

---

# Chapter 6. Server Push—A Killer Feature

Server push in SPDY means something very different than what web developers have meant for years. It has nothing to do with

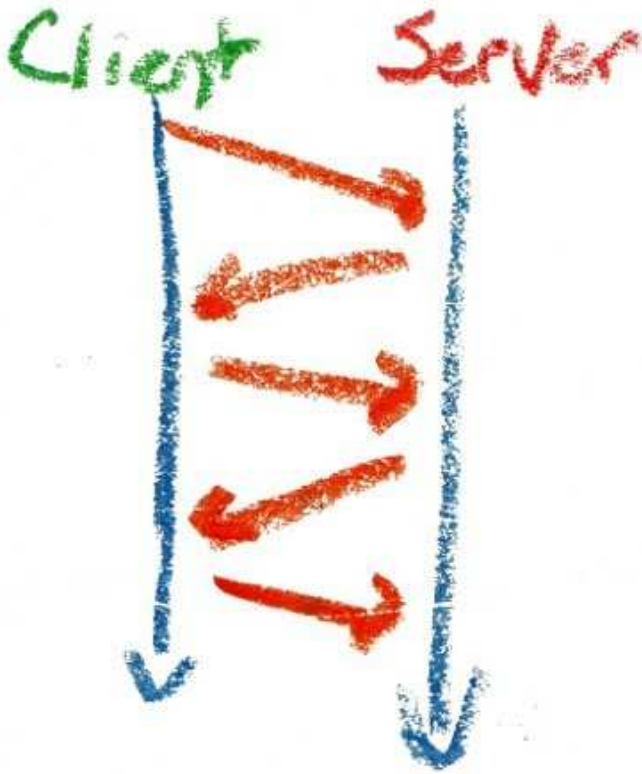
- Comet
- AJAX Long Polling
- Websockets

Rather think back to the sole purpose of SPDY — to decrease page load time. The longer the throbber throbs, the more likely a potential client is to hit the Back button.



With that in mind, SPDY server push means to push additional responses along with the normal response to a request.

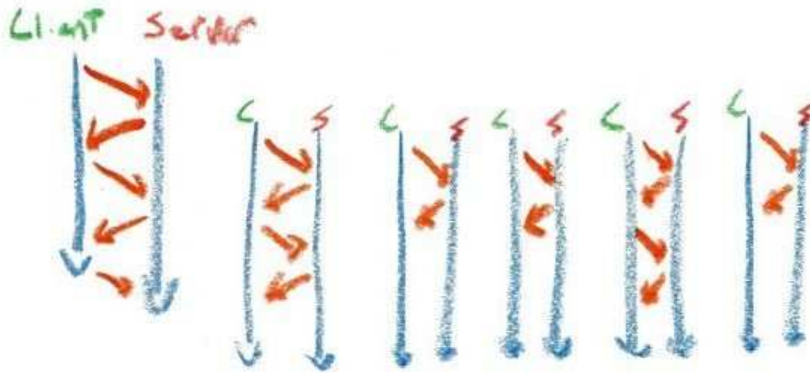
In vanilla HTTP, a request for a web page might look something like:



The web browser makes a request of a web server, which responds with a web page. Based on the contents of that web page, the browser will have to make subsequent requests of the server to ask for stylesheets, images, *etc.*.

If browsers only had a single pipe on which to talk to the server, the conversation would take a very long time. As described in Chapter 5, *The Basics*, HTTP does not allow asynchronous conversation, so the browser has to wait for a response before beginning a new request.

All modern browsers get around this by initiating up to 6 pipes to the web server:



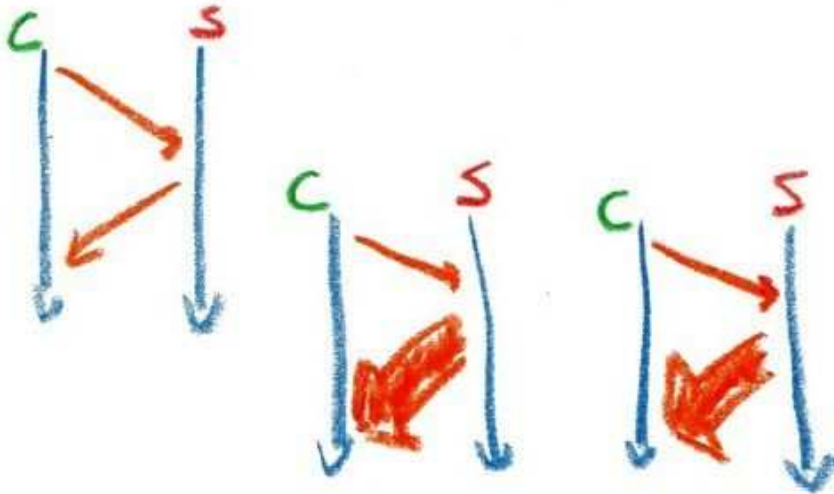
Each of those pipes still incurs the limitation of synchronous request/response pairs. If a web page has more than 6 additional resources that need to be loaded (typical modern sites can have 40+), then the browser will still be waiting a significant amount of time for round-trip latency on each of these pipes.

Additionally, as discussed in Chapter 5, *The Basics*, each of these pipes has to go through the same warm up period as the TCP/IP Cwnd is negotiated. The more pipes, the more likely there will be contention between the pipes, and the more likely it is that a network error will impact one of them.

Lastly, note that the secondary requests cannot begin until at least some of the web page response has come through. All modern web browsers are capable of making secondary requests once they have read the <head> section of a page, and know what CSS and Javascript will be needed. Even so, that does not begin until the response is at least underway.

To get around some of these limitations, web sites have, in recent years, begun to package multiple stylesheets into a single larger stylesheet. It is possible to do the same with Javascript files and images (the latter in the form of sprites).

Asset packaging makes HTTP conversations look more like:



An obvious deficiency with this approach is that the browser still needs to wait for the first response before it can know what to request subsequently.

Also problematic with asset packaging is additional maintenance. Asset packaging requires non-trivial amounts of configuration and management to ensure that these resources are combined and served properly.

## 6.1. SPDY Server Push

SPDY solves this problem by pushing additional resources along with the requested resource. If a browser requests a site's homepage, SPDY provides the capability to push along all of the images, stylesheets and Javascript files at the same time: