

Protocolos para Mejorar la Performance de las Aplicaciones Web

Marcelo F. Fernández¹, Fernando Tinetti², Gabrel H. Tolosa¹

mail@marcelofernandez.info, fernando@info.unlp.edu.ar, tolosoft@unlu.edu.ar

¹Departamento de Ciencias Básicas, Universidad Nacional de Luján.

²Facultad de Informática, Universidad Nacional de La Plata.

Resumen

El protocolo HTTP ha sido revisado en un par de ocasiones, la última, ya hace más de una década. Sin embargo, las características de la Web actual, los requerimientos de sus usuarios, y los niveles de masividad hicieron que los recursos que se proveen mediante éste hayan alcanzado un punto donde se han encontrado algunas limitaciones inherentes al diseño original del protocolo. Es por ello que el IETF (*Internet Engineering Task Force*), dentro del *HTTPbis Working Group*, está analizando modificaciones, ajustes y/o mejoras de fondo en pos de alcanzar un futuro estándar HTTP 2.0.

Como una propuesta reciente se presentó el protocolo SPDY, cuyo objetivo primordial es mejorar el rendimiento del servicio Web. Hoy en día se ha convertido en la base sobre la cual dicho WG está trabajando.

Este proyecto de investigación plantea estudiar las deficiencias mencionadas de HTTP y las mejoras disponibles, entre los cuales se encuentra SPDY. Además, producir un estudio académico de algunas de las condiciones que optimicen su implementación, tanto desde el punto de vista técnico y general como además involucrando un análisis de condiciones geográficas. Por último, se desarrollarán piezas de software que faciliten dichos análisis y sirvan como herramientas de apoyo para una eventual migración a SPDY o a HTTP/2.0.

Palabras clave: HTTP, SPDY, protocolos web, www.

Contexto

Esta presentación corresponde al proyecto de investigación “Estudio y propuestas de protocolos para mejorar la performance de servicios basados en Web”, aprobado por el

Departamento de Ciencias Básicas de la Universidad Nacional de Luján, para el cual el primer autor cuenta con una beca de investigación. categoría “perfeccionamiento” de la Secretaría de Ciencia y Técnica de la mencionada universidad.

Introducción

La Internet de hoy es una Red global omnipresente, donde el crecimiento cuasi-exponencial de la demanda de servicios de información ha “exprimido” en buena medida las capacidades y posibilidades de todos los protocolos que han sido pilares fundacionales de la misma desde un principio.

El protocolo HTTP, que pertenece a este selecto grupo y es uno de los protagonistas principales en el uso de la red, cuenta ya con más de dos décadas de haberse desarrollado (la versión 0.9 en 1991) y más de una de haber visto su última revisión, la versión 1.1 en 1999 [12].

Pero, según varias investigaciones [1,4,20], adolece de debilidades bajo ciertas condiciones, además de algunos problemas de larga data [3, 6,18,25], que han sido potenciados por la riqueza (multimedia, interactividad, etc.) contenida y provista mediante las páginas web actuales.

Esto tiene diversas consecuencias, donde por un lado impacta directamente en el tamaño, cantidad y variedad de recursos que HTTP debe transportar.

Y por el otro, el funcionamiento mismo de HTTP v1.1, utilizando conexiones persistentes (*Keep-Alive*) más *Pipelining* (ambos para optimizar el flujo de datos a nivel de TCP), sumado a múltiples conexiones en paralelo para incrementar la velocidad en la descarga, no es suficiente para conseguir transferencias

eficientes de una página web más todos los recursos que ésta referencia, desaprovechando la capacidad de la red que se dispone y afectando tanto a usuarios como proveedores [5,15].

Si bien su característica de *stateless* permite alta flexibilidad y escalabilidad, el uso de *Cookies* para mantener el estado es percibido usualmente como error de diseño [17]. Además, obliga a que los clientes web utilicen técnicas de *polling* para obtener notificaciones del servidor, cuando sería más eficiente que éstos notifiquen al navegador cuando sea necesario. Asociado con el punto anterior, pero también con la complejidad de la web en la actualidad, la cantidad de encabezados por cada petición HTTP se ha incrementado considerablemente, aumentando la cantidad de tráfico a transportar.

Por otro lado, la falta del soporte de prioridades en la obtención de recursos también puede verse como una desventaja que originalmente no existía por la sencillez de las páginas y servicios web. Y, al ser un protocolo definido mediante directivas en formato de texto ASCII, el procesamiento computacional de sus órdenes y validación de su especificación es más complejo y propenso a errores que si fuera binario. Las redes móviles y satelitales bajo estas nuevas condiciones son casos extremos y particulares que deben ser analizados en detalle.

Finalmente, la falta de confidencialidad, en esta etapa de plena vigencia de redes *wireless*, es otra problemática subsanada por TLS (*Transport Layer Security*) [11], pero que no ha reemplazado a HTTP, sino que la ha complementado de manera opcional.

Por ende, a nivel mundial, se considera que es momento de revisar si el funcionamiento de la web a este nivel puede ser modificado, mejorado, renovado y/o optimizado de acuerdo a los tiempos que corren [7,8,10]; aprovechando la confiabilidad y velocidad de los enlaces sobre los equipos de conectividad de los que se dispone hoy en día, transportando el tipo y volúmenes de recursos Web actuales.

Aún así, esta revisión debe llevarse a cabo respetando las enormes virtudes de las especificaciones preexistentes, y sin descuidar asuntos sumamente importantes como la compatibilidad hacia atrás, complejidad de implementación, facilidades para la migración,

etc., a fin de maximizar las posibilidades de resistir las diferentes pruebas en el campo, para que finalmente se defina una versión HTTP/2.0.

Es por eso que desde que fue presentado el protocolo SPDY como un proyecto abierto [7], a fines de 2009, ha convocado interés de muchos investigadores y organizaciones fuera de Google, su principal impulsor, dado que persigue los siguientes objetivos de diseño:

- Reducir al mínimo la latencia necesaria para la descarga de recursos HTTP.
- Reducir al mínimo el *overhead* (y por consiguiente el ancho de banda) en la subida y descarga de recursos HTTP.
- Minimizar la complejidad de instalación e implementación.
- No requerir ningún cambio en los contenidos de los sitios web.

SPDY, en su versión actual (número 3) [2], presenta las siguientes características técnicas como más salientes:

- Multiplexación de peticiones HTTP concurrentes en una única sesión TCP.
- Encabezados HTTP comprimidos.
- Niveles de prioridades en las peticiones de recursos HTTP.
- *Server-Pushed Streams*: Recursos enviados desde el servidor al cliente sin existir una petición previa exclusiva y asociada de éste.
- Implementa una capa de compatibilidad con HTTP, preservando su semántica: *Cookies*, *headers*, *encoding*, etc., funcionan de la misma manera que HTTP.
- Fuertemente acoplado con HTTP; la integración de protocolos forma parte de la especificación. Sin embargo, SPDY puede ser utilizado para transportar otros protocolos.
- Podría considerarse ubicado sobre TCP y entre SSL/TLS y HTTP, en la capa 5 (Sesión) del modelo OSI.
- En la práctica funciona sobre TCP y TLS/SSL, aunque no está atado a éstos; sí requiere un transporte confiable de mínima.

Esto permitiría, en principio:

- Mayor Escalabilidad: poder manejar más cantidad de usuarios y atenderlos más rápido desde el punto de vista del proveedor de contenidos.
- Velocidad: que los usuarios obtengan una mejora sustancial percibible en la carga de páginas y recursos Web.
- Mayor Seguridad: ya que de facto se utiliza TLS para la implementación.
- Compatibilidad hacia atrás: migración incremental y transparente para los usuarios.

Sin embargo, los estudios y análisis formales realizados hasta el momento, tanto por parte de SPDY como de otras propuestas desarrolladas en paralelo [10,16,19,29] son exiguos, y generalmente provienen del sector industrial, lo que da origen a esta propuesta de investigación académica.

Líneas de Investigación y Desarrollo

Entre los diferentes trabajos que se están llevando adelante se encuentran tres líneas principales:

- **Estudio y análisis de las debilidades de HTTP** y bajo qué circunstancias ocurren, basándose tanto en trabajos en curso como en experimentación propia. Además, se investiga el protocolo SPDY, los requerimientos para su funcionamiento y algunas de las ideas de las alternativas, tomando en consideración las ventajas y desventajas que presenta frente al protocolo HTTP.
- **Modelización de la red a nivel local y regional.** La intención aquí es poder realizar pruebas y experimentación sobre condiciones de operación de la red representativas de la región ya que todos los estudios actuales reportan resultados sobre redes de condiciones sustancialmente mejores pertenecientes a los principales países.
- **Desarrollo de librerías de código** tanto para dar soporte a los experimentos, documentar ejemplos, y desarrollar los objetivos de las fases posteriores [31]. Se pretende implementar programas para facilitar la migración a SPDY, como servicios intermediarios (proxy) y validadores de conformidad con las especificaciones

propuestas, como también una guía de recomendaciones a tener en cuenta para estudiar e implementar estos cambios.

Resultados y Objetivos

Se consiguió desarrollar un muxer/demuxer de frames SPDY (versión 2 y 3) con ejemplos básicos en sockets TCP y TLS sobre TCP en el lenguaje Python, con el fin inicial de comprender mejor lo estudiado, más allá de perseguir los objetivos mencionados anteriormente [31].

Basado en los resultados de las etapas iniciales, se propondrán ajustes sobre los protocolos estudiados y se realizarán pruebas de campo (redes locales e Internet), a nivel local y regional. Además, se propondrán e implementarán mecanismos migración a SPDY y/o HTTP/2.0

Algunos resultados iniciales sugieren que SPDY permite disminuir en tiempo de carga total de una página entre un 39%-55% accediendo por un canal cifrado y entre un 27%-60% por un canal no cifrado [30].

Sin embargo, aún no está claro cómo los diversos factores (de la red y de las páginas) inciden en esta mejora. Se espera poder definir un modelo que permita predecir la reducción del tiempo de carga de acuerdo a ciertas características de las páginas, lo que permitiría establecer recomendaciones que apunten a mejorar la performance (como las hay para diseño y posicionamiento, por ejemplo).

Formación de Recursos Humanos

El presente trabajo corresponde a una beca de investigación del primer autor, quien – además – se encuentra finalizando su maestría en Redes de Datos de la Facultad de Informática de la UNLP.

Complementariamente, se prevé incorporar al menos un pasante alumno al proyecto durante el año en curso y realizar un trabajo final de licenciatura en el área.

Referencias

- [1] Mike Belshe, "More Bandwidth Doesn't Matter (much)". Google Inc. 2010.
<https://docs.google.com/viewer?pid=sites&srcid=Y2hyb21pdW0ub3JnfGRldnxneDoxMzcyOWI1N2I4YzI3NzE2>
- [2] Mike Belshe, Roberto Peon, "SPDY Protocol v3", Google Inc., IETF Draft, 2012.
- [3] Yogesh Bhole and Adrian Popescu. 2005. "Measurement and Analysis of HTTP Traffic". *J. Netw. Syst. Manage.* 13, 4 (December 2005), 357-371.
- [4] Michael Butkiewicz, Harsha V. Madhyastha, and Vyas Sekar. 2011. "Understanding website complexity: measurements, metrics, and implications". In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference (IMC '11). ACM, New York, NY, USA, 313-328.
- [5] Nandita Dukkupati, Tiziana Refice, Yuchung Cheng, Jerry Chu, Tom Herbert, Amit Agarwal, Arvind Jain, and Natalia Sutin. 2010. "An argument for increasing TCP's initial congestion window". *SIGCOMM Comput. Commun. Rev.* 40, 3 (June 2010), 26-33.
- [6] Roy T. Fielding. "Waka: A replacement for HTTP". 2002.
http://gbiv.com/protocols/waka/200211_fielding_apachec_on.ppt
- [7] The Chromium Projects, "SPDY: An experimental protocol for a faster web". Google Inc., 2009.
<http://dev.chromium.org/spdy/spdy-whitepaper>
- [8] Preethi Natarajan et al, "Using SCTP as a Transport Layer Protocol for HTTP", Cisco Systems, IETF Draft. 2009.
- [9] IETF, "Httpbis Working Group", 2012.
- [10] IETF, "Httpbis HTTP/2.0 Proposals", 2012.
- [11] IETF Standard, "The TLS Protocol Version 1.0", 1999.
- [12] IETF Standard, "Hypertext Transfer Protocol – HTTP/1.1", 1999.
- [13] A. Langley, "Transport Layer Security (TLS) False Start", Google Inc., IETF Draft, 2010.
- [14] A. Langley, "Transport Layer Security (TLS) Next Protocol Negotiation Extension", Google Inc., IETF Draft, 2012.
- [15] Robert L.R. Mattson, Enhancing HTTP to Improve Page and Object Retrieval Time With Congested Networks. La Trobe University. 2008.
- [16] Robert L. R. Mattson and Somnath Ghosh. 2009. "HTTP-MPLEX: An enhanced hypertext transfer protocol and its performance evaluation". *J. Netw. Comput. Appl.* 32, 4 (July 2009), 925-939.
- [17] Jeffrey C. Mogul. 1995. "The case for persistent-connection HTTP". *SIGCOMM Comput. Commun. Rev.* 25, 4 (October 1995), 299-313. DOI=10.1145/217391.217465.
- [18] Thomas M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. 1997. "Exploring the bounds of web latency reduction from caching and prefetching". In Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems(USITS'97). USENIX Association, Berkeley, CA, USA, 2-2.
- [19] Gabriel Montenegro et al., "HTTP Speed+Mobility", Microsoft Corp., IETF Draft, 2012.
- [20] Jean-Francois Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems", H. Federrath (Ed.), Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, Proceedings, LNCS 2009.
- [21] Steven Rosenberg, Surbhi Dangi, Isuru Warnakulasooriya. "Data and Network Optimization Effect on Web Performance". Carnegie Mellon University, Feb 2012.
http://repository.cmu.edu/silicon_valley/91/
- [22] Marshall T. Rose, "BEEP: The Definitive Guide", 2002. ISBN: 978-0596002442.
- [23] Chris Strom, "The SPDY Book: Making Websites Fly", 2011.
- [24] W. Tarreau et al, "Proposal for a Network-Friendly HTTP Upgrade", IETF Draft, 2012.
- [25] Venkata N. Padmanabhan, Jeffrey C. Mogul, "Improving HTTP latency", *Computer Networks and ISDN Systems*, Volume 28, Issues 1–2, December 1995, Pages 25-35, ISSN 0169-7552, 10.1016/0169-7552(95)00106-1.
- [26] "MUX Overview". W3C. 1998.
- [27] "SMUX Protocol Specification". W3C. 1998.
- [28] Greg White. "Analysis of Google SPDY and TCP Nitewnd". Cable Television Laboratories, Inc., 2012.
http://www.cablelabs.com/downloads/pubs/Analysis_of_Google_SPDY_TCP.pdf
- [29] Wenxuan Zhou, Qingxi Li, Matthew Caesar, and P. Brighten Godfrey. 2011. "ASAP: a low-latency transport layer". In Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies (CoNEXT '11). ACM, New York, NY, USA, Article 20, 12 pages.
- [30] Bryce Thomas, Raja Jurdak, and Ian Atkinson. 2012. SPDYing up the web. *Commun. ACM* 55, 12 (December 2012), 64-73. DOI=10.1145/2380656.2380673
- [31] Marcelo Fidel Fernández, biblioteca Python-SPDY para Python versión 2.6 a 3.3.
<https://github.com/marcelofernandez/python-spdy>