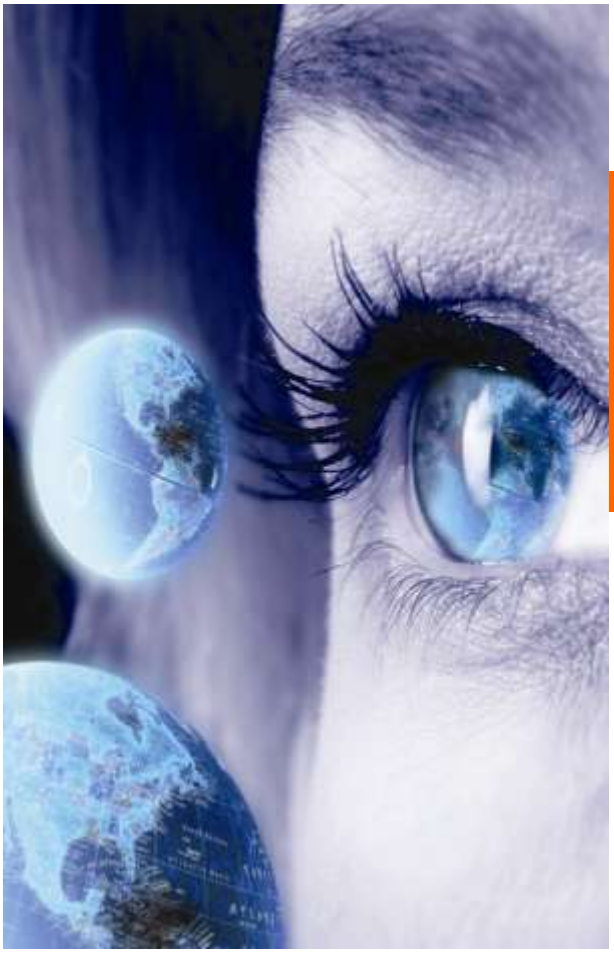
zeus

Fact Sheet



What is an Application Delivery Controller?

Zeus. Why wait…

What is an Application Delivery Controller?

Despite the best efforts of application developers, many common web and network-enabled application platforms are fragile and unstable, suffering from performance problems, the effects of flash floods and unreliable software components.

Organizations around the globe are looking to improve the reliability, availability, scalability and security of their IT systems, all while reducing internal costs and keeping IT staff focused on strategic technology projects that impact the organization's bottom line. The responsibility to fix these problems often falls between the application development experts and the network administration team.

Application Delivery Controllers

Application Delivery Controllers (ADCs) came about from a need to improve the performance and reliability of traditional web applications, but they have moved on significantly from simple hardware devices that load-balance TCP connections.

Health monitoring and application acceleration features directly address the performance, capacity and reliability problems that networked services are prone to, but that is just the beginning of their potential. The programmable nature and ever increasing set of features make modern software Application Delivery Controllers capable, powerful (and even fun!) tools for managing all sorts of limitations in network-based applications.

Common tasks that depend on network resources can be easily automated, new services can be tested and rolled out easily and security and access policies can be quickly deployed. Whether the applications depend on traditional HTTP, AJAX and REST, Web Services and SOAP or any other collection of protocols, traffic management software is proving to be a vital tool in the application developer's battle to tame and control services on the network.

What do they do?

An Application Delivery Controller will deploy a number of different features to improve the performance, reliability, security and ease-of-management of networked services.

Performance Improvements:

An ADC will improve the performance of a networked service using a number of different techniques:

- **Horizontal Scaling:** By load-balancing traffic across a number of servers, the ADC can easily increase the capacity of the service in a very controlled way. Load-balancing will direct each request to the fastest server at that instance, and session persistence will cleverly track user sessions so that applications that are not designed to be clustered can safely be scaled in this way.
- **Offloading Tasks:** Big performance gains can be realized by offloading compute intensive tasks like SSL decryption, compression and XML normalization onto the front-end ADC. This frees up CPU cycles on the application servers to concentrate on application logic.

- **Caching common responses:** By remembering the responses to common requests, the ADC can reduce the number of requests that have to be relayed on to the back-end server.
- **Network Optimization:** Many applications perform very poorly when talking to slow, remote clients over unreliable network links. A good ADC will use a range of tricks to optimize the network traffic, proxying and upgrading connections so that the application believes it's talking to fast local clients who support all the performance-improving features of the protocol.

It's not unusual to see performance benefits of 10 or even 40-fold when optimizing slow network traffic to a single server in this way!

Better reliability:

How can an Application Delivery Controller improve the reliability of your networked services?

- **Health monitoring:** If a server fails, or stops sending responses, the built-in and custom health monitors in an ADC will detect this and the ADC will avoid sending traffic to that server until it is repaired.
- **Deep traffic inspection:** The more sophisticated ADCs can be programmed to inspect the entirety of every response. If anything untoward is detected – a Java back trace for example – the request can be retried against a different server in the hope that it will give a more correct response.

With an ADC, several redundant instances of a service are often deployed, so that even if several machines were to fail completely, the service will continue to run.

Better security

The ADC acts as a gateway, a single point of entry to your applications. It can apply simple security policies, such as checking that protocol messages are well formed.

It may be able to apply very sophisticated policies, discarding denial-of-service attempts, filtering out attack signatures and validating requests against known problems with the back-end infrastructure. In many respects, you can think of an ADC as an application-layer network firewall, filtering and authenticating requests and responses as they traverse your network.

Making things easier to manage

Because it acts as a gateway, the ADC has a birds-eye view of everything that goes across your network. How quickly are services responding? What sort of traffic levels are occurring? Which servers are failing? This management information is vital when it comes to managing and deciding when to upgrade your infrastructure.

An Application Delivery Controller will even help you to manage your infrastructure. If you need to upgrade an individual server, look for an ADC with a 'draining' capability so that you can safely take a server out of service without interrupting any user sessions with that server. If you regularly install new servers or move components around your datacenter, look for an ADC that can be configured via an external API, such as SOAP, so that you can easily automate these operations.

Realizing the potential

The key feature that you'll find in the most flexible, capable traffic management devices is deep traffic inspection and programmability. It's no use having a vast armory of features to manage your application traffic if you cannot deploy these features in the way that is most appropriate to your needs, and there's no such thing as a one-design-fits-all solution.

The programmability of ADCs like Zeus Traffic Manager (using TrafficScript™ and Java™Extensions) takes them far beyond being just network optimization devices. You can begin to conceive all sorts of application logic that can be easily deployed in the traffic management layer:

- During office hours, bandwidth-limit all large downloads from each web site
- Rate-limit each user of my services to defeat brute-force login attempts
- Monitor incoming requests for service, and rewrite deprecated or legacy traffic so that it's suitable for the current generation of applications, rather than worrying about keeping old applications alive
- Build mash-up style responses on the fly, taking some XML from here, making a SOAP request there and assembling it all in one place
- Apply authentication, watermarking or security policies from a single, centralized place, rather than having to distribute and synchronize them across multiple servers and application types.

In fact, this huge degree of application intelligence lifts modern software ADCs out of the network infrastructure. It no longer makes sense to consider them alongside dumb Layer 4 or even Layer 7 network appliances managed by the network team. Many organizations are beginning to rethink the multi-tiered application model that they use – web server, app server and database – and are empowering their application developers by letting them add a new tier – the Application Delivery Controller – in front.

Zeus Traffic Manager

The Zeus traffic Manager is a good example of the new breed of Application Delivery Controllers. As a software ADC, it is flexible and can be deployed in a wide manner of ways. The TrafficScript™ programming language and Java™ extensions allow the application developer to take advantage of all of the features of the product, from rate shaping and service level monitoring to full request and response rewriting, and rapidly build innovative solutions to application problems. The Zeus KnowledgeHub at http://knowledgehub.zeus.com/ contains lots of sample code, hints and tips, and a free download for test and development purposes.

More information

www.zeus.com/products/

www.zeus.com/library/fact-sheets/

www.zeus.com/solutions/

knowledgehub.zeus.com/ - The Zeus KnowledgeHub

For further information, please email: info@zeus.com or visit www.zeus.com
Stay in touch with Zeus by following: blog.zeus.com or twitter.com/ZeusTechnology

Try before you buy.
Simply visit our website: www.zeus.com/downloads
Technical support is also available during your evaluation.

Zeus Technology Limited (UK)
The Jeffreys Building
Cowley Road
Cambridge CB4 0WS

Sales: +44 (0)1223 568555
Main: +44 (0)1223 525000
Fax: +44 (0)1223 525100
Email: info@zeus.com

Zeus Technology, Inc. (U.S.)
1875 South Grant Street
Suite 720
San Mateo, California 94402

Phone: 1-888-ZEUS-INC
Fax: 1-866-628-7884
Email: info@zeus.com
Web: www.zeus.com