

Proxy Adaptativo para Protocolos Web Avanzados

Licenciatura en Sistemas de Información

Pablo Maximiliano Lulic

22 de septiembre de 2013

1. Introducción

El protocolo HTTP tuvo su primera versión en mayo de 1996, culminando en 1999 con el estándar actual que es el HTTP 1.1. Es un protocolo sin estado, lo que conlleva a que se necesite realizar una conexión nueva por cada recurso que se necesite en un sitio.

Los sitios web de la actualidad difieren de las páginas de hace ms de 10 años, tanto en tamaño como en cantidad de recursos. Este fenómeno hace que el protocolo ya no tenga el mismo rendimiento que en épocas anteriores. A pesar del avance de la tecnología en cuanto a mejoras en las velocidades de los enlaces de red, un ancho de banda grande no es el único factor que interviene en la performance de la carga de las páginas.

Google propuso un protocolo llamado SPDY que busca mejorar la performance de HTTP. Permite múltiples streams en una conexión simple de TCP, además de otras características como push y hint. Aprovechando los resultados favorables de las mediciones de este nuevo protocolo, se busca realizar un proxy que maneje SPDY hacia los nodos interiores, y que hacia la world wide web maneje un algoritmo inteligente que pueda determinar según el perfil del sitio que método (HTTP, HTTPS ó SPDY) elegir para mejorar la performance.

2. El Proxy

2.1. Precondiciones del Software

1. Va a escuchar peticiones en una red LAN y estará conectado a Internet.
2. Se va a comportar como un servidor intermediario (*proxy*) para cualquier cliente de la LAN aceptando conexiones HTTPS y manejando el protocolo SPDY.
3. Para cualquier página en la Web, puede conectarse utilizando cualquier método (HTTP, HTTPS ó SPDY), optando por el más adecuado.
4. Almacenará los recursos que han solicitado los clientes (*caché*) para mejorar la performance en peticiones posteriores.
5. Requiere que el cliente de la LAN soporte SPDY y lo pueda utilizar contra un *proxy*.
6. Debe brindar un archivo PAC¹.

¹ Archivo de configuración para que el browser dirccione sus peticiones hacia el proxy.

7. Servirá una Interfaz Web en donde se podrá acceder a la configuración del proxy, estadísticas, logs, estado de las conexiones, información de los objetos en caché, etc.

2.2. Funcionamiento básico

1. El proceso principal escucha la petición en un socket.
2. Cuando llega una conexión entrante, se negocia el protocolo que se va a utilizar. El protocolo va a ser SPDY sobre SSL.
3. Realizar el procesamiento de la petición.
 - a) Parsear el request del cliente y extraer el recurso que se está pidiendo.
 - b) Revisar si el ítem está presente en la caché del *proxy*.
 - c) En el caso de que no estar presente en la caché o haya expirado, armar la petición del recurso.
 - d) Seleccionar el método de petición que sea más óptimo y hacer el pedido al servidor.
 - e) Recibir el recurso, actualizar la caché, guardar información necesaria para las estadísticas y para alimentar el algoritmo de selección del método.
 - f) Procesar el recurso, revisar peticiones posteriores para priorizar los streams² o para aplicar técnicas de *push* y *hint*.

En el caso de necesitar recursos que no son del dominio original, modificar el HTML para que sea transparente al cliente de donde se están trayendo los recursos y así poder aprovechar de las features de SPDY en la conexión local.
4. Enviar el recurso al cliente.

²En el caso de utilizar SPDY.