acmqueue

HTTP/2.0 — The IETF is Phoning It In

Bad protocol, bad politics

Poul-Henning Kamp

A very long time ago—in 1989—Ronald Reagan was president, albeit only for the final 19½ days of his term. And before 1989 was over Taylor Swift had been born, and Andrei Sakharov and Samuel Beckett had died.

In the long run, the most memorable event of 1989 will probably be that Tim Berners-Lee hacked up the HTTP protocol and named the result the “World Wide Web.” (One remarkable property of this name is that the abbreviation “WWW” has three times as many syllables and takes longer to pronounce.)

Tim’s HTTP protocol ran on 10Mbit/s, Ethernet, and coax cables, and his computer was a NeXT Cube with a 25-MHz clock frequency. Twenty-six years later, my laptop CPU is a hundred times faster and has a thousand times as much RAM as Tim’s machine had, but the HTTP protocol is still the same

A few days ago the IESG, the Internet Engineering Steering Group, asked for “Last Call” comments on the new “HTTP/2.0” protocol (<https://tools.ietf.org/id/draft-ietf-httpbis-http2>) before blessing it as a “Proposed Standard”.

EXPECTATIONS WILL VARY

Some will expect a major update to the world’s most popular protocol to be a technical masterpiece and textbook example for future students of protocol design. Some will expect that a protocol designed during the Snowden revelations will improve their privacy. Others will more cynically suspect the opposite. There may be a general assumption of “faster.” Many will probably also assume it is “greener.” And some of us are jaded enough to see the “2.0” and mutter “Uh-oh, Second Systems Syndrome.”

The cheat sheet answers are: no, no, probably not, maybe, no and yes. If that sounds underwhelming, it’s because it is.

HTTP/2.0 is not a technical masterpiece. It has layering violations, inconsistencies, needless complexity, bad compromises, misses a lot of ripe opportunities, etc. I would flunk students in my (hypothetical) protocol design class if they submitted it. HTTP/2.0 also does not improve your privacy. Wrapping HTTP/2.0 in SSL/TLS may or may not improve your privacy, as would wrapping HTTP/1.1 or any other protocol in SSL/TLS. But HTTP/2.0 itself does nothing to improve your privacy. This is almost triply ironic, because the major drags on HTTP are the cookies, which are such a major privacy problem that the EU has legislated a notice requirement for them. HTTP/2.0 could have done away with cookies, replacing them instead with a client-controlled session identifier. That would put users squarely in charge of when they want to be tracked and when they don’t want to—a major improvement in privacy. It would also save bandwidth and packets. But the proposed protocol does not do this.

The good news is that HTTP/2.0 probably does not reduce your privacy either. It does add a number of “fingerprinting” opportunities for the server side, but there are already so many ways to fingerprint via cookies, JavaScript, Flash, etc., that it probably doesn’t matter.

You may perceive webpages as loading faster with HTTP/2.0, but probably only if the content provider has a global network of servers. The individual computers involved, including your own, will have to do more work, in particular for high-speed and large objects like music, TV, movies, etc. Nobody has demonstrated a HTTP/2.0 implementation that approached contemporary wire speeds. Faster? Not really.

That also answers the question about the environmental footprint: HTTP/2.0 will require a lot more computing power than HTTP/1.1 and thus cause increased CO₂ pollution adding to climate change. You would think that a protocol intended for tens of millions of computers would be the subject of some green scrutiny, but surprisingly—at least to me—I have not been able to find any evidence that the IETF considers environmental impact at all—ever.

And yes, Second Systems Syndrome is strong.

Given this rather mediocre grade sheet, you may be wondering why HTTP/2.0 is even being considered as a standard in the first place.

THE ANSWER IS POLITICS

Google came up with the SPDY protocol, and since they have their own browser, they could play around as they choose to, optimizing the protocol for their particular needs. SPDY was a very good prototype which showed clearly that there was potential for improvement in a new version of the HTTP protocol. Kudos to Google for that. But SPDY also started to smell a lot like a “walled garden” to some people, and more importantly to other companies, and politics surfaced.

The IETF, obviously fearing irrelevance, hastily “discovered” that the HTTP/1.1 protocol needed an update, and tasked a working group with preparing it on an unrealistically short schedule. This ruled out any basis for the new HTTP/2.0 other than the SPDY protocol. With only the most hideous of SPDY’s warts removed, and all other attempts at improvement rejected as “not in scope,” “too late,” or “no consensus,” the IETF can now claim relevance and victory by conceding practically every principle it ever held dear in return for the privilege of rubber-stamping Google’s initiative. But the politics does not stop there.

The reason HTTP/2.0 does not improve privacy is that the big corporate backers have built their business model on top of the lack of privacy. They are very upset about NSA spying on just about everybody in the entire world, but they do not want to do anything that prevents them from doing the same thing. The proponents of HTTP/2.0 are also trying to use it as a lever for the “SSL anywhere” agenda, despite the fact that many HTTP applications have no need for, no desire for, or may even be legally banned from using encryption.

YOUR COUNTRY, STATE, OR COUNTY EMERGENCY WEBPAGE?

Local governments have no desire to spend resources negotiating SSL/TLS with every single smartphone in their area when things explode, rivers flood, or people are poisoned. Big news sites similarly prioritize being able to deliver news over being able to hide the fact that they are delivering news, particularly when something big happens. (Has everybody in IETF forgotten CNN’s exponential traffic graph from 14 years ago?)

The so-called “multimedia business,” which amounts to about 30% of all traffic on the net, expresses no desire to be forced to spend resources on pointless encryption. There are even people who are legally barred from having privacy of communication: children, prisoners, financial traders, CIA analysts, and so on. Yet, despite this, HTTP/2.0 will be SSL/TLS only, in at least three out of four of the major browsers, in order to force a particular political agenda. The same browsers, ironically, treat self-signed certificates as if they were mortally dangerous, despite the fact that they offer secrecy at trivial cost. (*Secrecy* means that only you and the other party can decode what is being communicated. *Privacy* is secrecy with an identified or authenticated other party.)

History has shown overwhelmingly that if you want to change the world for the better, you should deliver good tools for making it better, not policies for making it better. I recommend that anybody with a voice in this matter turn their thumbs down on the HTTP/2.0 draft standard: It is not a good protocol and it is not even good politics.

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

POUL-HENNING KAMP (phk@FreeBSD.org) is one of the primary developers of the FreeBSD operating system, which he has worked on from the very beginning. He is widely unknown for his MD5-based password scrambler, which protects the passwords on Cisco routers, Juniper routers, and Linux and BSD systems. Some people have noticed that he wrote a memory allocator, a device file system, and a disk-encryption method that is actually usable. Kamp lives in Denmark with his wife, son, daughter, about a dozen FreeBSD computers, and one of the world’s most precise NTP (Network Time Protocol) clocks. He makes a living as an independent contractor doing all sorts of stuff with computers and networks.

© 2015 ACM 1542-7730/15/0100 \$10.00