SPDY vs HTTP/1.1: An Empirical Evaluation of Network Protocol Performance

Stephen Matthew Chambers
University of New Hampshire - Main Campus

Follow this and additional works at: https://scholars.unh.edu/student_research

Part of the OS and Networks Commons

Recommended Citation
https://scholars.unh.edu/student_research/5

This Undergraduate Research Project is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Student Research Projects by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.
HTTP vs SPDY

Why change?

- As the Internet evolves, the reduction of page load time has an increased importance.
- The application layer should be changed to avoid altering existing implementations.
- SPDY is a Google proprietary protocol that is deployed in the production environment already on websites such as Google, Facebook, and Twitter.
- SPDY is the working base for HTTP/2.0.

### Background

- Optional compression encodings for data.
- Sends static header data throughout connection.
- Only the client can initiate a request.
- HTTP uses multiple connections because it can only process requests in a FIFO queue.
- Multiplexing over a single connection.
- Server push/Server hint: Server can either suggest a resource to request or push the request to the client unsolicited.
- Removes static information, such as the User-Agent and Host headers.
- Forces header compression.

### Experimental Setup

<table>
<thead>
<tr>
<th></th>
<th>HTTP</th>
<th>SPDY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Load Page</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>Bytes Transferred</td>
<td><img src="image3" alt="Graph" /></td>
<td><img src="image4" alt="Graph" /></td>
</tr>
<tr>
<td>Throughput</td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>Percentage Increase</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
</tr>
</tbody>
</table>

- Client requested a webpage with 100 small image files totaling ~3.4MB.
- PHP script used to generate distinct pages in order to avoid content caching.
- Presented results are averages of 5 runs.

### Throughput and Page Load Time

**Analysis**

- In a high latency network with zero packet loss, SPDY outperforms HTTP in terms of throughput as it takes advantage of SPDY's multiplexing.
- In a high packet loss network with near zero latency, SPDY outperforms HTTP. Very small latency masks packet loss problems, so SPDY can recover very quickly.
- In a bad network with high packet loss and high latency, HTTP outperforms SPDY. HTTP can perform load balancing with its multiple connections.

### Next Steps

Experiments are far from exhaustive. Different application types should be tested against; video files and dynamic content would take advantage of SPDY’s Server Push and Server Hint features.