# A Simplified Application Acceleration Architecture

Most organizations want to improve web application performance for their users, but they can't always justify the cost of implementing acceleration techniques. F5 offers a simple, low-risk, and low-cost acceleration solution that can significantly improve the end-user experience.

White Paper

by F5

# Introduction

Why are some websites slow? Why don't organizations implement solutions to accelerate their applications? It's certainly not the lack of user demand. Everyone likes faster applications. No one likes waiting for a page to load. Many authors have expended time and resources creating studies to demonstrate the economic or behavioral impact of poorly performing applications. They have all concluded that faster page-load times are better. Was there ever any doubt about this? Why, then, do these studies they exist? It's simple: they are there to justify the cost and overhead of solving web application performance problems.

What if the cost of accelerating web applications could be dramatically reduced? Organizations that formerly dismissed application acceleration as too difficult or costly could deliver the application acceleration that their users want, and give them the end-user experience they deserve.

In this white paper we give an overview of networks and applications, describe basic principles of web application optimization, and then examine the benefits and drawbacks of various acceleration techniques—including F5's. We then propose a simplified web application acceleration solution that lowers the operational and commercial barriers to bringing improved end-user experiences to everyone.

# Why There Is Still an Application Performance Problem

## Bandwidth and Latency

In 1998, Jakob Nielsen proposed that a high-end user's connection bandwidth grows by 50 percent each year. His prediction turned out to be accurate; examples show that there has been a near-linear growth in bandwidth from 1983 to 2013.[1]

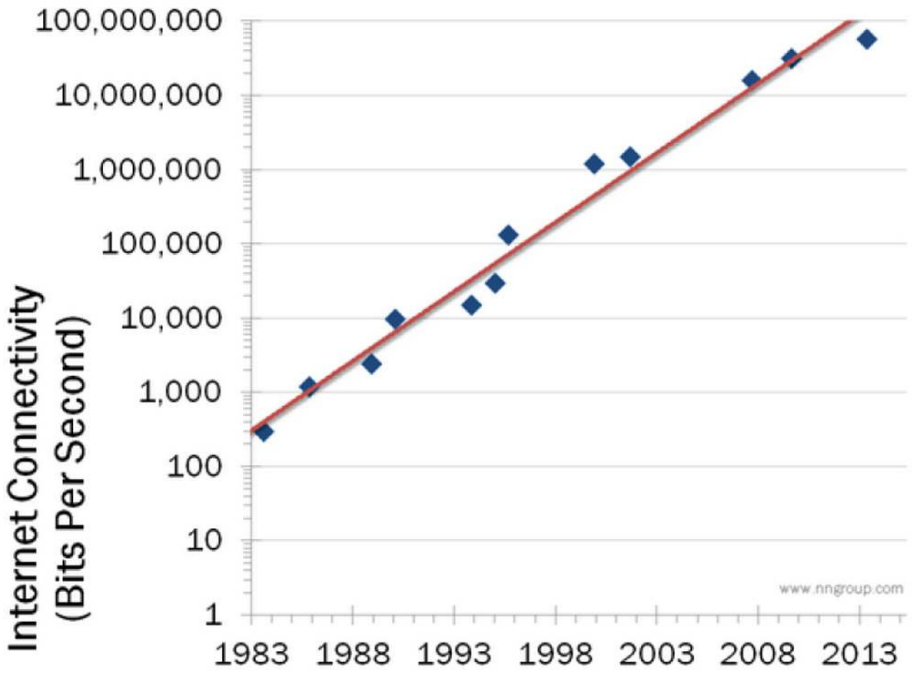


Figure 1: Bandwidth has continued to grow at a near-linear rate of approximately 50 percent per year.

With this growth in bandwidth, why is web application performance still an area of concern? While increased bandwidth is important for the transfer of large files, such as images, movies, and PDF documents, the nature of HTTP-delivered applications makes them far more sensitive to latency. The good news is that, up to a point, increased bandwidth has meant decreased latency, and drops in latency tend to produce a direct improvement in page load times.[2] As bandwidth increases above a threshold of around 5 Mbps, however, the decrease in latency becomes minimal.[3] Increasing bandwidth, therefore, does not automatically improve the end-user experience.

Another contributing factor is the increase in the number of mobile devices and wireless connections. This has resulted in an increase in latency. Whereas fixed-access connections, such as DSL or cable, typically exhibit 25-45 ms of latency, 3G and 4G wireless networks are in the range of 100-150 ms.[4] With this increase in latency comes a linear increase in page-load times. This results in a penalty of 1.3 to 2 times the page-load time of fixed-line connections.[5] In addition, bandwidth for mobile users is significantly lower than for fixed- line users. When mobile users access applications designed for fixed-line users, the result is that the combination of richer pages, higher latency, and lower bandwidth results in an experience that is not what users expect.

## Web Application Complexity

Having reviewed the influence of bandwidth and latency on application performance, we should examine the application itself. From 2012 to 2014, the total transfer size of the top 1000 Internet URLs has approximately doubled.[6] During the same period, the number of objects on a page has also increased. Larger pages with more objects require both more network round trips and greater bandwidth to load the page.

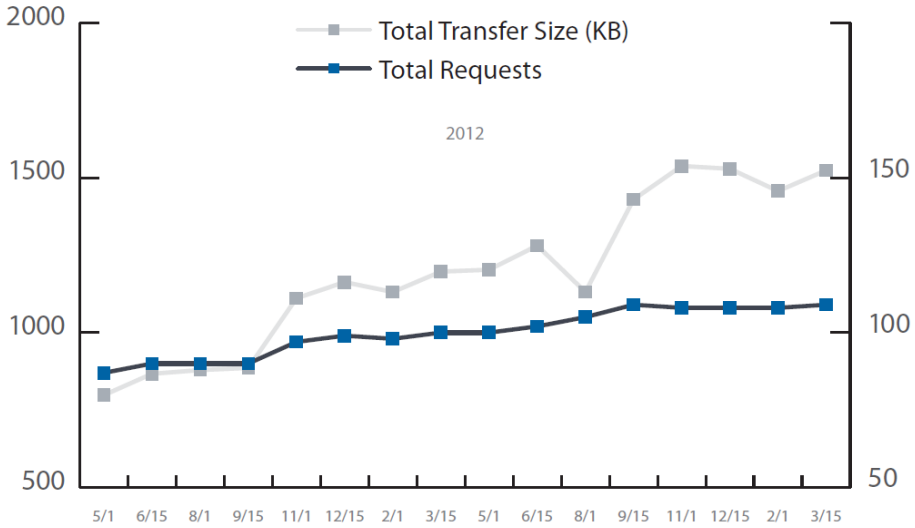


Figure 2: Page size and number of objects per page have continued to grow.[7]

The latency-sensitive nature of applications, increased use of mobile devices, and increased application complexity combine to keep application performance and end-user experience as an area of concern for organizations, despite the growth in bandwidth.

# Key Principles to Accelerating Web Applications

We will use the term "web application" to refer to any application rendered primarily using HTML and delivered from the origin server using HTTP/HTTPS. This could include a public website or an internal business application. The aim of accelerating these web applications is to improve the end-user experience with faster page-load times and more responsive interactivity. A faster-performing web application results in greater sales, better user engagement, and improved productivity.

Accelerating websites can be reduced to three essential principles:

- Send data as efficiently as possible
- Send data as infrequently as possible
- Send as little data as possible

## Send Data as Efficiently as Possible

Despite a range of optimization and compression techniques, data must still be sent to and from the client and the server. Sending this data as efficiently as possible can make significant improvements in end-user experience. Optimizing the transfer of data over the network can be done both at the transport layer through TCP optimization and at the application layer by optimizing HTTP requests using the SPDY protocol.

Optimizing the TCP connection for the client network conditions can produce impressive results. In tests using mobile 3G and 4G networks, we have found that page load times decrease an average of 7 percent to 29 percent between various different locations simply by optimizing the TCP connection separately for clients and servers.

The SPDY protocol, which is the starting point of HTTP 2.0, manipulates HTTP traffic with the goal of reducing page-load times and increasing security (SSL/TLS encryption is required to use the protocol). Using SPDY has been shown to improve page-load times due to the more efficient transfer mechanism of the SPDY protocol, which multiplexes multiple HTTP requests on a single TCP connection. While the measured page load improvements vary, nearly all studies show a worthwhile improvement in page-load times using SPDY.[8][9][10]

## Send Data As Infrequently As Possible

The fewer network connections and requests a client has to make to assemble a page, the faster it will load. Cutting out additional requests and connections can be achieved by techniques such as content inlining (for example, embedding the data to create an image in the HTML page rather than as an external link) or browser cache manipulation, using techniques such as F5 Intelligent Browser Referencing (IBR). IBR implements object versioning for cachable objects, such as images and scripts. Versioned objects provide the benefits of caching at the browser while still ensuring that content is not stale. Caching objects for an extended time removes the need for the browser to revalidate content. HTTP requests that simply check if content is still valid can add significantly to page-load times, even if no new data is transferred from the server.

## Send As Little Data As Possible

Reducing the data that the server needs to send to the client will result in a reduced network transfer time and faster page load. Reducing the amount of data transmitted can be done by eliminating redundant data from the server responses themselves or by compressing redundant data before it's sent.

Web application server software or a third-party device such as an Application Delivery Controller (ADC) can compress HTTP objects. Moving HTTP object compression to the ADC or other optimization device allows additional content manipulation and potentially more efficient compression. One of the advantages of using the SPDY protocol is the compression of the HTTP headers, which provides an incremental reduction in data transferred.

Removing unnecessary data from server responses can be done by removing whitespace or comments from text files, such as JavaScript or CSS, or by transforming images into a more efficient file format and stripping image metadata. While the gains in removing whitespace from individual files might be only small, the incremental impact is measurable, given the number of objects commonly required to load a page.

## Practical Implementations

How can organizations implement these acceleration techniques? How should they decide which method will be the best for them? Like any other project, when assessing acceleration, organizations need to evaluate the return on investment of any solution. A three-day project that results in an 8 percent performance increase might produce significantly more ROI than an eight-week application recoding project that results in a 20 percent performance increase.

### Improving Application Code

Optimizing application code is a viable solution for some applications. Applying the key principles outlined here (and in many books and articles) to web application code can help to eliminate many performance problems right at their source. Optimizing application code has some significant barriers, however. To successfully optimize an application code, organizations must:

- Have access to the code
- Possess or purchase the development skills to make the changes
- Commit to the operational overhead of testing and application rollout
- Make a commitment to continued tuning; it is not a one-and-done situation

These factors dramatically reduce the return on investment of optimization projects by significantly increasing the cost of improving application performance. Other techniques such as TCP optimization will still be required to efficiently transport the application traffic—again requiring skills and operational effort.

## Content Delivery Networks

A content delivery network (CDN) is an infrastructure that helps organizations deliver static web application content to users faster by being physically closer to them. Application content is replicated to multiple locations and users are directed to the nearest available site. This ensures that content is served to the user as quickly as possible, and it can eliminate much of the latency involved in delivering static application data. CDN use has grown considerably over the past years and content served from CDNs has been reported to account for up to 50 percent of the web application traffic delivered in North America.[11]

Clearly, many organizations are benefiting from CDN use for Internet content delivery. Large organizations with multiple data centers can build private a CDN to distribute internal application traffic.[12]

Using a CDN does not come without some drawbacks. Public CDN delivery of SSL objects poses challenges in certificate management: organizations must either hand over their keys or allow customers to load third-party, signed content on a page that is supposedly secured by the customers' certificate. In addition, SPDY support is lacking in most CDNs, and the use of CDNs can remove the HTTP multiplexing benefit of SPDY connections.

The subscription model adopted by most CDN providers can result in variable and unpredictable costs as the charges vary with the volume of content served. These charges can rapidly become significant as content volumes increase.

## Application Optimization Appliances or Software

Third-party web application acceleration software or appliances can provide a huge range of tools and techniques to accelerate web applications. A rich feature set of protocol and application manipulation tools enables organizations to control multiple aspects of application delivery and may extend down into TCP optimization layers. These tools can apply different acceleration techniques to sites, URIs, and objects to achieve fine-grained control of application behavior.

The F5 BIG-IP Application Acceleration ManagerTM and BIG-IP Local Traffic ManagerTM modules, for instance, contain a comprehensive feature set of tools, all of which can be further configured and customized to provide highly granular and highly effective application acceleration policy. To illustrate this, figure 3 lists some of the more common technologies available and a brief explanation of their function.

| Function | Technique | Explanation |
|---|---|---|
| Efficiency | TCP optimization | Manages client connections and server connections separately using state-of-the-art TCP optimization algorithms and techniques to improve performance across real-world networks. |
| | SPDY Gateway | Presents HTTP backend servers over SPDY to compatible browsers, resulting in reduced latency and page load times. |
| | Caching | Caching frequently used objects in the fabric enables direct delivery to the client without requesting them from the application servers. |
| Infrequency | Content inlining | Combines multiple objects into a single request/object so that the number of HTTP requests required is reduced, allowing for faster delivery of multiple objects on a single TCP connection. |
| | Intelligent Browser Referencing | Increases the efficiency of the client web browser's local cache and improves perceived access to an application site by reducing or eliminating requests for relatively static content, such as images and CSS files. |
| | Intelligent Client Cache (ICC) | A web acceleration technique for mobile and desktop browsers that support HTML5. ICC uses HTML5 local storage to build a cache of documents and resources within the client's browser. |
| Reduction | Compression | Compresses HTTP objects to transmit less data between clients and servers. |
| | Image optimization | Commonly provides 30-45 percent bandwidth savings for image files without degrading the user experience. |
| | Content minification | Removes comments and whitespace from script files to provide an incremental reduction in data transmitted. |
| Other | Content reordering | Improves apparent page load times by optimizing the order in which the browser requests objects. |
| | MultiConnect | Overcomes the limited number of TCP connections created by older browsers to allow up to 4x the number of TCP connections, resulting in reduced time to transfer content from server to client. |

Figure 3: The features and functions of common acceleration technologies.

These features and functions enable the building of application acceleration policies that help organizations solve the most complex and difficult application optimization problems. Using the right tools and techniques can release the best possible performance obtainable from the available network and application code.

In common with every other solution we have examined, there are drawbacks to be considered. Some architecture designs may require additional hardware or software, resulting in training and administration overhead. Ideally these functions should be consolidated onto existing infrastructure components, such as Application Delivery Controllers. The wide range of tools and techniques can themselves bring some disadvantages. Creating custom policies to accelerate applications takes time and a deep understanding of the application's design and behavior. Using advanced manipulation techniques, such as content reordering or inlining, can require considerable testing and validation with application development teams.

What about situations where the benefits to the organization won't justify the overhead of implementing a custom policy? The usual result is that applications are denied access to these services.

F5 believes no application should be left behind. The F5 Synthesis™ acceleration architecture offers organizations a solution to provide the right level of acceleration to applications combined with a simplified business model to neutralize acquisition cost constraints.

## A Simplified Solution

Many organizations and applications can benefit substantially from a combination of simple and low-risk application- and transport-layer application acceleration services.

By consolidating functions onto a high-performance platform such as the F5 BIG-IP system, organizations can combine security and availability services and seamlessly integrate application acceleration. Delivering application acceleration from a single platform creates efficiencies for both infrastructure and personnel. Consolidating to a single platform does not mean that all acceleration services need be delivered from a single device, since the F5 Synthesis high-performance application services fabric enables customers to build a highly available, all-active fabric. Application service workloads can be delegated to the most appropriate resources in the fabric.

Organizations can create a policy that accelerates nearly all HTTP applications, eliminating the need to select only the high-priority or high-value applications. In fact, many applications can be accelerated or optimized by deploying just three simple techniques.

## Choose an Optimized TCP Stack

By deploying a client-side TCP stack that is optimized for the delivery network or client device, organizations can gain significant application performance benefits. Most application acceleration projects involve situations where latency is affecting the end-user experience. Traditional congestion- control algorithms have been focused solely on packet-loss mitigation, but clients are increasingly accessing applications over mobile networks, where latency presents a greater challenge. F5 TCP Express™ supports the most recent developments in congestion control, focused on mitigating latency problems and enabling new innovations such as multipath TCP (MPTCP) support.

## Enable the Use of the SPDY Protocol

With the three most common browsers (Internet Explorer, Chrome, and Firefox) now all offering support for the SPDY protocol,[13] organizations should be looking to offer SPDY access to web applications. SPDY is an alternative application-layer protocol that overcomes the inherent inefficiencies in HTTP. Working with SPDY today can provide advantages as HTTP 2.0 becomes the web standard in the coming months. In many cases, the user experience is significantly improved by using SPDY due to reduced number of separate requests for content and reduced latency. The F5 SPDY Gateway enables clients to use the SPDY protocol and gain significant improvements in page-load times without requiring organizations to make any changes to web application servers.

## Use the F5 Fundamental Acceleration Policy

The F5 fundamental application acceleration policy deploys a set of simple application-layer acceleration tools that have been designed to optimize application performance with low risk and few or no configuration requirements. The fundamental policy is preconfigured to supply a range of simple but effective application acceleration tools. These techniques provide tangible acceleration benefits without significantly altering application data, making them suitable for rapid deployment and rapid return on investment.

The key advantage of this architecture design is that it can be applied to most HTTP applications without significant configuration or testing overhead. These techniques can be enabled wherever web applications are delivered and can provide an enhanced application user experience for everyone.

## Additional Acceleration Options

There are two additional application acceleration techniques that can further accelerate applications and do not generally require significant testing or configuration. They represent a sensible second step when additional acceleration is required.

### Image optimization

Optimizing images can result in reduced image file sizes (but not pixel count or displayed size). This is done by reducing the color count, stripping image metadata, and using more efficient compression formats. This typically results in a greater than 30 percent reduction in image size. For nearly all applications, the users will notice no difference in image quality and will enjoy a faster page-load time.

### Intelligent Browser Referencing

Intelligent Browser Referencing (IBR) sets long object expiration dates but does so intelligently by rewriting the object's URL to contain a checksum of the object's contents. At the same time, it maintains or establishes a short lifetime setting for the HTML page that references the objects. When the browser wants to see if there is updated content, it re-requests the main HTML page. If nothing has changed, the browser receives the same main HTML page as before and then loads all objects from its cache since they were served with a long expiration time. However, if an object has changed, it is assigned a new object URL, which is reflected in the main HTML page. The browser receives a new HTML page with a different URL only for the object that has changed. The browser requests only that single object and then loads the remaining objects from its cache. This technique eliminates wasteful HTTP requests to revalidate content, but removes any risk of stale content being rendered by the browser.

Of course, some applications and environments will require a greater range of techniques to solve their application acceleration challenges. With the F5 application acceleration architecture, customers can start with a basic policy and continue to add additional techniques, such as page reordering, content minification, or inlining, until the desired level of application performance is achieved. This represents a low-risk and high-return approach to an acceleration project. Organizations can see results rapidly and projects can progress in easy stages. With the power of the programmable BIG-IP platform, techniques such as A/B testing can be rapidly implemented to evaluate policy changes and effectiveness.

## A Simplified Business Model

The simplified business model that F5 Synthesis architectural vision introduced reduces the costs of delivering services so customers can increase the value of a rich and extensible set of services for all devices, networks, and applications. In support of this vision, F5 has introduced a tiered licensing model to consolidate purchasing options for its solutions and significantly reduce customer costs when application services are deployed in concert versus individually. This enables IT to focus on delivering the services applications require, rather than on the acquisition costs of obtaining them.

## Conclusion

The F5 application acceleration architecture enables organizations to deliver an improved user experience for all web applications and users. A low-risk yet powerful feature set enables organizations to apply a universal policy to most web applications, providing acceleration services with very little administrative overhead. Combining this streamlined policy with the simplified business model puts substantial application acceleration capabilities within reach of all organizations while still providing further options to solve the toughest of performance problems.

---

[1] Nielsen's Law of Internet Bandwidth, Nielsen Norman Group, 2013.

[2] Latency: The New Web Performance Bottleneck, Igvita.com, July 2012.

[3] Measuring Broadband America, Federal Communications Commission, February 2013.

[4] 3G/4G wireless network latency: How do Verizon, AT&T, Sprint and T-Mobile compare?, FierceWireless, November 2013.

[5] March 2014 Bandwidth Report, Website Optimization, March 2014.

[6] Interesting Stats, HTTP Archive.

[7] Trends, HTTP archive.

[8] A comparison of SPDY and HTTP performance, Microsoft Research, July 2012.

[9] SPDY Performance on Mobile Networks, Google Developers, April 2012.

[10] Not As Speedy As You Thought, Guy's Pod, June 2012.

[11] Massive Ongoing Changes in Content Distribution, Content Delivery Summit, Spring 2013.

[12] Building a CDN with F5, F5 Networks, April 2013.

[13] SPDY networking protocol compatibility tables, caniuse.com.

F5 Networks, Inc.
401 Elliott Avenue West, Seattle, WA 98119
888-882-4447 f5.com

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com