Performance Comparison of Congested HTTP/2 Links

Brian Card, CS 577
12/7/2014
Outline

• HTTP/1.1 and HTTP/2 Connections
• Problem Statement
• Experimental Setup
• Results
HTTP/1.1 and HTTP/2
Connections
Background of HTTP

• HTTP is the Hypertext Transfer Protocol, the underlying protocol of the Web
• Major version of HTTP/1.1 was finalized in RFC 2616 in 1999
• Why do we need a new version of HTTP?
Speed!

- Average website complexity is increasing!

- Users expect more features and better performance at the same time!

- Increasing capacity does not solve problem, latency has a significant affect on page load time

HTTP/1.1 – Connection Usage

- HTTP/1.1 connections can request a single object at a time
- But there are many objects in a webpage and some have dependencies
- What happens if I want to parallelize downloads?
  - Need to open multiple TCP connections (usually max 6)
- What happens if a less important object is downloaded first?
  - This is a problem
  - Head Of Line Blocking where a large unimportant file ‘clogs up’ the connection
Domain Sharding

- Maximum of 6 connections per domain
Domain Sharding

• Create More Domains!
Domain Sharding

- Average Number of Domains per website is 18!

http://httparchive.org/trends.php
Domain Sharding

• Average Number of (TCP) Connections per Page

http://httparchive.org/interesting.php
Other HTTP/1.1 tricks

• Inline files, (base64 encode an image, put it in CSS file)
• Concatenate files (push all CSS files into a single file)
• All trying to get the same amount of content to stream over a single connection or multiple simultaneous connections
HTTP/2 Multiplexed Streaming

• Can stream multiple resources over the same connections
• Resources are prioritized by specifying a dependency graph, no Head Of Line Blocking
HTTP/2 Multiplexed Streaming

- Single TCP connection per domain
- Up to 100 ‘Streams’ per TCP connection
Other features of HTTP/2

- Header Compression
- Binary Protocol (HTTP/1.1 is text)
- Encryption Required
- Server Push (Server can push resources uninitiated)
Problem Statement
But What About Congestion Control?
Congestion Control

• HTTP/1.1 uses many connections
• When a packet is dropped, only one connection goes into congestion avoidance
  – Throughput of a single connection is halved
• HTTP/2 has one connection, when a packet is dropped the throughput of the entire download is halved
• HTTP/2 connections may reach higher speeds faster, since new connections do not need to be created
• If websites do not load faster, people won’t use HTTP/2
This Project

- Quantify the effects of congestion control on HTTP/1.1 vs HTTP/2
- Will HTTP/2 make domain sharding disappear?
Experimental Setup
Overview

Web site hosted by multiple servers on a Virtual Machine. HTTP/1.1 and HTTP/2 servers are swapped out depending on experiment. Measure time to load page completely.
Setup Details

- MacBook Air 10.8.5 Mountain Lion
- http-server 0.7.4 (node.js)
- nghttp 0.6.4 (supports draft-ietf-httpbis-http2-14)
- VirtualBox 4.3.6
- Ubuntu 14.04 LTS
- Firefox Nightly 36.0a1
- Network Link Conditioner (Late July 2012)
- IP Aliasing to create multiple addresses
Setup Details (continued...)

- http-server used as the HTTP/1.1 Server
- nghttp used as the HTTP/2 server
  - Had to edit codebase to support multiple servers binding to different IP addresses on the same machine
- Sample website is hosted by either server
- Measure the time it takes to load under varying network conditions
- TLS enabled on both web servers
Firefox Settings

- Firefox network analyzer used to measure page load time
- Nightly build has latest HTTP/2 support
- Caching disabled:

<table>
<thead>
<tr>
<th><code>about:config Setting</code></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>network.http.spdy.enabled.http2</td>
<td>true</td>
</tr>
<tr>
<td>network.http.spdy.enabled.http2draft</td>
<td>true</td>
</tr>
<tr>
<td>network.http.use-cache</td>
<td>false</td>
</tr>
<tr>
<td>browser.cache.offline.enabled</td>
<td>false</td>
</tr>
</tbody>
</table>
## Sample Website

<table>
<thead>
<tr>
<th></th>
<th>HTML</th>
<th>CSS</th>
<th>JavaScript</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1*60KB</td>
<td>1*60KB</td>
<td>5*1KB</td>
<td>10*1KB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5*10KB</td>
<td>10*5KB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5*20KB</td>
<td>20*20KB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5*37KB</td>
<td>5*40KB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5*140KB</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>60KB</td>
<td>60KB</td>
<td>340KB</td>
<td>1360KB</td>
</tr>
</tbody>
</table>
Sample Website (continued...)

- Each file is mostly filled with comments to take up available space
- Content is rendered ‘below the fold’
- For sharded domains objects are distributed equally between domains
- This ensures equal load between domains
  - In the real world, resources would typically be segregated by domains (images all on one set of domains)
Results
Results

• ‘Warm-up’ run used before collection results
  – eliminates some startup costs on servers
  – do not want to test implementation differences between http servers

• Results are average of 3 runs
Varying Capacity in Ideal Conditions

Page Download Time for HTTP/1.1
No Delay, No Loss

Page Download Times for HTTP/2
No Delay, No Loss
Comparison of HTTP/2 vs HTTP/1.1

Difference In Page Download Time, Higher Values show HTTP/2 Advantage

![Graph showing comparison of HTTP/2 vs HTTP/1.1]

- Throughput Mbps
- Time Difference (ms)
- One Domain
- Two Domains
- Three Domains
- Five Domains
Comparison of HTTP/2 vs HTTP/1.1

Page Download Time HTTP/2 vs HTTP/1.1
No Delay, No Loss

Throughput Mbps

Percent Increase vs HTTP/1.1

0.5
5
50

0%
10%
20%
30%

-10%
-20%
-30%

One Domain
Two Domains
Three Domains
Five Domains
Varying Capacity, 79ms of Delay

Page Download Times HTTP/2 vs HTTP/1.1
79ms Delay, No Loss
One Domain HTTP/2 vs 5 Domains HTTP/1.1

HTTP/2 With One Domain vs HTTP/1.1 With Five Domains
79ms Delay, No Loss
Increase In Performance for Increasing Domains in HTTP/2?

HTTP/2 Speed Increases For Increasing Domains
79ms, No Loss

Percent Improvement Over Single Domain

Throughput Mbps

One Domain
Two Domains
Three Domains
Five Domains

Worcester Polytechnic Institute
Loss for HTTP/1.1 and HTTP/2

- Fixed Throughput at 5Mbps
- Fixed Latency at 79ms
- Varied Delay 1%, 2%, 5%, 10%
- Last value for single domain HTTP/2 is ‘optimistic’; ran more than three runs and timeouts were occurring
Loss For HTTP/1.1 and HTTP/2

Page Load Times With Varying Loss Rate
79ms Delay, 5Mbps Throughput

Page Load Time (ms)

Loss

0% 2% 4% 6% 8% 10% 12%

HTTP/1.1 Five Domains
HTTP/2 Five Domains
HTTP/2 One Domain
Conclusions

• HTTP/2 has increased performance over HTTP/1 for non-congested links
• Unclear if domain sharding will disappear with HTTP/2
• Under high loss links, HTTP/2 performs significantly worse
• HTTP/2 may not be ubiquitous, advantages for high bandwidth delay product links with no loss