

White Paper



# The Next Generation of Traffic Management

Advanced Application Delivery with NetScaler Request Switching™ Technology

## Table of Contents

Introduction .....	2
The Evolution of Traffic Management .....	3
The NetScaler Request Switching Solution .....	4
Request Switching and Application Delivery .....	4
Summary .....	7
Contact NetScaler .....	7

### Introduction

Today's IT departments face enormous challenges in optimizing their networks to deliver complex Web-based applications to a growing base of end-users. In addition, common-sense business constraints and security-conscious end-users mandate that these applications be secure so that communications remain private. But as these applications have become a "business-critical" component to the growth of companies, the required levels of availability, performance, and network scalability are not being met.

To date, no single technology addresses these challenges. Instead, IT managers have been forced to introduce significant complexity, latency and cost to their networks as they attempt

to cobble together solutions from multiple point products. As a result, businesses have compromised both the protection and secure delivery of their applications, and pay the price of degraded end-user responsiveness and network performance.

NetScaler's patented Request Switching technology solves all of these problems. By inspecting and directing incoming traffic based on the client request, Request Switching delivers applications in an accelerated, secure, and optimized manner.

## The Evolution of Traffic Management

Traffic management originated with the need to distribute Web traffic across different servers to enable more efficient utilization of site resources and increase overall site availability. In a properly configured and managed site, servers are configured so that if one server in a server farm goes down, the other servers are able to continue servicing user requests, shielding users from the server failure.

The first stage in the evolution of traffic management was the use of round-robin DNS. With this technique, the IP addresses of multiple servers are bound to a DNS name. When clients request the address associated with the DNS name, the DNS server responds with each server address in turn. In this manner, client traffic is spread among all the servers. This approach, while a good first step, did not provide the ability to monitor server state therefore leading to non-uniform load distribution. With unequal load distribution, some servers overloaded, with server requests being directed to failed servers, resulting in unacceptable service levels for clients and poor scalability for content providers.

To address some of the deficiencies of the round-robin DNS approach, traffic management evolved to server load balancers based on connection-level traffic management. Products in this category typically use a pass-through model in which modifications are made to client packets in order to route them to the appropriate servers. While these products were an improvement over earlier solutions, they were not able

to efficiently process individual requests. Rather, they made a single, key decision for all of a particular client's connections based solely on the first individual request received. The result was that connections were bound to a specific server based on a pre-configured load balancing algorithm, and the effect was non-uniform load distribution and server overload.

The next step in the evolution of traffic management brought products that make traffic distribution decisions at the content level. Traffic managers in this category take a deeper look within the packet data payload, but still distribute requests at the connection level. However, since most content switching devices still use the pass-through model, they are unable to accommodate HTTP 1.1's connection keep-alive feature. Because content switches must make a switching decision before a connection is forwarded to a server, they cannot affect requests that occur on an already-established connection. This means that each connection can only contain one request in order for content switching to be effective. By restricting connections to one and only one request, content switches preclude the connection keep-alive benefits of reduced TCP connection load for servers and improved response time for clients. In addition, due to the deeper inspection required for making a switching decision based on content, content switching is often much slower than connection-based load balancing.

## The NetScaler Request Switching Solution

NetScaler's patented Request Switching technology represents the next generation of traffic management. By breaking the link between connections and requests and inspecting all traffic at the request level, Request Switching provides high-performance, secure, and scalable application delivery.

At the core of Request Switching is a new operating paradigm in which the NetScaler system manages client connections and server connections separately from one another. Because the NetScaler system is an endpoint in each client or server connection, rather than simply passing the connection through, it offers many new acceleration and optimization techniques previously unavailable in any traffic management system.

Because the Request Switching engine was purposely built to inspect traffic at the request level, load balancing and content switching are performed in a very efficient manner. Policies and filters can be applied to incoming requests with no loss of performance. Sophisticated load distribution algorithms and health checking mechanisms ensure even distribution and continuous availability. Advanced acceleration and optimization techniques speed delivery of content to end-users while reducing the load on servers.

## Request Switching and Application Delivery

Request Switching is the basis of the NetScaler Application Delivery System. By managing traffic at the request level, Request Switching enables high-performance acceleration and optimization for delivering content to users regardless of location or connection speed. Since the NetScaler system is an active participant in the client/server communication channel, it can take advantage of HTTP version 1.1's support for persistent connections. Multiple requests can be serviced over a single client

connection, eliminating connection setup times and latency for the end-user. Persistent server connections can handle multiple client requests—even from many different clients, and even if the clients don't support persistent connections. This reduces the TCP connection processing load on the servers, allowing them to concentrate on their main task of serving content.

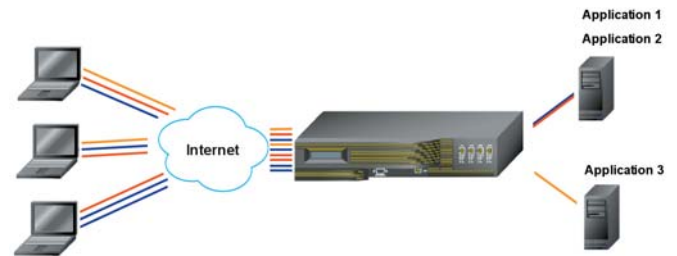


Figure 1. Request Switching using Persistent Connections

In addition, Request Switching's optimized TCP/IP stack allows the NetScaler system to eliminate the server's dependency on the client's connection speed. Once a request has been made to a server, the response can be sent to the NetScaler system at full LAN speeds. The NetScaler system will buffer the response and send it to the client at the client's own speed, freeing the server to move on to the next request.

As a result of offloading TCP processing and buffering tasks to the NetScaler system, each server is able to handle many more simultaneous requests that it would be capable of serving on its own. The resulting cost savings in server hardware and software, data center space, and maintenance is significant.

Compression is a common technique used to enhance performance and reduce bandwidth usage. Compressed content takes less time to download and thus improves user response times, while at the same time using less bandwidth. However, using compression has its drawbacks. Real-time compression is a very processor-intensive task that can overload busy servers. Pre-compressing content is an

option, but the additional management brought about by another type of content is an issue.

The NetScaler Application Delivery System solves these issues by integrating a high-performance compression engine, called AppCompress, with Request Switching to provide real-time compression of application traffic. Data coming from servers is compressed at the request level before it is sent to clients, improving download times and saving bandwidth charges. No separate compression appliance is required, servers are freed from compression duties, and maintenance of pre-compressed content is eliminated. Again, infrastructure and management costs are reduced while maintaining performance at a high level.

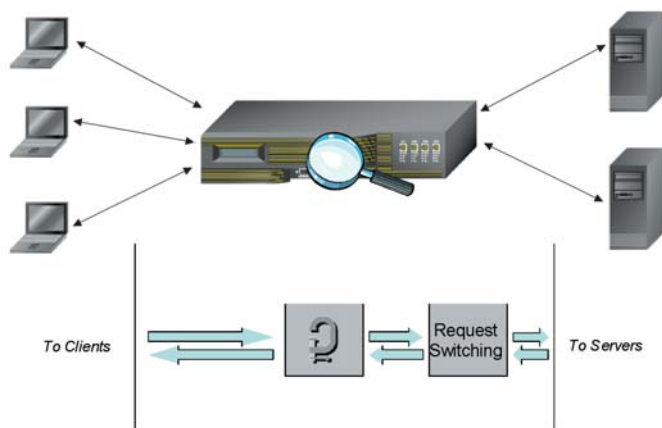


Figure 2. Request Switching with Compression

Caching is another popular method for increasing the performance of Web applications. Caches place content closer to the end-user, improving response time and reducing load on the origin servers. However, a cache server is yet another device to add complexity and management tasks to a network, and often standalone caches don't provide the desired control or dynamic content support.

By combining the Request Switching engine with a high-speed in-memory cache known as AppCache, the NetScaler Application Delivery System is able to offload a large percentage of requests from the server and deliver content at

high speed. Both static and dynamic content are supported with flexible rules for defining content cacheability and expiration. The combination of high-performance caching with content switching, compression, and other optimization features results in the best possible user response times and Web site scalability.

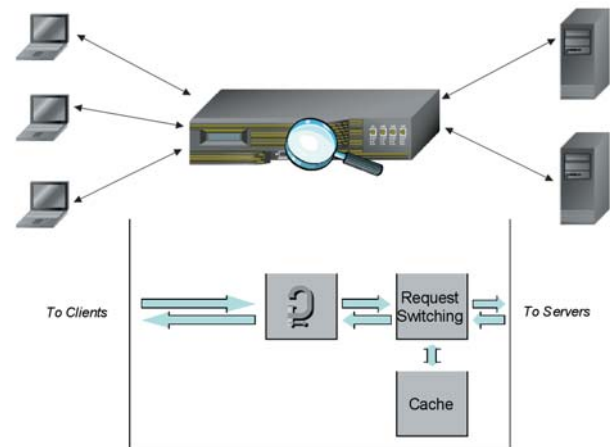


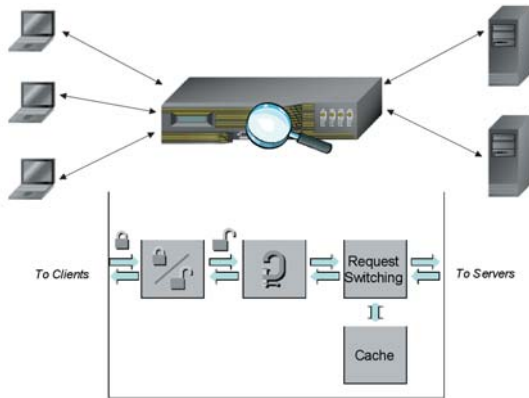
Figure 3. Request Switching with Caching

Secure content is becoming more and more common as businesses roll out new applications. However, delivery of secure content has its challenges. Because traditional traffic managers are unable to decrypt secure content, they cannot perform any content switching or other processing which requires content-level inspection. In addition, encryption and decryption are very resource-intensive tasks for servers.

Request Switching takes advantage of the cryptographic accelerator built into NetScaler systems to solve the issues associated with delivering secure content. As a request arrives, NetScaler decrypts it and then applies content switching policies and filters before sending the request to the appropriate server. Responses from the server can be compressed or cached, then encrypted for transmission to the client.

This approach allows traffic management and optimization techniques to be applied to secure content, and improves performance by removing cryptographic processing from

the servers. The availability of high-performance encryption also means that IT managers need not limit the use of secure content because of performance concerns. Security can be increased while maintaining scalability and response time at desired levels.



**Figure 4.** Request Switching with SSL Offload

Request Switching also provides scalable protection against malicious traffic. Because all traffic is inspected at the request level, attacks are prevented from reaching server infrastructure. Request Switching's optimized TCP/IP stack is well-equipped to handle network-level denial of service attacks, and requests are checked for validity before they are sent to a server. Content filtering policies can be applied to ensure that only legitimate requests are forwarded, and any request that does not meet the desired criteria is considered malicious and dropped or reprioritized. In addition, rate limits can be configured to place a cap on the amount of load directed to any one server to prevent overloads.

## Summary

NetScaler's Request Switching traffic management represents a very significant advance in managing application traffic for a global infrastructure. By managing traffic at the request level and breaking the connection between the client and server, Request Switching provides optimal control over application traffic. Combining Request Switching's granular traffic management with advanced optimization and acceleration techniques delivers performance, flexibility, security, and manageability unmatched by any other solution. □

## Contact NetScaler

**NetScaler, Inc.**  
**Corporate Headquarters**  
180 Baytech Drive  
San Jose, CA 95134  
Phone: 408 678 1600  
Toll Free: 1 800 NETSCALER  
Fax: 408 678 1601

**NetScaler Pvt. Ltd.**  
#69/3, THE SIRIUS  
Millers Road  
Bangalore – 560052  
Phone: 91 80 51341000  
Fax: 91 80 51303000

**NetScaler UK Limited**  
1 Farnham Road  
Guildford  
Surrey GU2 4RG  
United Kingdom  
Phone: 44 1483 549440  
Fax: 44 1483 549441