HTTP/3

HTTP/3 se está implementando en producción para todos nosotros, y eso podría ser importante para usted, para mí y para Internet.

HTTP es el protocolo (valga esta redundancia) que hace que la web funcione. Fue descrito (e implementado) por primera vez en 1989 (ya usted lo sabe: Tim Berners-Lee y su equipo y el equipo del CERN, todo eso es historia contemporánea), pero fue especificado formalmente como HTTP/0.9 en 1991.

HTTP/0.9

HTTP/0.9 era un protocolo simple construido sobre TCP, usando cabeceras de texto plano, sin cifrado, con funcionalidad muy limitada, pero haciendo su tarea lo suficientemente bien como para esas inicios de la red de redes.

HTTP fue un gran éxito, con el navegador web Mosaic, luego Netscape (uno de los pioneros en código abierto) y en 1995 el Microsoft Internet Explorer (y su juicio por monopolio de la industria). Y millones de sitios web, comercio electrónico, etcétera.

HTTP/1.0

La gente empezó a añadir alguna funcionalidad no estándar tanto en el lado del cliente como en el del servidor (los servidores principales en ese momento eran NCSA HTTPd -más tarde conocido como Apache-; servidor Netscape; Microsoft IIS), y el HTTP 1.0 fue producido en 1996 (RFC 1945).

HTTP/1.1

Pero mientras tanto, la gente ya estaba trabajando en HTTP 1.1, por lo que muchas de sus características ya estaban soportadas por clientes y servidores en 1996. Fue aprobado formalmente en 1997 como RFC 2068, y actualizado en 1999 con 176 páginas en la RFC 2616.

HTTP 1.1 fue el protocolo que hizo posible el crecimiento explosivo de la web a finales de los 90 y principios de los años 2000, aún cuando fue hecho para TCP y con cabeceras de texto plano. Pero se podía usar con cifrado, una sola conexión TCP con varias interacciones, y soportaba muchos mecanismos adicionales como la RFC 2617 (incluidas con sus fe de erratas correspondientes).

HTTP 1.1 fue tan exitoso (y los intereses sobre HTTP eran tan variados y poderosos), que llegó hasta el 2014 sin actualización.... ¡seguía siendo HTTP 1.1! :

- Sintaxis y encaminamiento de mensajes (RFC 7230).
- Semántica y Contenido (RFC 7231).
- Peticiones condicionales (RFC 7232).
- Rengo de peticiones (RFC 7233).
- Almacenamiento en *caché* (RFC 7234).
- Autenticación (RFC 7235).
- Esquema inicial de registro de autenticaciones (RFC 7236).
- Registro inicial de método HTTP (RFC 7237).
- El código de estado 308 -redirección permanente (RFC 7238).
- Extensión de reenvío de HTTP (RFC 7239).
- Cabecera preferida para HTTP (RFC 7240).

Esta renovación de HTTP 1.1 tuvo en cuenta las prácticas y demandas de la industria, y trató de poner orden en un período en el que la web era la aplicación dominante en Internet, y necesitaba estar a la altura de ese estándar. Pero era obvio que se necesitaba una reforma más drástica.

Google desde el año 2009, con el apoyo de otros actores importantes, había estado trabajando en SPDY y cambió al HTTP de algunas maneras significativas: era binario (olvídese de las cabeceras de texto sin formato), soportaba la multiplexación de conexiones TCP, compresión de cabeceras y varios otros cambios.

Para 2012, Chrome, Firefox y Opera ya soportaban SPDY, y algunos sitios web grandes también (por ejemplo Facebook, Google y Twitter). Estaba en su camino para convertirse en un estándar de facto. Su principal ventaja fue que redujo la mucho la latencia .

En el sitio web de la Fundación Mozilla existe en idioma castellano un excelente artículo que explica el funcionamiento de HTTP/1.1.

HTTP/2.0

Pero en lugar de dejar que SPDY se convirtiera en un norma de hecho, tras una convocatoria abierta se utilizó como a SPDY como base para HTTP/2, una nueva y brillante versión del HTTP. Fue lanzada en 2015 bajo las RFC 7540 (HTTP/2) y RFC 7541 (HPACK).

Después de este lanzamiento de los nuevos HTTP/2, Google detuvo el desarrollo de SPDY, y los esfuerzos volvieron a las "normas oficiales".

HPACK, por cierto, es una adición muy interesante a HTTP/2. Hace codificación Huffman de encabezados, reduciendo significativamente el tamaño (especialmente para peticiones con una carga útil pequeña) y es más resistente a los ataques dirigidos a olfatear *galletitas*.

HTTP/2 ha tenido mucho éxito en su adopción, aunque pocos usuarios lo han notado (excepto por un mejor rendimiento web para ellos). Las razones principales son probablemente la multiplexación de las peticiones HTTP en conexiones TCP individuales, y la compresión de encabezados. Pero HTTP/2 no era lo suficientemente radical. En un camino que ya puede ser familiar, Google comenzó a trabajar en un rediseño muy completo de HTTP: el protocolo QUIC. QUIC es radical en varios sentidos, pero probablemente lo más notable es su uso de UDP como capa de transporte, en lugar de TCP. Esto significa que QUIC tuvo que implementar la fiabilidad y el orden, al igual que TCP junto a TLS, sin embargo lo hace a su manera muy particular.

HTTP/3

Así como sucedió con SPDY y HTTP/2, QUIC se convirtió en la base de HTTP/3. Actualmente, HTTP/3 está todavía en fase de realización, pero se espera que sea estable hasta que esté totalmente normalizado. Ya está soportado por algunos navegadores web (Chrome y próximamente Firefox), así como importantes programas y librerías como cURL.

Hoy, Cloudflare ha anunciado que está soportando HTTP/3 en sus servidores. Dado que Cloudflare está ejecutando más del 10% de todos los servidores web del mundo esta es una muy buena noticia para HTTP/3. Cloudflare ha dado el nombre clave QUICHE a su desarrollo particular del HTTP/3, acá su repositorio en GitHub.

[SVG: QUICHE (https://github.com/cloudflare/quiche)]

QUICHE (https://github.com/cloudflare/quiche)

Situación actual

Al momento de escribir estas líneas hay abundante material (en idioma inglés) para aprender e investigar sobre esta innovación tecnológica:

- The Internet changes: HTTP/3 will not use TCP anymore.
- HTTP/3 explained.
- The next version of HTTP won’t be using TCP.