

Currently, HTTP and the web in general exist solely as a technology or system that can pull information from servers. For instance, when the user clicks a link on a webpage, an http request is sent, a response received and the entire page redraws. Now, using javascript, code that runs in the browser, these requests can be made asynchronously. A website can request any url like normal, but when the response is returned, rather than redrawing the entire page, the response from the server can be used to execute javascript code to systematically update the page. A good example of this, sports web scoreboard. The webpage can periodically send a request to the server to update the score for a game. This significantly reduces network traffic, because only the update score needs to be retransmitted.

The problem with this system is that the website must continually poll for or request update information. This leads to a large amount of unnecessary traffic. A smarter system would be one where the server can send updated information to every client connected to the webpage whenever the score changes. This can further reduce the unnecessary network traffic, and it can produce updates that are more real-time. The problem is there is currently no single good way of using http to push updates to clients in the browser. My topic would be to research some various implementations of http push and analyze the pros and cons of each.

I am open to doing this as a presentation, but I think I would prefer to do this as a paper.