AN EFFICIENT APPROACH TO SECURE WEB USING SPDY PROTOCOL

Sanket patel[1], Dr.Sharnil pandya[2]

[1]Student, Computer Science &Engg, PIET Vadodara, Gujarat, India
[2]Associate professor, Computer Science &Engg, PIET Vadodara, Gujarat, India

ABSTRACT

SPDY is a protocol using for improving the loading speed of webpage .this protocol also reduce the data loading time and transfer the data packets fastly. SPDY also allow making the multiple requests in signal TCP connection because SPDY uses the multiplexing technique. SPDY create the SSL certificate for the single session and allow transferring the data over network protocols. SPDY use the TCL for the compressing the data. In the SPDY protocol the request send to the server for the data pushing so the SPDY improve the webpage loading time.

Keyword: - *SPDY, Multiplex, SSL, HTTP*

1. INTRODUCTION

SPDY (Speedy) Protocol is an application layer protocol built by chromium projects .SPDY was derived from English word Speedy. SPDY is designed to reduce the latency of web pages. The Speedy was designed in order to speed up the transfer and access to web pages. SPDY attempts to protect the existing rule of HTTP in order to avoid changes or re-writing of existing web applications. [1] HTTP protocol design leads to the high latency of web pages. The HTTP uses a text-based protocol header but it is not good because it uses costly parsing. HTTP is connectionless and therefore, most request or response messages contain much unnecessary information for concurrency. HTTP uses multiple TCP connections. SPDY provides a session layer for HTTP to multiplex all request and response messages belonging to one session into one SPDY session. Then it transmits the data over one TCP sessions. This can be used by the web server to send data without an extra client request. This feature can reduce the latency. SPDY wasn't only designed to make HTTP speedy but there are various solutions to web latency like SCTP i.e. Stream Control Transmission Protocol.[11] The stream control Transmission Protocol is a transport layer protocol. It replaces the TCP and provides multiplexed streams and stream-aware traffic.

2. SPDY/HTTP2.0

SPDY was developed by Google as a way of adding effectiveness to the normal HTTP protocol. It uses compression and even predicting requests so resources are sent faster from the server to the browser. In 2009 Google announced a new protocol SPDY and they promised that it would considerably fasten up the web browsing. But SPDY does not replace HTTP. It creates a session between the HTTP application layer and the TCP transport layer. It speeds downloading using multiplexed streams, request prioritization and HTTP header compression. Last year Google insisted that SPDY could reduce byte count by 15% and packet count by 40%, and that SDPY's header compression could reduce latency by up to 85% .[21]
Twitter enabled SPDY on its servers in March 2012, at the time making it the second largest site known to deploy SPDY. [11] In May 2012 F5 Networks announced support for SPDY in its BIG-IP application delivery controllers. [6]In June 2012 NGINX, Inc. announced support for SPDY in the open source web server Nginx. [7]
Cloud flare is also providing a beta of SPDY on their servers (using Nginx) from June 2012, though users who would like to use/test it must be paying customers as SPDY is built on top of TLS, and only paying customers can

use SSL/TLS Certificates. [7] In July 2012 Facebook announced implementation plans for SDY. [8] By March 2013 SPDY was implemented by their public web server. [9] In August 2012 Wordpress.com announced support for SPDY across all their hosted blogs. [10]

3. PROBLEM STATEMENT

Single request per connection. Because HTTP can only fetch one resource at a time (HTTP pipelining helps, but still enforces only a FIFO queue), a server delay of 500 ms prevents reuse of the TCP channel for additional requests. Browsers work around this problem by using multiple connections. Since 2008, most browsers have finally moved from 2 connections per domain to 6.

Exclusively client-initiated requests. In HTTP, only the client can initiate a request. Even if the server knows the client needs a resource, it has no mechanism to inform the client and must instead wait to receive a request for the resource from the client.

Uncompressed request and response headers. Request headers today vary in size from ~200 bytes to over 2KB. As applications use more cookies and user agents expand features, typical header sizes of 700-800 bytes are common. For modems or ADSL connections, in which the uplink bandwidth is fairly low, this latency can be significant. Reducing the data in headers could directly improve the serialization latency to send requests.

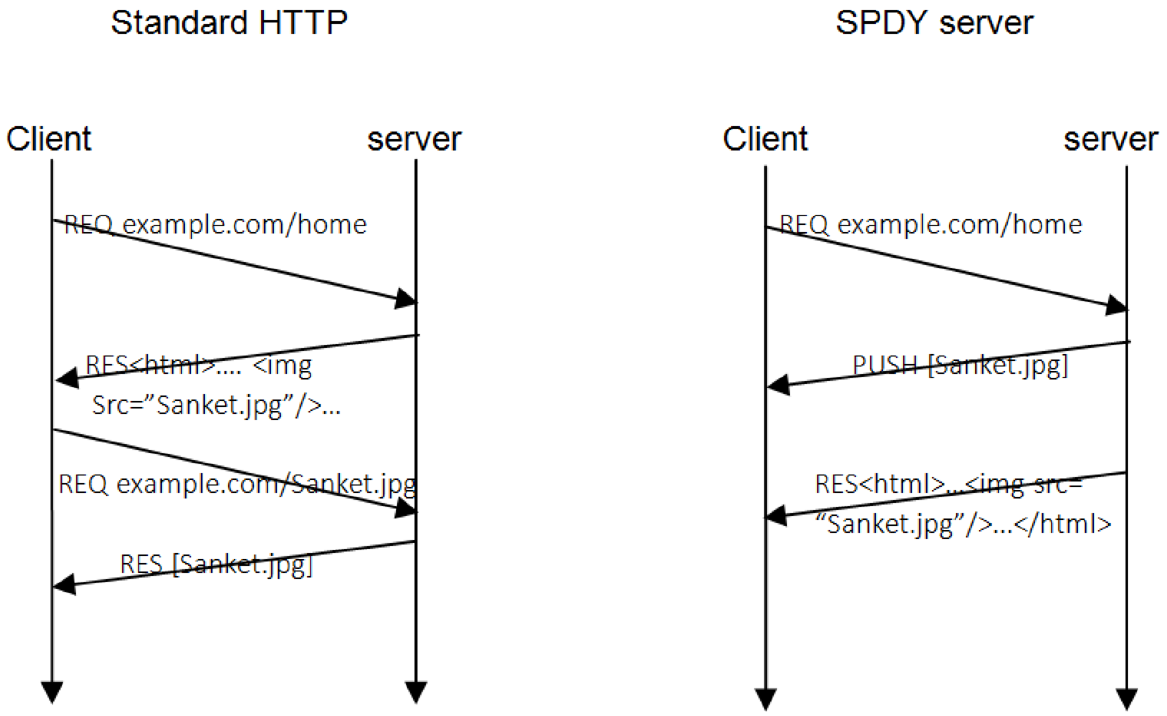

Fig 1.1: Request handling over HTTP and SPDY[18]

Redundant headers. In addition, several headers are repeatedly sent across requests on the same channel. However, headers such as the User-Agent, Host, and Accept* are generally static and do not need to be resent.

Optional data compression. HTTP uses optional compression encodings for data. Content should always be sent in a compressed format. The internet has continued to evolve over the last 20 years but the standard has not. The timeline below illustrates just a few of the major events in HTTP's lifetime. There is a huge gap in the evolution of the standard, but during this time frame many key events happened, such as the ability to access the internet from your phone and make a phone call from your computer.

4. PROPOSED WORK

The protocol is being proposed by keeping in mind that speeds up the speed of loading website. Also this protocol make the data comparison on the header data and that is use the SSL certificate for the verification of the user authentication for the security purpose. Now need to web faster with the security so for that using SPDY protocol and SSL certificate. The goal of the proposed system is decrease the load time and makes the many more HTTP requests to run across a signal TCP session.

5. IMPLEMENTATION ENVIROMENT

SPDY adds a session layer at top of SSL which allows for multiple parallel, interleaved streams through a single TCP connection. The GET and POST message formats remain the same. SPDY specifies a new framing format for encoding and transmitting the data over the wire. Streams are bi-directional, and can be initiated by the client and server. SPDY aims to achieve lower latency through basic (always enabled) and advanced (optionally enabled) features. SPDY is an Application layer protocol for transporting content over the web, designed specifically for minimal latency. There are three main points to SPDY that are (most) close to modern and emerging web architectures. [13]

1. Only single, asynchronous connection is allowed between client and server

2. SPDY is placed above TCP and encapsulates HTTP

3. Requests can be prioritized

4. SPDY also use TCL protocol

5. Network protocol used

The main principle of SPDY is the use of a single, asynchrony-nous connection (1) between client and server to reduce latency inherent in network transfer times. Clients then send a series of requests with or without priority (3) desired over that connection. Those requests are encapsulated into SPDY (2) and sent to a SPDY-capable web server infrastructure where they are translated and processed before being sent back. This process is, as Google points out, much faster than traditional acceleration techniques involving parallelization of requests because a browser simply cannot open a number of connections to the web server commensurate with the number of objects (requests) it must retrieve [3]. Connection limitations and the synchronous nature of the HTTP protocol impose a performance penalty that is not possible to eliminate. Google has very impressive performance results from using SPDY. [6][9]

6. REFERENCES

[1] *SPDY Multiplexing Approach on Long-latency Links* ,A. Abdel Salam,M.Luglio,C.Roseti,F.Zampognaro, department of Electronics engineering, university of rome"tor vergata"Rome ITALY.IEEE WCNC'14 Track 4.

[2] *Can SPDY Really Make the Web Faster?*, Yehia Elkhatib, Gareth Tyson and ,Michael Welzl Lancaster University, UK ,Queen Mary, University of London, UK ,University of Oslo, Norway, ISBN 978-3-901882-58-6 c 2014 IFIP

[3] *SPDY Over Satellite: Performance Optimization Through an End-to-End Technology*, Cesare Roseti, Ahmed Abdel Salam, Michele Luglio, and Francesco Zampognaro, 978-1-4799-8498-5/15/$31.00 ©2015 IEEE

[4] *Is HTTP/2 Really Faster Than HTTP/1.1?* Hugues de Saxc´e, Iuniana Oprescu, Yiping Chen Orange Labs Issy-les-Moulineaux, France,979-1-4679-7131-5/15©2015IEEE

[5] *How the Web Was Won Keeping the Computer Networking Curriculum Current with HTTP/2*, Colin Allison, Hussein Bakri, School of Computer Science University of St Andrews St Andrews, UK, 978-1-4799-8454-1/15/$31.00 ©2015 IEEE

[6] *Low-Rate Denial-of-Service Attacks against HTTP/2 Services*, Erwin Adi, Zubair Baig, Chiou Peng Lam, Philip Hingston School of Computer and Security Science, Security Research Institute Edith Cowan University Perth, Australia , 978-1-4673-6537-6/15/$31.00 ©2015 IEEE

[7] *Power Efficient Mobile Video Streaming Using HTTP/2 Server Push*, Sheng Wei Viswanathan Swaminathan Mengbai Xiao Adobe Research Adobe Systems Inc. 345 Park Ave, San Jose, CA 95110, USA, 978-1-4673-7478-1/15,2015 IEEE

[8] *Towards a SPDY'ier Mobile Web?*,Jeffrey Erman, Vijay Gopalakrishnan, Rittwik Jana, and Kadangode K Ramakrishnan ,1063-6692 © 2015 IEEE

[9] *The upcoming new standard HTTP/2 and its impact on multi domain websites,* Heejung Kim, Jongseok Lee, Ikhyun Park, Hyungkyung Kim, Dong-Hoon Yi and Taesung Hur Future Technology R&BD Department, KT Institute of Convergence Technology Seoul, Republic of Korea, Copyright 2015 IEICE

[10] *DYNAMIC ADAPTIVE STREAMING OVER HTTP/2.0*, Christopher Mueller, Stefan Lederer, Christian Timmerer, and Hermann Hellwagner Alpen-Adria-Universitat Klagenfurt UniversitatsstraBe 65-67 9020 Klagenfurt am W6rthersee, Austria, ALICANTE (FP7-ICT-248652)

[11] *"Announcing SPDY draft 2 implementation in nginx"*. Nginx, Inc. 2012-06-15. Retrieved 2012-06-16.

[12] *A comparison of SPDY and HTTP performance* Jitendra Padhye and Henrik Frystyk Nielsen.July 26, 2012.

[13] Finley, Klint. *"Facebook Makes Itself a Bit More SPDY"*. Wired. Retrieved 18 March 2013.

[14] "Just enabled *#SPDY for all http://WordPress.com -hosted sites"*. 2012-08-28. Retrieved 2012-08-28.

[15] *SPDY Accelerator for Improving Web Access Speed Gen*, Mineki, Satoshi Uemura, and Teruyuki Hasegawa KDDI R&D Laboratories Inc.

[16] Hypertext Transfer Protocol HTTP/1.1 RFC 2616 HTTP/1.1 June, 1999

[17] https://support.zenmate.com/hc/en-us/articles/204423722-Chrome-error-ERR-SPDY-PROTOCOL-ERROR-

[18] http://serverfault.com/questions/615321/error-code-err-spdy-protocol-error

[19] https://www.chromium.org/spdy/spdy-whitepaper

[20] http://seminarprojects.com/Thread-spdy-protocol

[21] http://www.slideshare.net/schellkenig/dynamic-adaptive-streaming-over-http20?qid=23d4d7a3-5d10-4f78-897d-bce0debf1067&v=qf1&b=&from_search=4

[22] https://www.chromium.org/spdy/spdy-tools-and-debugging

[23] https://developers.google.com/speed/spdy/