

Upgrade to core HTTP protocol promises speedier, easier web

February 20 2015, by Peter Maynard



Credit: AI-generated image ([disclaimer](#))

Hypertext Transfer Protocol, HTTP, is a key component of the world wide web. It is the communications layer through which web browsers request web pages from web servers and with which web servers respond with the contents of the page. Like much of the internet it's been around for decades, but a [recent announcement](#) reveals that [HTTP/2](#), the [first](#)

[major update in 15 years](#), is about to arrive.

The original HTTP protocol was the protocol first used by Sir Tim Berners-Lee at CERN where the [web](#) was created in 1991. This was improved over many years and finalised as [HTTP 1.1](#) in 1999, the current standard used worldwide. Over the years the web has changed dramatically, introducing images, complex style sheets and Javascript code, Flash and other embedded elements and more. The original HTTP was a simple protocol for a simple web, it was not designed to handle increasingly media-rich websites.

For example, Google handles [40,000 web searches per second](#) every day. To handle the pressure of serving billions of internet users, the company's technicians launched a project in 2009 called [SPDY](#) (pronounced "speedy") to improve HTTP. Originally only for internal use, other sites fielding heavy traffic such as Twitter, Facebook, Wordpress and CloudFlare also implemented SPDY having seen its performance improvements.

This caught the attention of the Internet Engineering Task Force ([IETF](#)), which develops and promotes internet standards. IETF decided to use SPDY as the basis for HTTP/2 in 2012 – and the two protocols were developed in parallel. Even though Google spearheaded the protocol's development, the work is continued by the IETF's open working groups as it has done for other protocols for more than 30 years.

Google recently [announced](#) it was dropping SPDY in favour of the soon-to-arrive HTTP/2.

The drawbacks of HTTP 1.1

Web pages today can generate many requests for images, CSS style

sheets, video and other embedded objects, off-site adverts, and so on – perhaps a hundred of these per page. This adds unnecessary strain to the [web server](#) and slows the web page loading time because HTTP 1.1 only supports one request per connection.

HTTP 1.1 is sensitive to [high latency connections](#) – those with a slow response time. This can be a big problem when working on a mobile device using cellular networks, where even a high-speed connection can feel slow. [HTTP pipelining](#) allows the browser to send another request while waiting for the response of a previous request. While this would go some way to tackling high latency, it is susceptible to [problems of its own](#) and is disabled by default in most browsers.

The benefits of HTTP/2

Rather than using clear text, HTTP/2 is now a binary protocol which is quicker to parse and more compact in transmission. While HTTP 1.1 had four different ways to handle a message, HTTP/2 reduces this to one. To tackle the multiple request issue HTTP/2 allows only one connection per site but using [stream multiplexing](#) fits many requests into a single connection. These streams are also bi-directional, which allows both the web server and browser to transmit within a single connection. Each stream can be prioritised, so browsers are able to determine which image is the most important, or prioritise a new set of streams when you change between browser tabs.

HTTP is a stateless protocol – every connection comprises a request-response pair unconnected to any connections before or after. This means every request must also include any relevant data about the connection – this is sent in [HTTP headers](#). As HTTP 1.1 evolved, the headers have grown larger as they incorporate new features. HTTP/2 uses header compression to shrink this overhead and speed up the connection, while [improving security](#).

A final addition is [server push](#). When a web page is requested, the server sends back the page, but must wait for the web browser to parse the page's HTML and issue further requests for things it find in the code, such as images. Server push allows the server to send all the resources associated with a page when the page is requested, without waiting. This will cut a lot of the latency associated with web connections.

Web version 2?

Once web servers and [web browsers](#) start implementing HTTP/2 – which could be as soon as a few weeks from now – the web-browsing experience will feel quicker and more responsive. It will also make developers' lives easier by not having to work around the limitations of HTTP 1.1.

In fact, some of the latest versions of popular browsers (Firefox v36, Chrome v40 and Internet Explorer v11) already support HTTP/2. For Chrome and Firefox, HTTP/2 will be used only over encrypted connections (SSL) – this, along with the [Let's Encrypt](#) initiative, will probably boost the adoption of encryption more widely.

This story is published courtesy of [The Conversation](#) (under Creative Commons-Attribution/No derivatives).

Source: The Conversation

Citation: Upgrade to core HTTP protocol promises speedier, easier web (2015, February 20) retrieved 3 May 2024 from <https://phys.org/news/2015-02-core-http-protocol-speedier-easier.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is

provided for information purposes only.