SearchNetworking.com

The ESSENTIAL GUIDE for Upgrading your Network

CHAPTER 2

Moving Toward the Application-Centric Network

By Robin Layland

The data center has continued to evolve and so have the requirements placed on the network. Not long ago, it was enough for the networking group to provide load balancing based on the TCP/IP address of applications, automatically notifying operations when a server was not responding and stopping network-based attacks such as denial of service (DoS). Today, much more is required. The network must be able to route traffic based on the information within the URL or even details within the application data. It is not enough to monitor an application; now, if a particular transaction within an application is not responding or is slow, the network must alert operations.

Security means going beyond stopping network-level DoS attacks to preventing application-level DoS attacks and actively protecting applications. The network must provide these functions to banks of specialized appliances such as firewalls and DNS appliances, not just servers. In addition, the network must help keep down the costs of provisioning the data center while providing this increased functionality.

Lastly, the network has to help with response time. The move to Web-based applications, with their richer and larger transactions, has negatively affected users' response time. The network is still the first place where management looks for help in solving response-time problems. The old answer of increasing the speed of the network is expensive and may not even help, given the way HTTP works.

Application delivery controllers emerge

A new class of networking products has emerged to address all these needs. They evolved from server load balancers (SLBs) and can be thought of as next-generation SLBs. A new name was needed to differentiate them from SLBs and, unfortunately, the market has given them several names. They go by application switch, application front ends, and application delivery controllers (ADCs), with ADC slowly becoming the most common term. Many of the vendors that provide SLBs have evolved their product lines to become ADCs, but that is not true for all the vendors that pro-

vide SLB solutions. In addition, a few new vendors have built new equipment to meet this challenge.

ADCs are different from SLBs in several ways, but all the new functionality is based on how deep they can efficiently look inside a message. SLBs always had deep packet inspection, and ADCs have taken it deeper. The easiest way to think of an ADC is that it can understand the application.

A common example shows how new capabilities of ADCs allowed a company to handle explosive growth. Company A manages financial employee benefits for other companies and government agencies. Its business grew rapidly when it landed a large number of new contracts. All of its customers wanted it to begin the employee benefit at the same time—

ADCs are different from SLBs in several ways, but all the new functionality is based on how deep they can efficiently look inside a message.

at the first of the year—but its existing server farm was not up to the task. Company A needed to quickly set up a large number of servers, both to handle different customers and also different sets of sub-applications for each customer, while meeting the security requirements that critical financial applications require. The application design also needed to maintain session persistence once a customer had logged onto a particular application on a server.

ADC Key Capabilities

THE KEY DIFFERENCES between ADCs and traditional load balancers are:

- Routing and load balancing are based on information within application data
- They take over functions that are done in the server to save server resources
- They accelerate application response time
- They provide application-level security

ADCs allowed Company A to set up complicated routing rules that looked within the application-level data along with providing cookie insertion. The security and acceleration features helped them meet their service goals for the customers. All this was provided without any modification to the company's original software design.

Application knowledge

SLBs can route a packet to a server based on the information in the TCP/IP header and, in the case of Web (HTTP) applications, based on the HTTP header. ADCs have taken that ability and expanded it to include any application header and any field in the application data. This has given application owners increased flexibility. An example of an e-commerce site demonstrates how this new ability is used. One person is browsing the site while another is starting the checkout process. If the site experiences a problem, such as slow response time due to high volume, the ADC can send the person checking out to a faster server, ensuring that he doesn't encounter errors. It could go even further and monitor how much people are spending, sending someone with a large dollar value to the faster server. The ADC gives the business the ability to create rules based on any information within the application data.

This flexibility comes with a price. ADCs are not magicians, and out of the box the ADC does not understand the business it is meant to support. It has the ability to implement rules to make the business run better, but only if it is told how to do so. In the previous e-commerce example, if the goal is to provide fast response time to the big spenders, a rule must be created telling the ADC where in the application data to look and what a big spender is. This means the person setting up the ADC has to understand the application and the business. This is knowledge that most networking professionals don't have.

Taking advantage of the flexibility of the ADC requires the networking department, which owns the ADC, to work closely with the company's business and application developers to create the rules. ADC vendors have helped by providing a large set of common rules based on research and

An ADC has the ability to implement rules to make the business run better, but only if it is told how to do so.

what they have learned from other customers. Many of these general rules can be used out of the box or with simple modifications to adapt them to the business's needs. Understanding the business is still a prerequisite for knowing whether the general rules are useful, however, and detailed knowledge of the business is needed to gain a competitive advantage. Creating the business-specific rules is the hardest part of taking advantage of the ADC.

Each set of virtual ADCs can have its own set of rules; a change to one group's rules does not affect another group's virtual ADC.

One solution is to allow each business unit to create its own set of rules. But this sometimes leads to another problem. ADCs generally support multiple applications and servers, cutting across business units. It is not cost-effective for each application or business unit to have its own ADC. Combining all the rules into one master set can cause problems, however. The resulting very large rule set can slow down the ADC, and when one business unit's rules cause trouble, it sometimes affects every business unit.

An emerging solution to this problem is virtualization. Virtualization allows an ADC to be logically partitioned into multiple virtual ADCs. This lets network management allocate a set of virtual ADCs to each business unit. Each set of virtual ADCs can have its own set of rules; a change to one group's rules does not affect another group's virtual ADC. Virtualization allows the network manager to restrict each group's ability to manage only its own virtual ADC and to define the management capabilities they access.

Server offload

Another major trend made possible with ADCs is offloading work from servers to the ADC. There are two primary drivers for this trend. First, the ADC can perform tasks more cost efficiently than servers. Second, moving the function to the ADC simplifies deployment by disconnecting the function from the server software and hardware.

The problem is that, in many cases, the server would need new hardware or software to efficiently perform the function, something not always possible and sometimes problematic, given the number of servers. Offload-

ing is used primarily for SSL processing and encryptions and for TCP/IP processing.

Acceleration

Moving applications to a Web interface creates several networking problems. First, Web applications send more data per transaction than traditional client/server applications, resulting in slower response time for users accessing them over the WAN. The second problem is the way HTTP works. HTTP breaks a transaction into multiple objects that are independently sent. A group of objects is often sent serially. This serial sending of objects can slow response time so that increasing the bandwidth doesn't help response time.

The ADC can't eliminate these problems, but it can help. The first way is by compressing the objects using the browser's built-in compression algorithm, gzip. The ADC can also cache objects. The first time the ADC sees an object, it stores that object. When a user asks for the object, the ADC can send the copy from its cache. This reduces response time by eliminating the time it takes to fetch the object from the server. It also has the added benefit of reducing the load on the server, providing another form of offloading. The ADC can take caching a step further by directing the user's browser to cache the object. When the user requests the object again, the ADC directs the client to use the object in its cache, saving the time it would take to resend the object.

Web applications send more data per transaction ... resulting in slower response time for users accessing them over the WAN.

The caching feature generally applies to static objects, but some vendors also apply the technique to dynamic objects. With static objects, the ADC must be told how long to keep the object, because even static objects can change. This is a case where individual application rules can be applied to guide the ADC. The rules are especially important for caching dynamic objects.

Application security

The ability to handle network DoS and distributed DoS attacks was one of the first security features added to SLBs. The load balancer was ideally situated to protect the data center from these attacks. The ADC has

expanded on that capability by taking on new security functions. Also, as the need grows for multiple intrusion detection and prevention systems and firewalls to protect the data center, the ADC takes over the role of balancing and routing traffic to the growing number of security appliances.

The ADC's role in security is based on its ability to perform deep packet inspection and understand applications. These abilities most commonly appear in the extension of the ability to stop DoS attacks at the application layer and in Web firewalls. As a general rule, the new security features apply only to Web-based applications.

An application DoS attack has the same characteristics as a network-level DoS attack: Servers are flooded with requests in an attempt to keep them so busy that they cannot process normal traffic. In a network DoS attack, this is done at the TCP level, generally using SYN requests. Application DoS attacks are more complicated. They flood the server with a legitimate request, such as an inquiry. They can use any legitimate request. Since the ADC understands application flows, it can recognize when the number of particular requests is outside the norm and eliminate the attacking requests before they reach the server, just as it would eliminate SYN requests in a network DoS attack.

Web Firewalls Enhance ADC Security

ADCS HAVE ALSO leveraged their understanding of applications with the addition of Web firewalls. Examples of this type of protection offered by Web firewalls include:

- Ensuring that the application receives only valid inputs
- Ensuring that buffer overruns don't reach the application
- Handling command and SQL injection attacks
- Stopping cross-scripting attacks
- Preventing cookie tampering and problems resulting from applications improperly handling errors

XML

In the future, Web firewall functionality will be expanded to include XML transactions by incorporating an XML firewall. An XML firewall ensures that XML data within the application data is not exploited, patching the holes left by the application much as a Web firewall does at the HTTP level. One simple example of the role an XML firewall plays is when XML data includes a payment. The XML firewall would make sure that no unauthorized person tampered with the amount, such as changing a small payment to a large payment. This is increasingly needed as application-to-application processing removes people from the process.

A more controversial area is the role that the ADC will play in XML translation. Translation is needed between two applications that use XML when they have defined the same field differently—they both use XML but with a different dialect. This problem is generally solved by adding middleware to one of the applications and letting the middleware perform the translation. This translation role is a function some within the ADC community are proposing for ADCs. One of the primary reasons is to leverage the application rules the ADC already has.

Functionality of ADCs has grown significantly from the original load balancer, with the focus shifting deeper into the message, understanding the application and expanding its role to include security and application acceleration. The next few years

In the future, Web firewall functionality will be expanded to include XML transactions by incorporating an XML firewall.

will see these new abilities become standard and more fully developed.

The challenge will be to make it easier to manage the increasing number of ADCs required to run a data center and provide the application intelligence they need. This will be accomplished by allowing applications groups—the people who understand the applications and have the detailed knowledge to fully utilize ADC capabilities—to have a greater role in configuring the ADC. Virtualization is the key technology for allowing the network group to maintain control over the ADC while giving the business unit the control it needs to customize the ADC for its business. ■

Robin Layland is president of Layland Consulting. As an industry analyst and consultant, Robin has covered all aspects of networking from both the business and technical sides and has published more than 100 articles in leading trade journals, including *Network World, Business Communication Review, Network Magazine* and *Data Communications*. Prior to his current role, Robin spent a combined 15 years at American Express and Travelers Insurance in a wide range of jobs, including network architect, technical support, management, programming, performance analysis and capacity planning.