



* PERFORMANCE

HTTP/2

Michael Gooding on why the latest version of HTTP will be a game-changer

➤ HTTP/2 has arrived and its goals are simple: to improve the performance of HTTP by targeting the way the protocol is used today. In other words, to load websites even more quickly. The new standard has a host of features that address today's web usage patterns. I'll explore them here.

MULTIPLEXING

Multiplexing is a way of requesting and receiving more than one web element at a time, and it is the cure for the head of line blocking in HTTP/1.1. Currently, each client request needs to wait until the server's response to the previous request arrives. If a request stalls, the whole page download is delayed.

HTTP/2 is a binary framed protocol that means requests and responses are broken up into chunks called frames, with meta information that identifies the request/response they are associated with. This allows objects to overlap on the same connection without causing confusion, and for them to be sent/received in any order. A first request may take longer to complete but it won't hold up the delivery of any subsequent objects, meaning faster page load.

HEADER COMPRESSION

Headers have grown in size. Using a combination of lookup tables and Huffman encoding, HTTP/2 can reduce the number of bytes sent in a request down to zero. In an average web session, compression rates above 90 per cent are not uncommon.

With HTTP/1.1, a modest page with 75 objects and an average header size of 500 bytes might take the browser four TCP round trips just to request the objects. The new protocol doesn't improve the size of responses, but on the request side the results are significant. With the same parameters and 90 per cent compression with HTTP/2, a browser can send all the requests in a single round trip.

DEPENDENCIES AND PRIORITISATION

Browsers have introduced pre-loaders to ensure they request the most important stuff first. If, in the new model, a browser simply requests everything at the same time and allows the server to return objects as quickly as possible, there will ironically be a reduction in page performance, because key objects required for rendering may not be delivered first. The designers of HTTP/2 have addressed this: by communicating to the server what objects are dependent on other objects and listing the priorities, the server can make certain the critical data is delivered to the browser right away.

SERVER PUSH

Server Push allows the server to send the browser an object before it is asked for, providing instant page delivery even in the worst conditions. However, in order to push the correct objects without wasting valuable bandwidth, the server needs to know what the user is likely to need next, and what the state of the browser cache is. Whilst HTTP/2 provides the tool for Server Push today, I am sure we will see some innovative uses over the coming years.

... AND FOR THE EVERYDAY USER?

No one will have to change their website or applications to ensure they continue to work properly. HTTP/2 is compatible with existing code and APIs, with the one major difference being you will need to deliver using TLS if you're not already. Organisations should, however, start thinking about how they can capitalise on the new performance and security capabilities. **n**

Read the extended version at netm.ag/http2-271

*** PROFILE** Former developer Michael (@Michael_G_81) works at Akamai to identify performance bottlenecks and solve complex problems for some of the biggest websites in the world