F5 PERFORMANCE REPORT

APPLICATION DELIVERY CONTROLLER PERFORMANCE REPORT

A comprehensive performance evaluation of F5 Networks 8800, Cisco ACE and Citrix NetScaler 12000 Application Delivery Controllers

APPLICATION DELIVERY CONTROLLER PERFORMANCE REPORT

Table of Contents

APPLICATION DELIVERY CONTROLLER PERFORMANCE REPORT

Executive Summary

This report documents the performance of the top Application Delivery Controllers offered by F5 Networks, Cisco® Systems, and Citrix® Netscaler®. Through the development and implementation of robust, transparent, and reproducible testing methods, F5 demonstrates a significant performance advantage over the competition in a comprehensive set of tests.

The market for Application Delivery Controllers (ADCs) is very competitive, with nearly every vendor claiming a performance advantage in one scenario or another. Unfortunately, the claims from each vendor rest on differing definitions of performance criteria. Each vendor has their own definitions of various terms (such as L7 and connection), preferred configuration settings for the different devices, and presentation of results. Combining these factors significantly reduces the value of typical vendor performance claims. With these inconsistent definitions between vendors, especially in the context of published data sheets, performance metrics cannot be fairly compared between vendors.

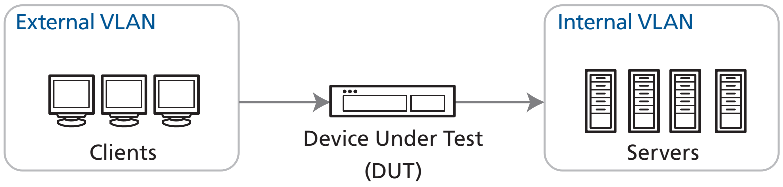
F5 is proud to release the industry's most transparent and robust performance testing guide into the public domain. To create a more level playing field, F5 is releasing a guide on *how to create performance testing methodologies*. With this publicly available guide, customers have a framework for evaluating the performance of multiple ADC products with consistent definitions and evaluation criteria.

Demonstrating the new performance testing guide, F5 has created this report comparing the latest products from F5, Cisco, and Citrix (F5's BIG-IP LTM 8800, Cisco's ACE 8GB, and the Citrix NetScaler 12000). This report compares and evaluates many common performance metrics across a wide variety of scenarios. Tests include using various web page sizes, HTTP request counts, and different types of traffic including Layer 4 (L4), Layer 7 (L7), SSL, and HTTP acceleration technologies such as compression and caching. Using a broad range of test parameters and public testing guidelines ensures prospective customers can make informed decisions based on their application requirements, without concern of vendor bias from selective reporting or conflicting performance definitions.

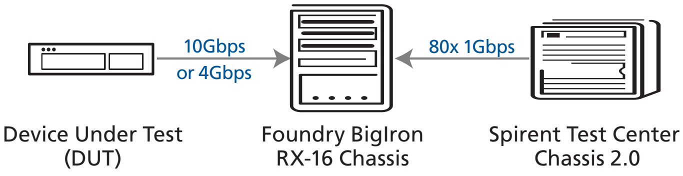
Introduction

This document contains the results of testing conducted under the available *performance testing methodologies* document. To ensure complete transparency, F5 is also publishing the *configurations for all devices involved in the testing*, including the load generation tool and L2 switching infrastructure. A dictionary of terminology used in this document is available in Appendix A.

The following diagram illustrates a high-level view of the environment in which devices were tested.



To remove the possibility of test infrastructure impacting the results of the testing, each device used as part of the test infrastructure was individually evaluated to ensure its performance capabilities far exceeded the needs of testing. As a further measure to eliminate vendor bias, the L2 switching infrastructure was selected from a vendor not otherwise used in the tests. The following diagram provides a high-level view of the physical test environment.



For more information about the test methodology, refer to the *performance testing methodology* document and the additional test details (including FAQ) in Appendix B.

Test Result Highlights

- **L7 Requests Per Second**: BIG-IP 8800 delivered over 700,000 L7 inspected requests per second, a more than 8x advantage over Citrix NetScaler and Cisco ACE.
- **L4 Throughput**: BIG-IP 8800 delivered near line-rate 10 Gbps throughput, demonstrating an advantage of more than 25% over Cisco ACE, and over 2.5x beyond Citrix NetScaler.
- **SSL TPS**: BIG-IP 8800 delivered more than 48,000 new SSL TPS, an advantage of more than 70% over Citrix NetScaler, and nearly 4x more than Cisco ACE.
- **HTTP Compression Throughput**: BIG-IP 8800 delivered 4Gbps of compressed throughput, more than 3x beyond what Citrix NetScaler achieved. Cisco does not support HTTP compression.

APPLICATION DELIVERY CONTROLLER PERFORMANCE REPORT

Scope

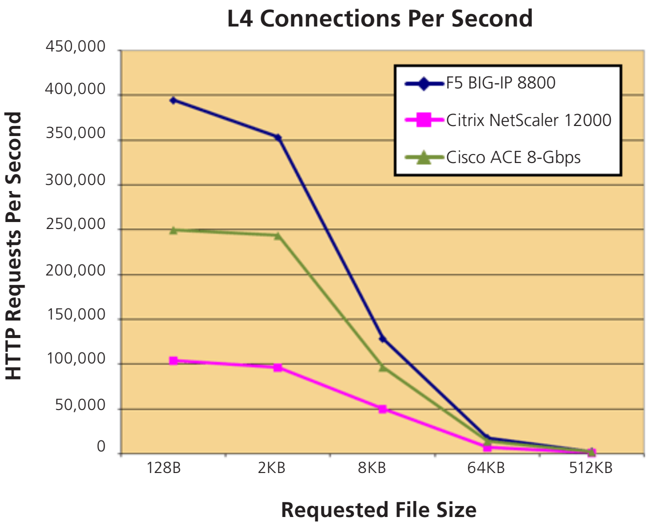
This document details the results of performance testing conducted according to the most comprehensive performance test guidelines available for public use. Even with the most advanced test methodologies, some limit in scope is required. Accordingly, the testing in this report includes only the most commonly used deployment configurations and product features.

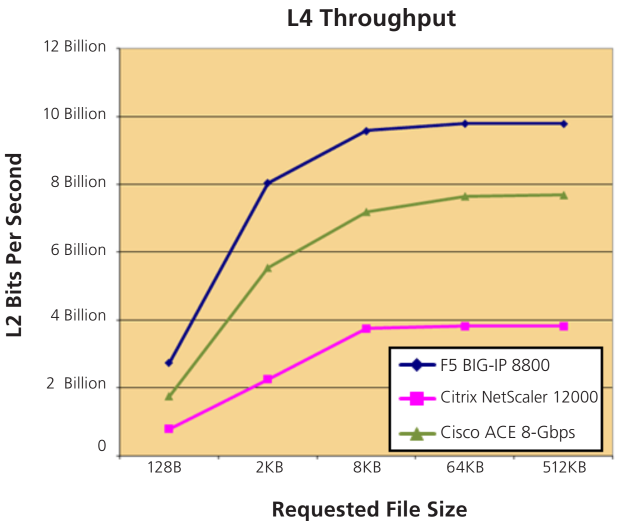
The performance of all devices varies significantly depending on client speed, server speed, network conditions, combinations of features in use, and so on. With that in mind, the ***relative*** performance of the devices tested are likely to be similar in many different situations, so these results should serve as a valuable performance guide for most common deployment scenarios.

Test Results – L4

L4 performance is a measure of basic TCP/IP load balancing, a baseline configuration with the minimum set of features enabled. L4 performance is most relevant for applications that deal with a lot of bulk data, where little application awareness is required. Load balancing FTP servers and some types of streaming media are common scenarios appropriate for L4 load balancing.

Most vendors typically have a L4 mode that uses a different "packet path" in hardware or software that is more optimized for L4-only traffic. All three devices tested had configuration options for a L4-only (or TCP only) mode, shown for comparison in the following results. L4 tests often show the highest connections per second and/or throughput results that are possible for a given Application Delivery Controller. This makes L4 testing appropriate for use in baseline capacity planning, as it is unlikely performance under more complex scenarios (i.e. with additional features enabled) will be higher than the baseline L4 results.





Analysis - L4 Performance

The BIG-IP 8800 reached a peak of 395,000 connections per second at the 128 byte response size, and reached peak throughput of 9.8 Gbps starting at the 64KB response size. The throughput performance of the BIG-IP 8800 is, for all practical purposes, line-rate 10 Gbps. This level of performance is made possible by the BIG-IP 8800's PVA10 chip (Packet Velocity ASIC 10), F5's custom-engineered ASIC. By handling this traffic in an ASIC, the main CPUs are not used at all, and are free to process other traffic.

Cisco ACE achieved 7.7 Gbps, essentially reaching the 8 Gbps throughput that it is limited to by license. ACE hit a peak of 249,000 connections per second, which is within the range of expected performance given the issue noted in the FAQ section (updated October 2007).

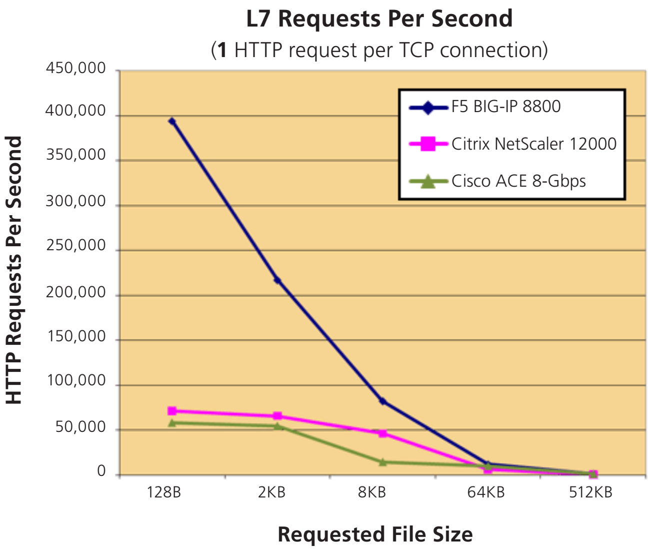
Citrix NetScaler managed up to 105,000 connections per second, at a peak of 3.8 Gbps with the largest response (see the FAQ entry for more information). It was noted during L4 tests that NetScaler consistently had the highest time to last byte , adding 3ms even with a 128 byte response.

Test Results – L7

L7 performance tests measure basic HTTP-aware load balancing; every HTTP request is inspected and then directed to the appropriate group of servers. This technology is commonly used to allow servers to host more specialized content. For example, one group of servers may store small images, another group might store large zip files, and a third could handle requests for dynamic web pages. This test performs a simple inspection of the HTTP URI to identify requests for images and direct them to a different group of servers.

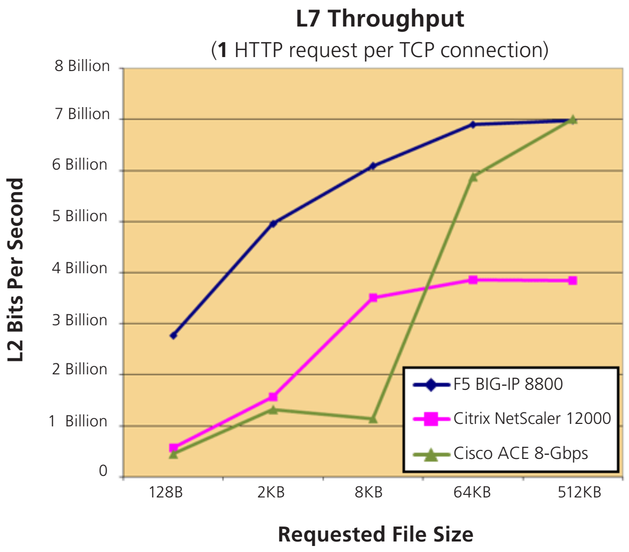
L7 performance is relevant to most applications being deployed today – it is the performance metric most often referenced when comparing Application Delivery Controllers. The most important difference between a L4 test and a L7 test is that the _DUT (Device Under Test)_ must inspect the application-layer data transferred between the clients and servers. Because every client request must be inspected, an increase in requests sent by the clients means additional stress placed on the DUT. Additionally, _HTTP request multiplexing (connection reuse)_ is enabled during these tests to provide server offload and ensure that all HTTP requests in a given connection are inspected.

The following results include three very similar tests, with each test varying the number of HTTP requests per TCP _connection_. The slight difference in the tests demonstrates the effect of TCP/IP processing vs. HTTP processing on the DUT. With one _request per connection_, there is an equal amount of TCP/IP processing and HTTP processing per request (a single TCP/IP connection is setup, a single request is sent, response received, and TCP/IP connection teardown). Adding additional HTTP requests per connection requires less TCP/IP processing, placing more stress on the L7 inspection capabilities of the DUT. Different applications will have different needs with respect to requests per connection, but it is important to know that modern web browsers attempt to send as many requests per connection as possible.



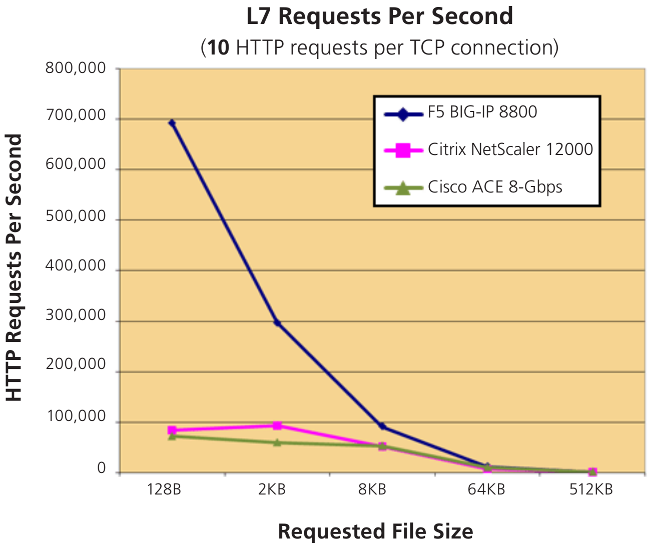
L7 Requests Per Second

(**1** HTTP request per TCP connection)



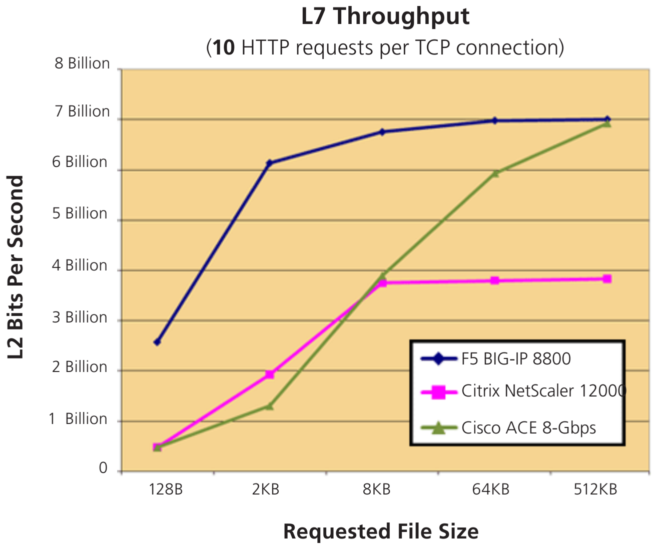
L7 Throughput

(**1** HTTP request per TCP connection)



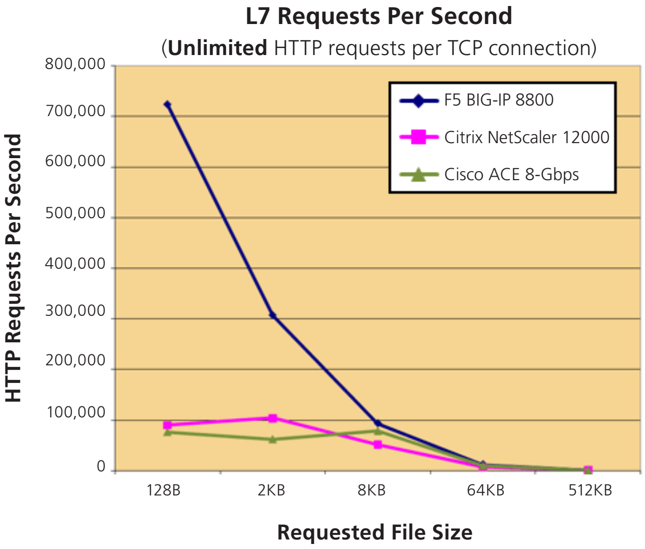
L7 Requests Per Second

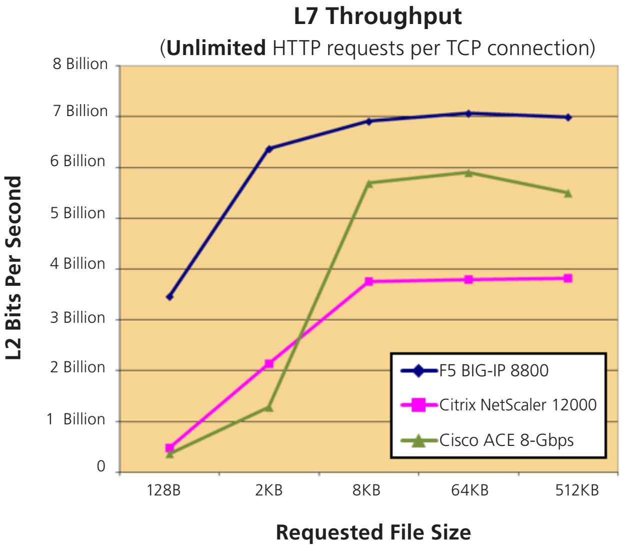
(**10** HTTP requests per TCP connection)



L7 Throughput

(**10** HTTP requests per TCP connection)





Analysis - L7 Performance

The first L7 test (1 request per connection) primarily shows the efficiency of the TCP/IP processing capabilities of each DUT. As the number of requests per connection increases to 10, then unlimited, the focus shifts to demonstrating the efficiency of the HTTP processing engine of the DUT. In general, TCP/IP connection setup is more computationally-intensive than basic L7 inspection of a single HTTP request, so as the number of requests per connection increases, so does the rate of HTTP requests per seconds.

With a single request per connection (sometimes referred to as HTTP/1.0 Connections Per Second, or L7 Connections Per Second), the BIG-IP 8800 achieved 394,000 requests per second with a 128 byte response. At 10 requests per connection this increases to 692,000, and then when unlimited, reaches 724,000. Throughput for the largest response size was 7 Gbps in all three scenarios. The performance of the BIG-IP system in this configuration (which inspects and load balances HTTP requests and provides server offload via request multiplexing) represents a 5.5x to 9.5x advantage over the competition.

Citrix NetScaler achieved requests per second performance from 71,500 (1 request per connection) up to 90,400 (unlimited), with throughput rates of up to 3.8 Gbps. It was noted that during the 10 and unlimited requests per connection tests, Citrix NetScaler achieved slightly higher requests per second performance at a 2KB response size compared to 128 byte response size. While this is a somewhat odd result, it was consistently reproducible across reboots and other diagnostic efforts. It was speculated that a different number of concurrent users would have achieved the expected result. Test guidelines do not allow for specialized settings per device. Please see the *FAQ entry regarding this issue* for more information.

Cisco ACE had the lowest L7 requests per second performance of all the devices tested, ranging from just under 60,000 to 76,400 with small file sizes. However Cisco ACE did achieve the highest throughput using a 512KB response size in the 1 HTTP request per connection test. Cisco managed 7.006 Gbps of L7 throughput compared to the BIG-IP device's 6.981 Gbps in the same test. Overall, Cisco had throughput rates from 5.5 – 7 Gbps with large file sizes.

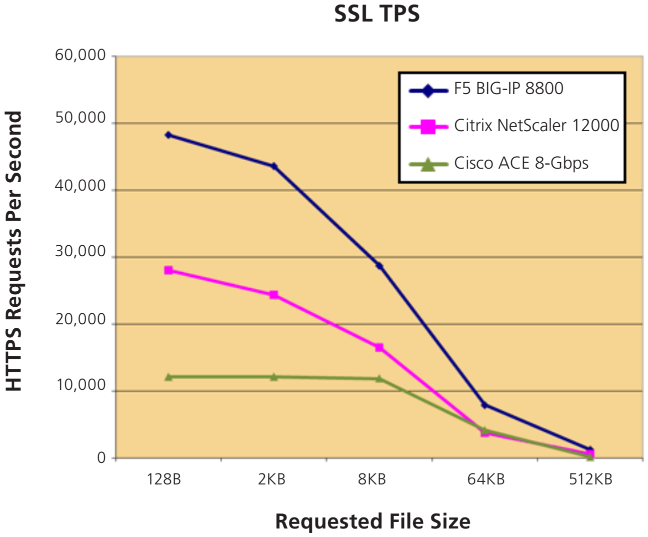
It was noted that Cisco maintained similar requests per second performance for multiple file sizes from 128 byte through 8KB in the unlimited requests per connection test. It appears Cisco ACE was not limited by its ability to process throughput, nor by TCP/IP connection setups. Instead, it seems the limit for Cisco ACE was in its ability to inspect HTTP requests faster than about 76,000 per second.

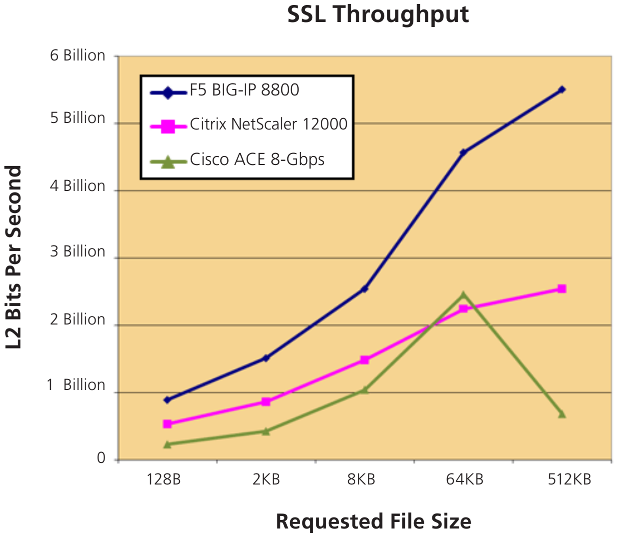
Test Results – SSL

SSL is used around the world to secure communications between users and applications. SSL is a standard encryption protocol available in every major operating system, web browser, smart phone, and so on. SSL technology helps make online shopping secure, enables secure remote access (SSL VPN) and much more – SSL is ubiquitous in commercial and consumer networking security solutions. SSL provides security using a combination of public key cryptography (typically RSA®), and symmetric encryption (commonly RC4, 3DES, or AES). Both RSA and the various symmetric encryption algorithms are computationally-intensive, and require specialized hardware to achieve acceptable performance or large scale in the nearly all commercial uses of SSL (such as web based Email, online stores, secure logins, online banking web sites, and SSL VPNs).

SSL TPS performance is a measure of encryption offload capability. For small response sizes, this primarily measures the RSA handshake operations that occur at the start of every new SSL session. This RSA operation is computationally-intensive; all major SSL offload vendors use specialized hardware to accelerate this task. For larger responses, the computational cost of the RSA operation is less relevant. Because the RSA operation only occurs once at the beginning of a session, the true comparison of performance is the throughput of encrypted traffic, also known as symmetric encryption or bulk crypto. Bulk crypto is a measure of the amount of data that can be encrypted in a given second. If a vendor has SSL offload hardware that can process bulk crypto, it is apparent in tests with large response sizes.

Tests were conducted across a range of file sizes to demonstrate the performance of both public key operations (small files) and bulk crypto (large files).





Analysis - SSL Performance

Processing SSL traffic is computationally-intensive, leading all Application Delivery Controllers (ADCs) to have some form of hardware-assisted SSL acceleration. Hardware acceleration provides a huge boost for SSL performance, but even very good SSL hardware is only as fast as the software driving it. The software of the ADC is still responsible for TCP/IP processing (SSL hardware does not help with this task). The ADC also needs software that brokers data to and from the SSL acceleration hardware (a driver). With all of the components required to process SSL traffic, it should not be a surprise that the resulting SSL performance of a device is a combination of its TCP/IP efficiency, SSL driver efficiency, and the speed and quantity of SSL acceleration ASICs.

The BIG-IP 8800 continues to lead the industry in SSL performance by a significant margin, with up to 48,300 new SSL transactions per second with small response sizes, and over 5.5 Gbps of throughput with large response sizes.

Citrix NetScaler matched its public specification by reaching 28,000 SSL TPS. SSL throughput for Citrix NetScaler reached 2.6 Gbps with large response sizes.

Cisco ACE was only able to sustain 12,100 SSL TPS -- 19.2% below the expected 15,000 SSL TPS published in ACE's specification. It was noted during the "ramp up" phase of the 128 byte test that Cisco ACE achieved roughly 20,000 SSL TPS for one measurement cycle, with the next measurement being approximately 14,000. After these two anomalously high measurements during ramp up (before steady state), Cisco ACE performed consistently at around 12,000 SSL TPS. It was theorized that Cisco ACE may be able to sustain SSL TPS performance closer to 15,000 for small response sizes if the test were changed to attempt only 15,000 SSL TPS. This theory was not tested, as test guidelines did not permit specialized settings for each vendor. This behavior only affected Cisco ACE.

While testing Cisco ACE, it was noted that the number of SSL TPS remained very close from a 128 byte to 8KB response size (as throughput increased with size), perhaps indicating SSL hardware was not the bottleneck, but instead the SSL driver or some other software that prevented higher levels of performance. A similar behavior was observed for L7 HTTP inspection performance (see above). ACE had an odd drop in throughput at the 512KB response size, but this seems to be a known issue without workaround (see the FAQ entry). This behavior was also consistently reproducible after reboots and other diagnostic efforts.

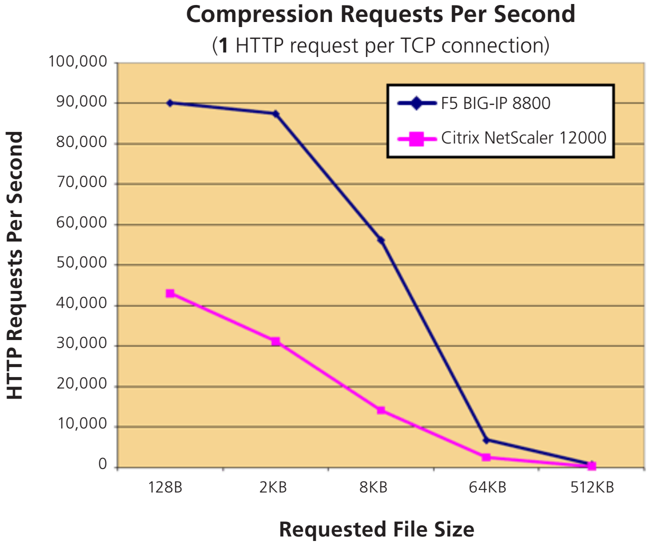
Test Results – HTTP Compression

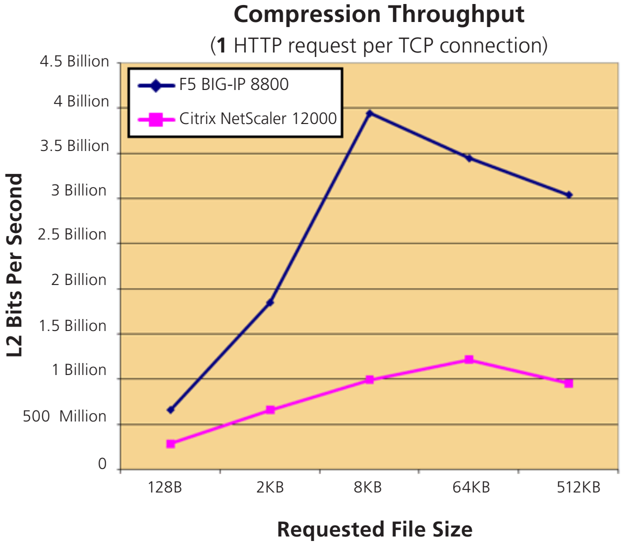
HTTP Compression performance is a measure of the standard compression algorithms supported in all modern web browsers. In situations where the bandwidth between clients and servers is limited, compression can provide significant performance benefits to end users. Compression can also help companies to achieve cost savings by reducing the bandwidth required to serve web-based applications.

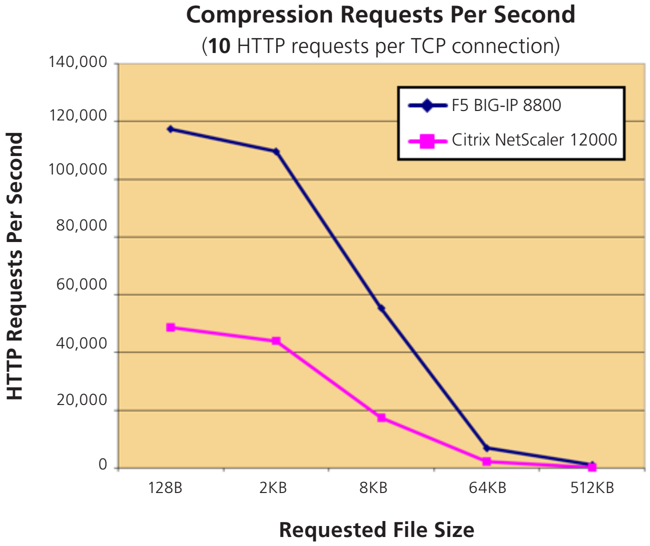
The benefits of compression are widely understood, but compression is not universally used because it's very computationally-intensive for servers. As a result, it is increasingly common to offload HTTP compression functionality to Application Delivery Controllers.

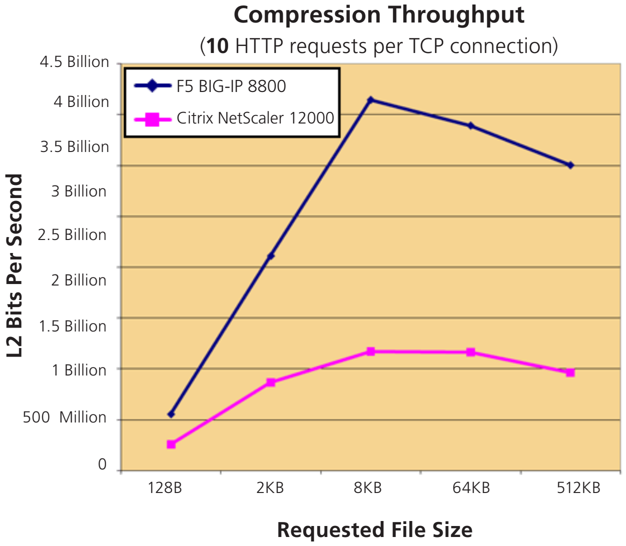
The most important metric when measuring compression is *throughput*. More data sent from the servers directly correlates with more compression work for the DUT.

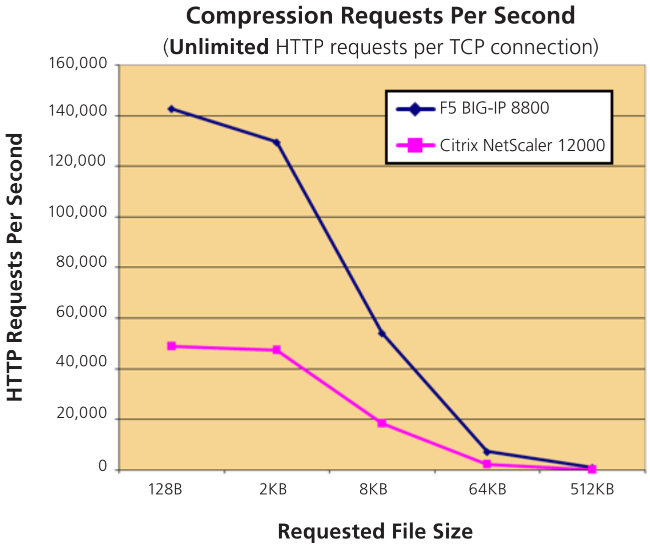
***Note:** Cisco ACE 8GB does not appear in these results because it lacks support for HTTP Compression.

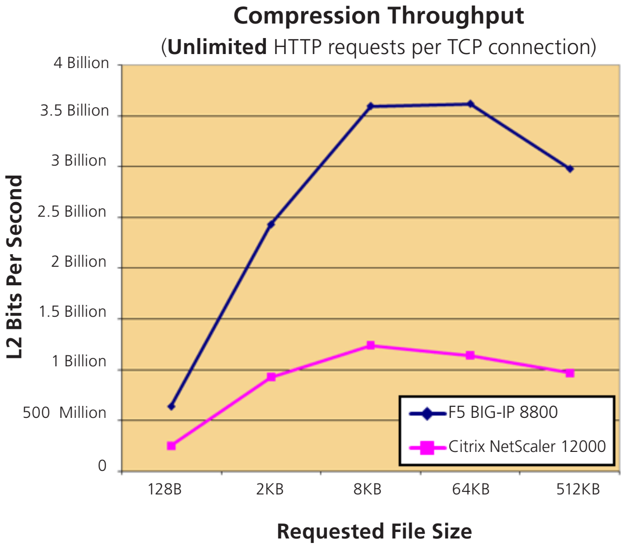












Analysis - HTTP Compression Performance

HTTP compression is a standardized technology documented in the HTTP/1.1 specification (RFC2616). Like SSL, compression is a very computationally-intensive technology. Some modern Application Delivery Controllers use specialized ASICs to perform compression, while others use general purpose CPUs (sometimes multiple CPUs at once). Historically, the specialized ASICs used for compression were less efficient at reducing the size of data, though they had much lower latency compared to CPU implementations. Some of today's compression hardware is both faster and more efficient in typical configurations than CPUs, meaning there is no longer a trade-off in choosing hardware or software compression implementations.

The files downloaded in this test (128 byte, 2KB, 8KB, 65KB and 512KB) are the same files used in all tests throughout the entire report (there is only one set of files used). These files contain typical HTML, and excluding the 512KB file, they are roughly 60% to 80% compressible (typical for HTML). The 512KB file was created by duplicating the contents of the 64KB file until it was 512KB in size. The method used to create the 512KB file ensures that the file is much more compressible than the others (it contains the same content duplicated several times). As a result, the 512KB file is closer to 90% compressible. With this in mind, the compression throughput at the 512KB data point is naturally lower as more resources are consumed to compress this highly compressible file.

Having a clear understanding of compression performance requires that we look at compression ratio in addition to throughput. The compression ratio measures how efficiently the DUT compressed the data from the server before sending it to the client. A compression ratio of 2:1 means data size was reduced by half (for example, 1000 bytes compressed to 500 bytes). Higher compression ratios are better.

File size	128B	2KB	8KB	64KB	512KB
BIG-IP	0.45528	1.59636	2.15674	3.14766	3.40133
Citrix NetScaler	0.56036	1.21030	1.30240	2.41297	7.74997

The results demonstrate that attempting to compress a 128 byte response actually made the response bigger. Both the BIG-IP and Citrix NetScaler would not normally be asked to compress 128 byte responses – the products were configured to always compress (regardless of minimum size) for the purpose of illustration.

The BIG-IP 8800 achieved from 90,000 to 142,700 requests per second with a 128 byte response (1 requested per connection to unlimited), with a throughput for large responses from 3.6 - 4 Gbps.

Citrix NetScaler reached 43,000 to 49,000 requests per second, with throughput from 1 - 1.24 Gbps, very near the 1.3 Gbps documented in Citrix NetScaler's specifications.

The BIG-IP 8800 has an advantage in both throughput and compression ratio for most of the response sizes, but Citrix NetScaler has a significant advantage in compression ratio for the 512KB response size. It was postulated that Citrix NetScaler's default compression settings were well-suited for the type of duplication known to exist in the 512KB file. Both products have various configuration options to tune compression for higher compression ratios or higher throughput. These settings were not evaluated.

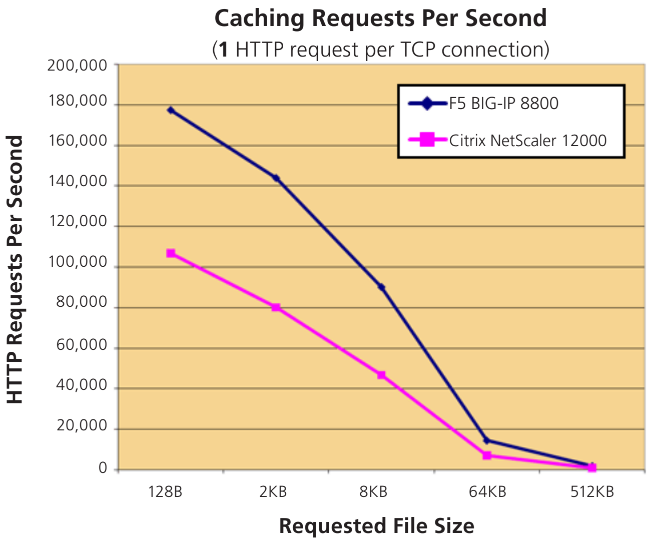
As noted previously, the Cisco ACE does not support HTTP compression.

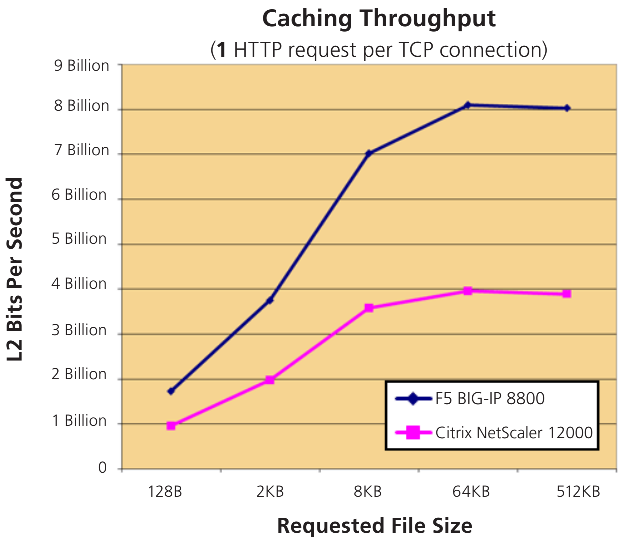
Test Results – HTTP Caching

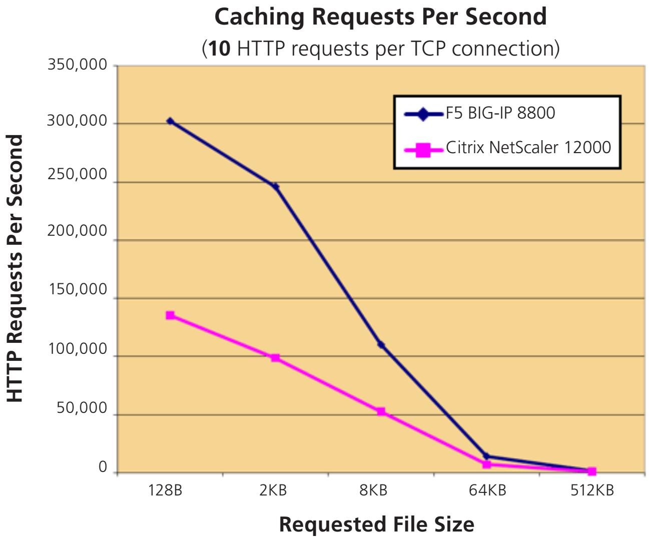
HTTP Caching performance is a measure of how quickly the DUT can serve HTTP responses from its own internal HTTP cache. HTTP Caching helps reduce server load by handling the majority of requests for static content, allowing the servers to focus resources on the business-specific aspect of the application. The use of HTTP Caching can also improve end user performance, because the servers have more free resources to process user requests.

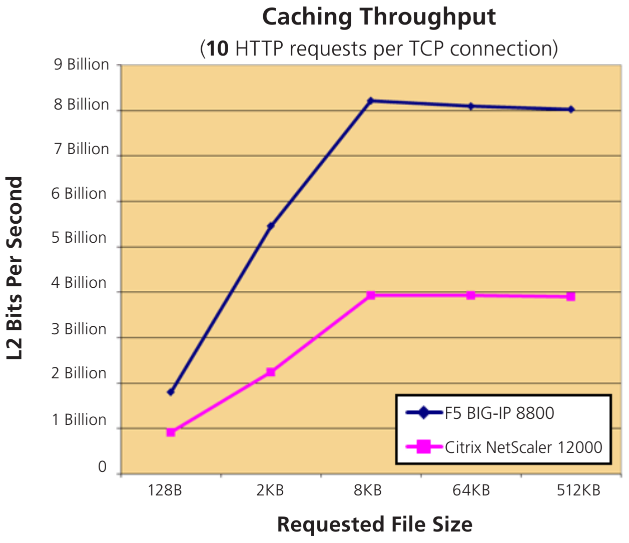
Another valuable, but less obvious, aspect of HTTP Caching is the fact that the devices providing the caching functionality incur lower load increases for serving cached objects compared to requesting content from the backend servers. By serving content from its own cache, a caching device avoids the round-trip-time required when requesting content from the servers, improving response time. HTTP Caching, when coupled with HTTP Compression (processed on the ADC or servers), can further lower resource requirements by reducing the number of times the same content has to be compressed.

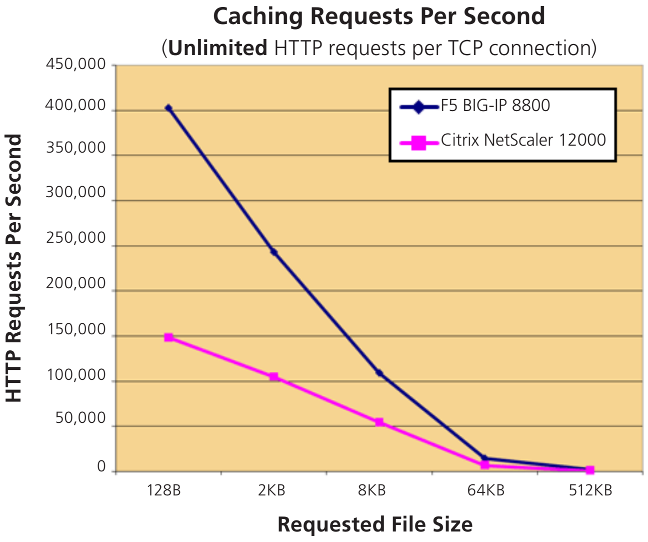
***Note:** Cisco ACE 8GB does not appear in these results because it lacks support for HTTP Caching.

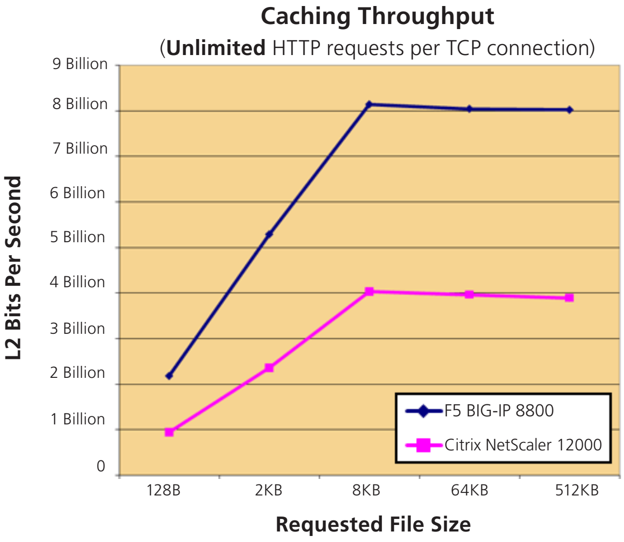

Caching Requests Per Second
(**1** HTTP request per TCP connection)


Caching Throughput
(**1** HTTP request per TCP connection)


Caching Requests Per Second
(**10** HTTP requests per TCP connection)


Caching Throughput
(**10** HTTP requests per TCP connection)





Analysis - HTTP Caching Performance

Many Application Delivery Controllers provide integrated HTTP caching (typically in-memory) used to offload web servers. To provide value, these caches must be very fast (faster than accessing the web server), so it is important to understand their performance characteristics.

With a 128 byte response, the BIG-IP 8800 achieved between 177,000 requests per second (1 request per connection) and 403,000 requests per second (unlimited requests per second), with throughput at 8 Gbps for large responses in all three tests. With 10 or more requests per connection, the BIG-IP 8800 achieved 8 Gbps for file sizes 8KB and larger (8KB, 64KB, and 512KB).

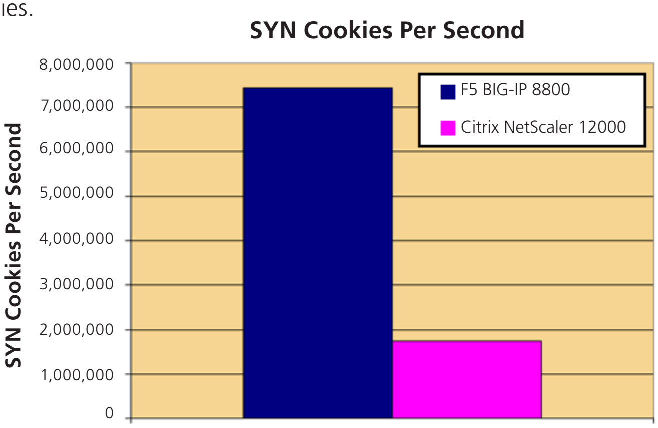
Citrix NetScaler performed 107,000 requests per second with 1 request per connection, and up to 149,000 with unlimited requests per connection. Throughput was 3.9 Gbps in all three tests, starting at a file size of 64KB for the 1 request per connection tests, and 8KB for the 10 and infinite request per connection tests.

As noted previously, Cisco ACE does not support HTTP caching.

Special Test – SYN Cookies

The concept behind SYN cookies, and the value they provide, is very straight forward. TCP connection setup events are resource-intensive; without SYN cookies, a TCP host receiving a single packet with the SYN flag set will go through the resource-intensive connection setup process. SYN cookies provide a mechanism that requires receiving more than one packet from a remote host before connection setup is performed. Implementing SYN Cookies reduces the possibility of a Denial of Service (DoS) attack from hosts sending a flood of SYN packets.

The BIG-IP 8800 includes F5's custom PVA10 ASIC, which has native support for SYN Cookies in the ASIC itself at a very high rate of speed – faster than any other device we have seen. NetScaler does not have specialized hardware to handle SYN Cookies. Cisco ACE does not support SYN Cookies.



Analysis - SYN Cookies

The BIG-IP 8800 generated 7.44 million SYN Cookies per second with 0% CPU utilization.

Citrix NetScaler was able to generate 1.74 million SYN Cookies per second at 100% utilization of the CPU which handles this traffic, an advantage of more than 4x for the BIG-IP 8800.

Published specifications for the Citrix NetScaler 12000 platform suggest it can handle up to 2.3 million "SYN attacks per second". Unfortunately, there doesn't seem to be an official definition for this term ("SYN attacks per second"). Some vendors discuss their performance in the face of SYN attacks as a combination of SYN Cookies they can generate, and selective packet loss. Depending on a variety of factors, this can allow a device to absorb a SYN flood attack that exceeds their ability to generate SYN Cookies, while still allowing them to process some valid traffic. It is not clear what definition would be needed to achieve NetScaler's 2.3 million SYN attacks per second specification.

For more information about how this testing was performed, see *the FAQ entry*.

As noted previously, Cisco ACE does not support SYN Cookies.

Conclusions

The results in this report clearly demonstrate that the BIG-IP 8800 is the fastest Application Delivery Controller on the market today, based on common performance metrics, with performance several times higher than the competition in many instances. To put this in context, in order to achieve the L7 requests per second performance of a single BIG-IP 8800 it would require four Cisco ACE devices and four Citrix NetScaler devices all working together (all eight of them); and they would still be more than 50,000 requests per second away from matching a single BIG-IP 8800.

This report is limited to performance comparisons, and only a common set of capabilities and features. With this in mind, it must be reiterated that performance detached from functionality is not a product strength at all, but merely a weakness disguised as a strength. The BIG-IP system's success in the Application Delivery market has always been heavily dependent on industry-leading functionality like TMOS, iRules, and iControl, combined with industry-leading performance as demonstrated in this report.

More information about the BIG-IP 8800, including ordering information, is available at **www.f5.com**.

The guide for creating performance testing methodologies can be found at DevCentral: *http://devcentral.f5.com/LinkClick.aspx?link=http%3a%2f%2fdevcentral.f5.com%2fdownloads%2ff5%2fcreating-performance-test-methodology.pdf&tabid=73&mid=935*

The configurations for all devices involved in the testing, including the load generation tool and L2 switching infrastructure, is also available at DevCentral: *http://devcentral.f5.com/LinkClick.aspx?link=http%3a%2f%2fdevcentral.f5.com%2fdownloads%2ff5%2fperformance-report-configs.zip&tabid=73&mid=935*

Appendix A: Terminology and Definitions

Term	Meaning
DUT	Device Under Test. DUT generically refers to the device currently being tested. For example, while testing the BIG-IP 8800, DUT refers to the BIG-IP 8800.
Client/Server	In this document, the terms Client and Server refer to clients or servers simulated by Spirent Avalanche/Reflector load testing equipment.
L4	Refers to the minimum set of DUT configuration parameters required to load balance TCP traffic, translating only the destination IP address and TCP port number, without any connection re-use or L7 inspection.
L7	Refers to the minimum set of DUT configuration parameters required to inspect HTTP request URIs, and based on the URI, select between different groups of back-end web servers. The term L7 alone does not imply connection re-use.
Connections Per Second	TCP connection setup (3 packets), HTTP request, HTTP response, and TCP connection teardowns completed in a single second, using Spirent configuration options to close with TCP FIN (instead of RST).
Requests Per Second	HTTP requests and associated responses completed in a single second.
SSL TPS	SSL Transactions Per Second. An SSL transaction is defined as: TCP connection setup, SSL session setup, HTTP request, HTTP response, SSL shutdown, and TCP teardown.
Compression Throughput	HTTP client data sent + HTTP server data sent in a single second.
Application Delivery Controller (ADC)	Generically used to refer to the type of devices tested in this report. ADC's provide functionality such as load balancing, application acceleration, and server offload.
HTTP Request Muliplexing (Connection reuse)	Refers to an ADC's capability to keep an HTTP connection open between the ADC and the server, even while there is no connection between the ADC and the client. Additionally, it specifies that these connections be used to handle as many HTTP requests from as many clients as possible, such that a small number of connections will be open between the ADC and servers, while many connections are opened and closed between the ADC and client. The net result is that servers are offloaded from having to process many TCP connection setup/teardown events. This is called OneConnect™ by F5.
Test Run	The execution of all stated performance tests, one after the next, without any changes to the DUT or test environment (the load generation equipment programmatically executes every test and test variation without interruption).
Time to First Byte (TTFB)	Number of milliseconds (ms) from the time a SYN is sent by the client, until the first byte of data is received from the DUT. Typically used as a measure of connection setup efficiency.
Time to Last Byte (TTLB)	Number of milliseconds (ms) from the time a SYN is sent by the client, until the last byte of data is received from the DUT. Typically used as a measure of overall packet processing efficiency of a device, indirectly measuring the latency added by the device.

Appendix B: Additional Testing Details

High-Level Test Requirements

- Default configuration settings will be used as much as possible.
- Settings for the load generation equipment will be the same for all devices tested.
- A single DUT configuration must be used for all test runs (no changes during testing).
- At least five test runs will be completed for all devices.
- After each test run, the DUT and load generation equipment will be restarted – restarts of the DUT will not occur at any other time.
- Reported test results will be the maximum achieved out of all test runs, and any significant variation between test runs will be noted.
- The DUT will not be connected to for management purposes (CLI, SNMP, Web, etc) during test runs.

Details of Tested Equipment

Vendor	Product	Model	Software Version	Uplink to Switch
F5 Networks, Inc.	BIG-IP LTM	8800	9.4.0	1x 10 Gbps
Cisco Systems, Inc.	Application Control Engine Module (ACE)	8-Gbps	3.0(0)A1(4a)	1x 10 Gbps
Citrix Systems, Inc.	NetScaler Enterprise Edition	12000	8.0 build 45.4	4x 1Gbps

Load Generation Equipment Information

Hardware / Software
All testing was performed with the latest L4-L7 performance testing equipment from Spirent® Communications. The hardware is a Spirent TestCenter 2.0 chassis with 10x L4-L7 blades (CPU_5000A), running software version was 2.00.3102.0010. Six blades were designated clients, and 4 were designated servers.

Settings
As often as possible, default settings were used. Settings that were changed included: Load spec, HTTP version (1.1), max HTTP requests (variable), custom content files (128 byte, 2KB, 8KB, 64KB, 512KB), congestion control was disabled, multiple HTTP servers were configured (to use TCP ports: 80, 81, 8080, 8081), SSL ciphers (RC4-MD5), and of course the Action List (variable). Port settings and associations are boiler-plate. No other baseline settings are changed.

The baseline settings used are available in a *Spirent SPF file*. All tests were automated using the Spirent TCL API. The automated script first incorporates the baseline settings, and then changes the settings required to match the test objectives for each test. The settings changed in the automated script are: Action List (changes the request target and the number of action list items), and the number of HTTP requests per client connection. No other settings are changed by the script. This makes it easy to use the baseline Spirent SPF file to easily recreate the test results of any test without the need of the full TCL API framework or having to re-run all tests.

Appendix C: Frequently Asked Questions

How can these results be trusted, as a device vendor is running the tests?

All device configurations and any information required to reproduce the testing has been made publicly available. All of the devices tested and equipment used in the test infrastructure are available on the open market. This allows any individual or company to reproduce the test results that have been shown in this report. Additionally, it should be noted that for several tests the performance results in this report match or are very near the performance claims from the vendors themselves. Parity with vendor claims is a strong indication that these test results are valid for comparison.

Why do some of the results not match vendor claims?

There can be multiple reasons for this, and determining which reasons apply in a particular circumstance is beyond the scope of this report. Speaking generically across the entire industry (not regarding this report or the vendors in this report), there are several common reasons. For example, different vendors may use different load testing equipment to test their products, where the load testing settings used may be different (and perhaps no equivalent settings are available). In some cases, the definition of terms used by vendors are different, but when using the same definitions, their published claims may be consistent with other public reports. In some cases vendors use non-standard performance testing tools (i.e. internally developed), which cannot be fairly used for comparison. In many cases vendors publish performance claims that are not based on exact test results, but are instead chosen to be a round number, near the achieved results. These issues and many others may be a factor in the differences between vendor claims and the results of public reports. In any case, knowing that all devices were tested with the exact same load testing equipment and settings, these results are at very least accurate relative to each other. For example, the BIG-IP 8800 would have achieved a better compression ratio with a simpler text file, but the same is also true for NetScaler, so regardless of the file chosen the relative performance of the devices is what is most meaningful.

Why report the maximum performance instead of an average of all test runs?

There was no significant difference between the average and maximum performance for any of the devices tested.

Why was SSL session ID reuse not tested?

All three vendors tested in this report use specialized SSL acceleration hardware that provides very similar performance for new sessions compared to re-used sessions. Even with 9 reuses of an SSL session ID, SSL TPS performance increases are typically not more than approximately 15% for a 128 byte response. As file size increases, the performance gain quickly declines to 0%, and may even be slower than no re-use in some cases. Given the very limited usefulness of this information when evaluating performance, it was decided to remove these tests in favor of using the space for expanded analysis of other tests results.

Why were other SSL ciphers not tested?

See the *Why was SSL session ID reuse not tested?* entry above. Briefly stated, the performance difference between various common symmetric SSL ciphers (RC4, 3DES, AES) is rather small with modern SSL acceleration hardware. Additionally, each vendor had very similar relative performance with the various ciphers, so this information was deemed not significantly useful for sizing or comparison, and it was decided to use the space within the report for additional explanatory text.

What's different about the special SYN Cookie test?

The test marked "Special" was beyond the scope of the methodology used for the rest of this report, and we felt it was important that this difference be clearly understood. For example, the CPU of the DUT was monitored during the SYN Cookie test, which was not permitted under the standard methodology. Additionally, the load testing hardware and load testing settings had to be quite different to match the requirements of the test. Although there are several necessary differences, we are equally committed to the repeatability and transparency of this testing. The normal Spirent equipment and settings were used to run a 128 byte L4 test, with one change: the maximum attempted traffic was 1,000 connections per second. This Spirent traffic was used to verify that some regular traffic would succeed during the DoS attack. Next, an Ixia® 1600T chassis with two TXS4-256MB cards was used to generate the SYN traffic (using IxExplorer software). For each DUT, the IP port and MAC address of the L4 virtual server was programmed as the destination into a stream on 5 separate interfaces, with each stream using only the minimum parameters required to simulate a SYN packet from a single source IP with incrementing TCP port numbers. Four interfaces were turned on at 20% of line rate to start, then increased in 1% increments until the rate of valid packets received was more than 5% different from the send packet rate (indicating the DUT was falling behind). With four interfaces, NetScaler reached its limit at 29% of line rate on four interfaces. In the case of the BIG-IP 8800, after reaching 100% on four interfaces, a fifth interface was added and ramped in the same way. This test only measures the number of SYN Cookies that can be generated (measured as the sum of the packet receive rate from all interfaces), and does not attempt to determine the maximum number of SYN's that could be sent to the DUT while still processing some valid traffic.

How were the number of users and other load generation equipment settings decided upon?

The load generation equipment was tested without a DUT, in a back to back configuration to determine optimal settings. With the help of our test equipment vendor, many tests were run with very slight variations, until the settings were decided to be optimal, or very near optimal for the load generation equipment. These settings are not optimized for any of the devices tested, because test requirements do not allow for differing test settings for each device, and it wouldn't be fair to pick one DUT as a baseline. The downside to this approach is that each DUT may not achieve is maximum potential. After some testing with load generation settings varied for each DUT, it was decided that the performance differences were not significant enough to be of value. Additionally, every environment will have significant variations from every other environment, so the choice of settings for this testing is no better and no worse than many other possible choices. The load specification was "SimUsers", and was tested at both 624 and 6,240.

Why was Citrix NetScaler only tested with a 4 Gbps connection?

Citrix NetScaler does not provide a 10 Gbps interface, and supports a maximum of 4 gigabit interfaces per link aggregation group. The test environment is designed to simulate a common and simple customer deployment with all clients routed into a single external VLAN, and then delivered to servers which all reside on a single internal VLAN. With a limit of 4 interfaces per link aggregation group, there was no way to make use of additional interfaces. Public information from the Citrix web site suggests this platform is capable of a maximum of 6 Gbps (not tested), still less than that demonstrated on the F5 BIG-IP and Cisco ACE in throughput-bound tests. Also note that most requests per second metrics were not affected by a 4 Gbps throughput limit.

Why was Cisco ACE tested with only an 8 Gbps license (updated October 2007)?

At the time of testing Cisco ACE was only available with licenses up to 8 Gbps. In some press releases and marketing collateral beginning in April 2006, Cisco mentioned a 16 Gbps version of ACE. As of June 2007, Cisco ACE data sheets, release notes (June 2007), product manuals and Cisco Global Price List all indicate that only 4 Gbps and 8 Gbps are available.

Update October 2007: F5 has learned that a 16Gbps version of ACE became generally available for purchase later in June 2007, and has been widely advertised since the middle of September 2007, including updated product data sheets that now provide ordering information for 16Gbps. This new data sheet for Cisco ACE indicates that L4 throughput can be as high as 16Gbps with newer "ACE20" hardware and a new license. This new data sheet shows no other changes in performance. L4 connections per second, SSL TPS, etc, remain the same as previous claims. Accordingly, this performance report remains valid for comparison against the latest Cisco ACE in all respects but L4 throughput. Cisco's L4 throughput claim is 16Gbps. This has not been validated.

Why did Cisco ACE have poor SSL performance, particularly at 512KB response?

At the time of testing, there were several known issues with the Cisco ACE product that did not have workarounds (documented in Cisco ACE release notes) affecting high throughput SSL or large response size. Cisco ID "CSCsh19382" says SSL throughput is limited to 3.4 Gbps, "CSCsh54479" says SSL throughput drops below 4 Gbps, "CSCsj37029" says stress testing with large pages may result in poor SSL performance. The first issue mentioned was documented in Cisco release notes in December 2006. The latest software release (dated June 2007) had not yet resolved this issue.

Why does the NetScaler configuration include servers on more ports (80, 81, 8080, 8081)?

Test objectives specify separate tests for HTTP request multiplexing (connection re-use), separate tests for HTTP compression, separate tests for HTTP caching, and so on. The test methodology also requires that the DUT configuration not be altered between tests, meaning that a single configuration must work for all tests. The Citrix product has the ability to specify when certain features are used (such as compression and request multiplexing) on a per-server basis (or "service" in Citrix NetScaler parlance). With the combination of features required, it was necessary to have four different services defined for each server, on TCP ports 80, 81, 8080, and 8081. Like other products, Citrix provides the option to enable/disable some features based on L7 decisions. It was found that NetScaler performed better with the multiple service definitions, so this faster configuration was used.

What type of Supervisor module and line card was used with Cisco ACE?

Cisco ACE was housed inside of a Catalyst 6509 chassis, with a Supervisor Engine 720 (WS-SUP720-3BXL), connected to the main switch via a CEF720 4 Port 10-Gigabit Ether line card (WS-X6704-10GE).

Why did Citrix NetScaler not achieve 275,000 HTTP requests per second?

The public data sheet for Citrix NetScaler 12000 suggests that performance up to 275,000 HTTP requests per second is possible with this platform. The results in this report did not reproduce this claim. Citrix NetScaler has published an HTTP requests per second performance figure for several years, and in researching this, it was found that Network Computing magazine reviewed both the BIG-IP system and NetScaler in 2005. Similarly, both were evaluated in Network World magazine review in 2006. The results from both of these publications were much closer to the results found in this report, compared to the number mentioned in the Citrix NetScaler data sheet. In this report, Citrix NetScaler achieved a maximum of 148,757 HTTP requests per second (during the caching test), which is still much lower than reported on the spec sheet.

In researching this issue, a test was conducted with Citrix NetScaler configured using the L4 configuration (no HTTP inspection, no per-request load balancing, no HTTP request multiplexing), with the test equipment configured like the L7 Unlimited requests per connection test. Using these settings, results in the range of 270,000-300,000 requests per second were informally observed. Using the same test, both the BIG-IP 8800 and Cisco ACE were informally observed achieving over 1,000,000 HTTP requests per second.

Why did Cisco ACE not achieve more than 260,000 L4 connections per second?

According to the latest release notes from Cisco available at the time (June 2007), a known caveat tracked by Cisco ID "CSCsd80505" limits performance to this level, unless tests are designed to not exceed a maximum of 330,000 connections per second. Testing requirements do not allow for each device to have their own load testing settings. Additionally, it is not valid to artificially limit the maximum amount of attempted traffic, as this would not be possible in a production deployment. This issue was first published in Cisco release notes from December 2006.

Why aren't the request per second numbers in the analysis sections more precise?

The request per second numbers in this report have been rounded to the third digit for easier readability. For example, 123,123 is rounded to 123,000 in accordance with significant digits theory. This has a negligible impact on statistical accuracy, typically less than 1%, and never more than 1.5%.

F5 Networks, Inc.
Corporate Headquarters
401 Elliott Avenue West
Seattle, WA 98119
(206) 272-5555 Voice
(888) 88BIGIP Toll-free
(206) 272-5556 Fax
www.f5.com
info@f5.com

F5 Networks
Asia-Pacific
+65-6533-6103 Voice
+65-6533-6106 Fax
info.asia@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
+44 (0) 1932 582 000 Voice
+44 (0) 1932 582 001 Fax
emeainfo@f5.com

F5 Networks
Japan K.K.
+81-3-5114-3200 Voice
+81-3-5114-3201 Fax
info@f5networks.co.jp