An ADC improves upon TCP's slow-start algorithm and error correction. Additionally, it also reduces the time required to establish TCP connections. Each object requires its own TCP connection (short-lived connections), but the ADC can reuse connections between itself and the server, eliminating the need to start a new connection for each object.

■ **QOS/DiffServ**: ADC can apply the quality of service already set by the application, or can enforce QOS policy.

■ **Traffic/Application shaping**: Shaping combined with QOS ensures that less important traffic does not slow down mission-critical packets. Shaping provides the granularity that QOS alone can't, and it allows the accelerator to differentiate between different Web applications instead of just treating all Web traffic the same.

■ **Path reduction**: The ADC can reduce the number of back-and-forths between the client and server. For example, consider an object requiring authentication. The normal flow would be as described earlier in this article. But an ADC can intercept the authentication request from the server, send it directly to the authentication server, get the response and forward it directly to the server.

■ **SSL offload**: The ADC can implement SSL processing in hardware, making the processing faster and more efficient.

■ **Files, Attachments**: An ADC applies protocol improvements and caching to file requests, as demonstrated in the Microsoft CIFS example above.

■ **Application Specific**: ADC vendors are developing techniques that apply to specific applications, such as Expand Networks' acceleration for Citrix flows. This area, along with the XML and SOAP development, is where ADC vendors will be differentiating themselves.

Acceleration techniques can help with datacenter services

What application can ADC currently accelerate the most? Acceleration techniques are farthest along for Web applications, files, attachments and any application such as CAD/CAM with large message sizes. The current level of development has less effect on IM and older client/server applications, but there are exceptions with individual vendors.

Acceleration also has a positive effect on datacenter servers. Applying acceleration techniques reduces the need for server resources, a process referred to as server offload. For example, when the ADC uses its or the browser's cache for an object, the server doesn't have to expend its cycles retrieving and sending the object. The same applies when the ADC pools TCP connections or offloads SSL processing. The result is less work for the server, allowing servers to expend the saved resources on processing more transactions.

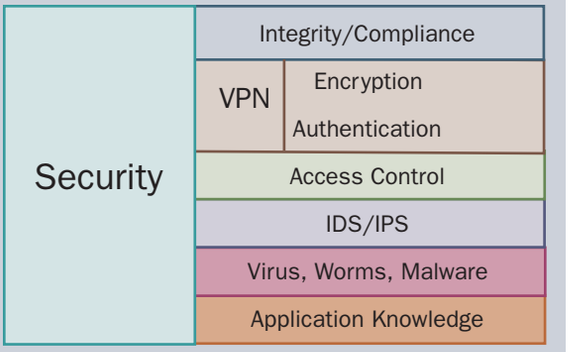
The ADC should be deployed at the network edge

Application Security

Unifying and extending a wide range of security functions further differentiates ADCs from previous networking solutions. Figure 4 shows a breakdown of the ADC's many application security functions.

One of the ADC's primary security functions is policy enforcement. The role begins when a user first enters the network and the ADC can perform integrity checking on the device to make sure it is in compliance with corporate policy. This includes checking such items as the patch release of the system and whether it has the latest anti-virus software.

FIGURE 4 Security Functions



The ADC performs the VPN encryption/decryption function to verify that the user is who they say they are, and to understand where the user is allowed to go. Another important reason the ADC needs to perform the VPN functions is because the ADC needs to see the packet in clear text.

With the ADC's ability to fully inspect packets and messages, it is also the ideal place to put detection of viruses, worms, malicious code and spyware. Full packet inspection also allows the ADC to implement the full range of specific application firewall functions for XML and other key protocols. The inspection combined with the IDS/IPS function allows the ADC to fully understand what the users are doing.

The Where

Architecture needs to answer two basic questions: the what and where. The what is provided by the functional model presented above. The next question is where should an ADC go?

The most important place for an ADC is in the datacenter or in front of the WAN. The datacenter ADC, also called an Application Front End, provides the full range of functions—high availability, load balancing, acceleration and security. The ADC needs to understand which traffic is local and which is destined for remote users. The full range of acceleration techniques should be applied to the remote traffic, but would be a waste or even slow down the local traffic.

To gain the full benefit of acceleration, the ADC also needs to be placed at the edges of the network. This includes at remote sites, Internet gateways and wireless gateways/controllers.

Two primary factors will drive deployment of additional ADCs. First, much acceleration technology requires symmetrical ADC deployment on either end of the connection. Additionally, to gain the full the benefits of caching objects, the object needs to be close to the user. Much of the TCP error correction and flow control benefits also depend on paired ADCs.

The second reason for placing ADC at the edges of the network is traffic flow. A large percentage of enterprise traffic flows to the datacenter, but traffic also flows directly to the Internet, to servers elsewhere than the datacenter, to outsourced applications and directly between users. Without an ADC at the edges, this traffic would bypass the ADC's security and acceleration functions.

The ADC at the edges does not have to be as feature-rich as the datacenter ADC. At the edges, there is little need for load balancing or server availability as found in the datacenter ADC. However, this does not mean that the ADCs at the datacenter and remote site can be from different vendors; each vendor currently implements its own proprietary techniques, requiring single-vendor implementations.

There is the possibility that many of the ADC edge functions can be done in software on the client. Orbital Data is an example of a vendor providing this option. For security, client software is already needed for VPNs, and vendors such as Cisco already provide a rich security client.

State Of The Industry

Application Delivery Architecture and ADC are new, having emerged in the last two years. That means that an ADC with all the capabilities described here does not yet exist, but several vendors are getting close and others are providing important functions. Vendors providing ADCs originated in various niches:

- Symmetrical accelerators.
- Asymmetrical accelerators or application front ends.
- Wide area file acceleration (WAFS.)
- Secure device or application firewalls.

Symmetrical accelerator vendors include Blue Coat Systems, Certeon, Expand Networks, F5 Networks, Juniper Networks, Orbital Data, Packeteer, Riverbed and Silver Peak. The asymmetrical group includes Array Networks, Blue Coat, Cisco, Citrix, Crescendo, F5, Foundry, Juniper, Nortel and Radware. Several of the vendors, such as F5, Juniper and Blue Coat, provide both solutions.

WAFS acceleration solutions for Microsoft file servers are provided by vendors including Cisco,

Expand Networks and Tacit, but many of the other vendors can provide transparent solutions that solve the problem. Additionally, Netli is providing acceleration as a service. While many of the vendors provide the same functionality, such as advanced compression, there can be a wide variation in details, and in the resulting amount of acceleration provided.

The application security functions can be found from a wide range of vendors. The acceleration vendors provide many of the functions. Most vendors provide VPN. Application firewall vendors such as Breach, Citrix, Imperva, NetContinuum, Protegrity and Whale Communications also provide a range of application security features.

This is a new technology that touches many areas. Over time, the leading ADC vendors will offer a similar set of functions, but for several years there will be wide variation. It is up to the network manager to determine the enterprise's unique needs, and to select the best product for his or her environment.

How is a network manager to determine what is the best solution? A check list of all the acceleration technologies is not as important in selecting the right vendor. The important criteria are:

- Does it reduce response time?
- How much load does it remove from the servers?
- What security threats does it handle?

In the early stages, network managers need to concentrate on solving their particular response time, server load or security problems, but always keep in mind the larger application delivery architecture they are moving towards□

Companies Mentioned In This Article

Array Networks (www.arraynetworks.net)
Blue Coat Systems (www.bluecoat.com)
Certeon (www.certeon.com)
Cisco (www.cisco.com)
Citrix (www.citrix.com)
Crescendo (www.crescendonetworks.com)
Expand Networks (www.expand.com)
F5 Networks (www.f5networks.com)
Foundry (www.foundrynet.com)
Juniper Networks (www.juniper.net)
Microsoft (www.microsoft.com)
Netli (www.netli.com)
Nortel (www.nortel.com)
Orbital Data (www.orbitaldata.com)
Packeteer (www.packeteer.com)
Radware (www.radware.com)
Riverbed (www.riverbed.com)
Silver Peak (www.silver-peak.com)
Tacit Networks (www.tacitnetworks.com)

For several years to come, products will vary widely