

A decorative header featuring a horizontal line. Above the line, there are four overlapping spheres: a green one on the far left, a blue one in the middle, a red one slightly behind and to the right of the blue one, and a yellow one on the far right.

# SPDY - A Web Protocol

Mike Belshe  
Velocity, Dec 2009



# What is SPDY?

## Concept

SPDY is an application layer protocol for transporting content over the web with reduced latency.

## Basic Features

### 1. Multiplexed Stream Support

SPDY can send many sessions concurrently over a single TCP connection without serializing requests.

### 2. Request Prioritization

SPDY implements request priorities. A client can request as many items as it wants from the server, and request that the server use best-effort to return the content in the highest-priority first. This allows the client to be free to request resources without having to worry that those requests will clog the channel.

### 3. HTTP Header Compression

SPDY compresses HTTP headers, leading to fewer packets and fewer bytes transmitted.

## Advanced Features

### 1. Server Push Streams

SPDY allows either the client or server to initiate a stream once the client has established a connection.

### 2. Server Hint Streams

The server often knows a client will need a resource. These headers assist with informing the client about resource it would otherwise discover much later.



# SPDY Goals

---

- Avoid requiring the website author to change content
  - Allow incremental changes
  - Performing "better" with content changes is okay
  - Performing "worse" without content changes is unacceptable
- Be a drop-in alternative to HTTP, and perform always better, never worse than HTTP

# What we built

- Hi-speed, in-memory server which can serve both HTTP and SPDY responses efficiently.
- Modified Chrome client which can use HTTP or SPDY
- Benchmarking infrastructure
  - A mechanism for high-fidelity page replicas
    - Preserves origin server headers, content encodings, everything
    - Preserves multiple-subdomain separations
    - Preserves URLs without rewriting

# Network Efficiency

Test:

Download the same "top 25" pages via HTTP and SPDY

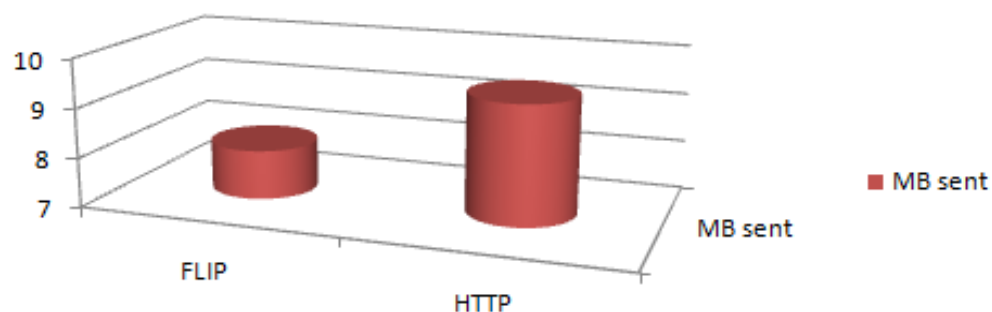
Network simulates a 2Mbps DSL link, 0% packet loss.

	FLIP	HTTP	% delta
# Pkts	8487	14142	-40%
Avg Pkt Size	943B	667B	41%
MB Transferred	8.0MB	9.4MB	-15%

**Total TCP Packets Sent**  
FLIP Sends 40% Fewer Packets



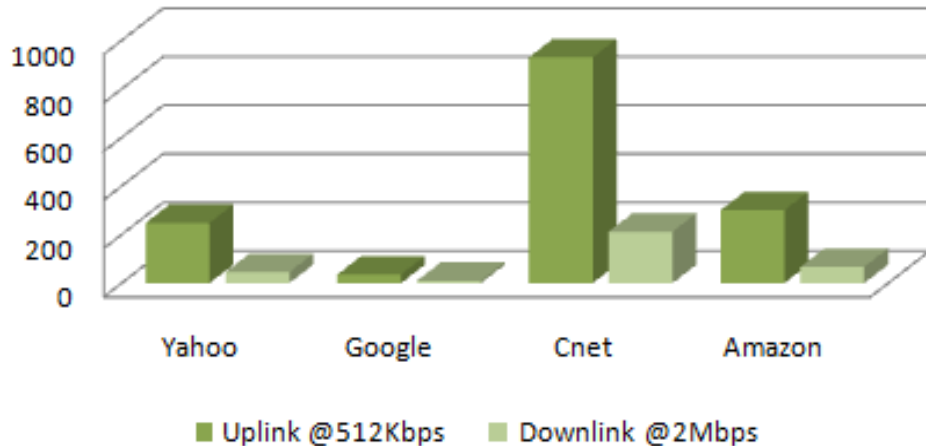
**TCP Bytes In & Out**  
FLIP transmits 15% fewer bytes



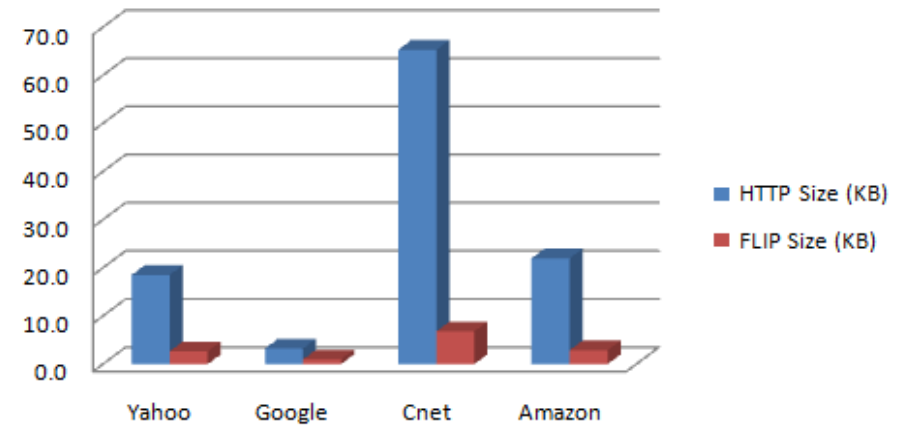
# Header Compression

On low-bandwidth links, headers are surprisingly costly. Headers alone can cost more than 1s of latency.

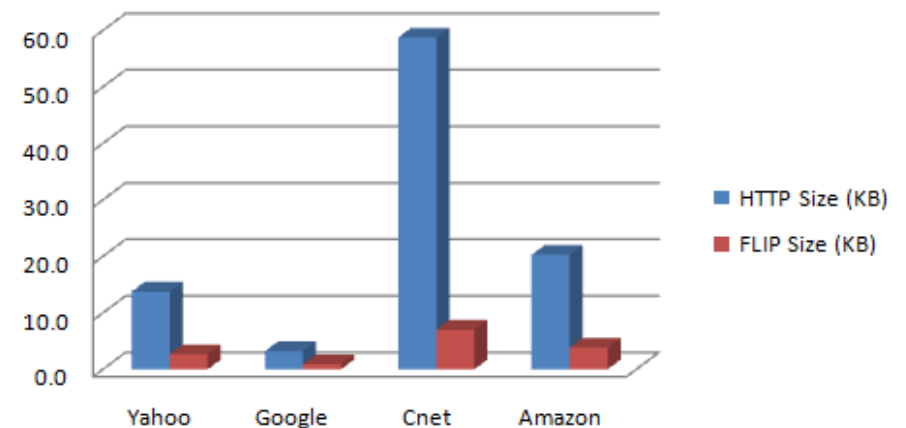
### Time Saved 45-1142ms per Page Load



### Request Header Compression ~88%



### Response Header Compression ~85%



# SPDY Stream Multiplexing

---

SPDY employs a simple low overhead framing mechanism  
All frames reference a stream-id, and include a length  
Data frames have an 8 byte overhead

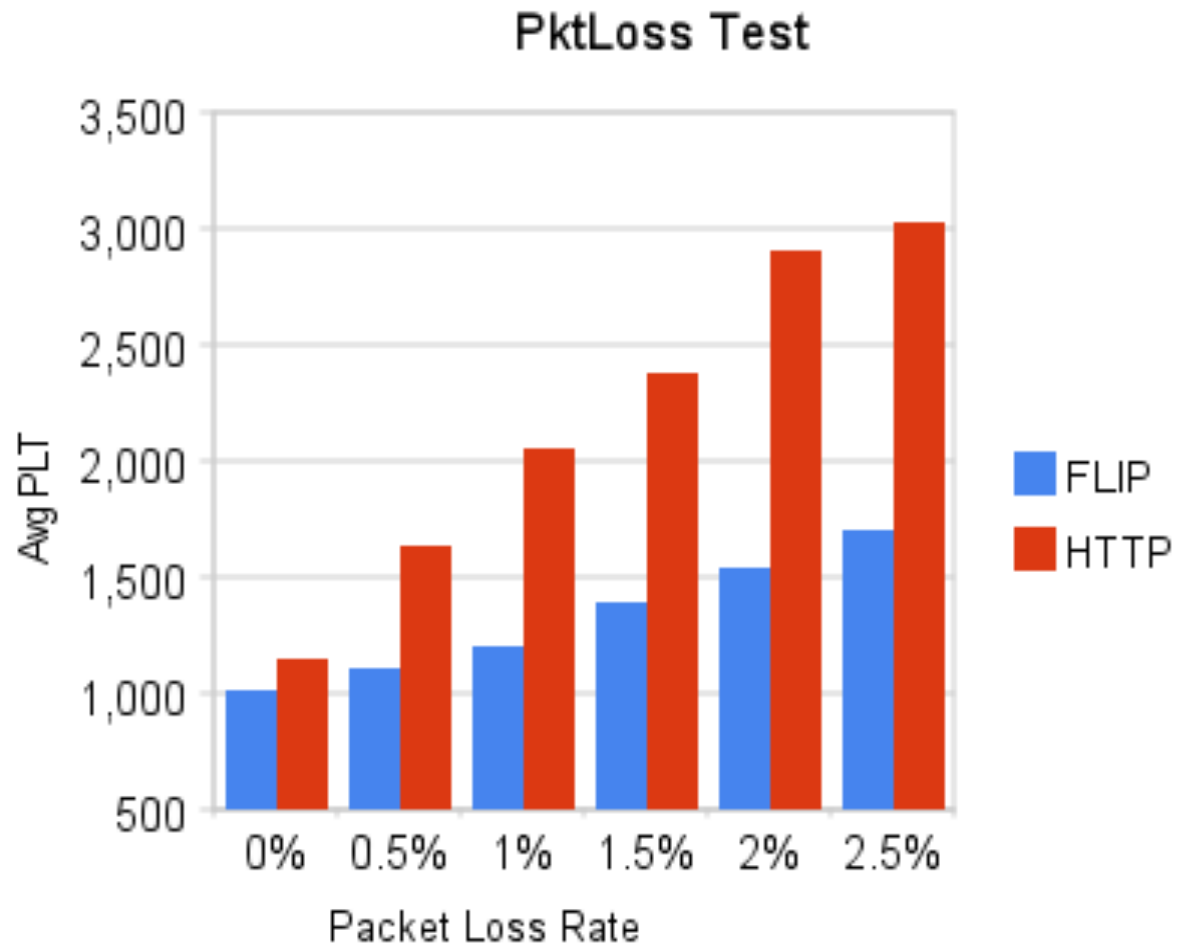
Efficiency of multiplexing relies on sender using frames which are not too large. We've used 4k frames, for an overhead of 0.2%



# Results: Packet Loss

We believe real world packet loss is ~1%.

SPDY is 41%-47% faster for PL between 1 and 2%.

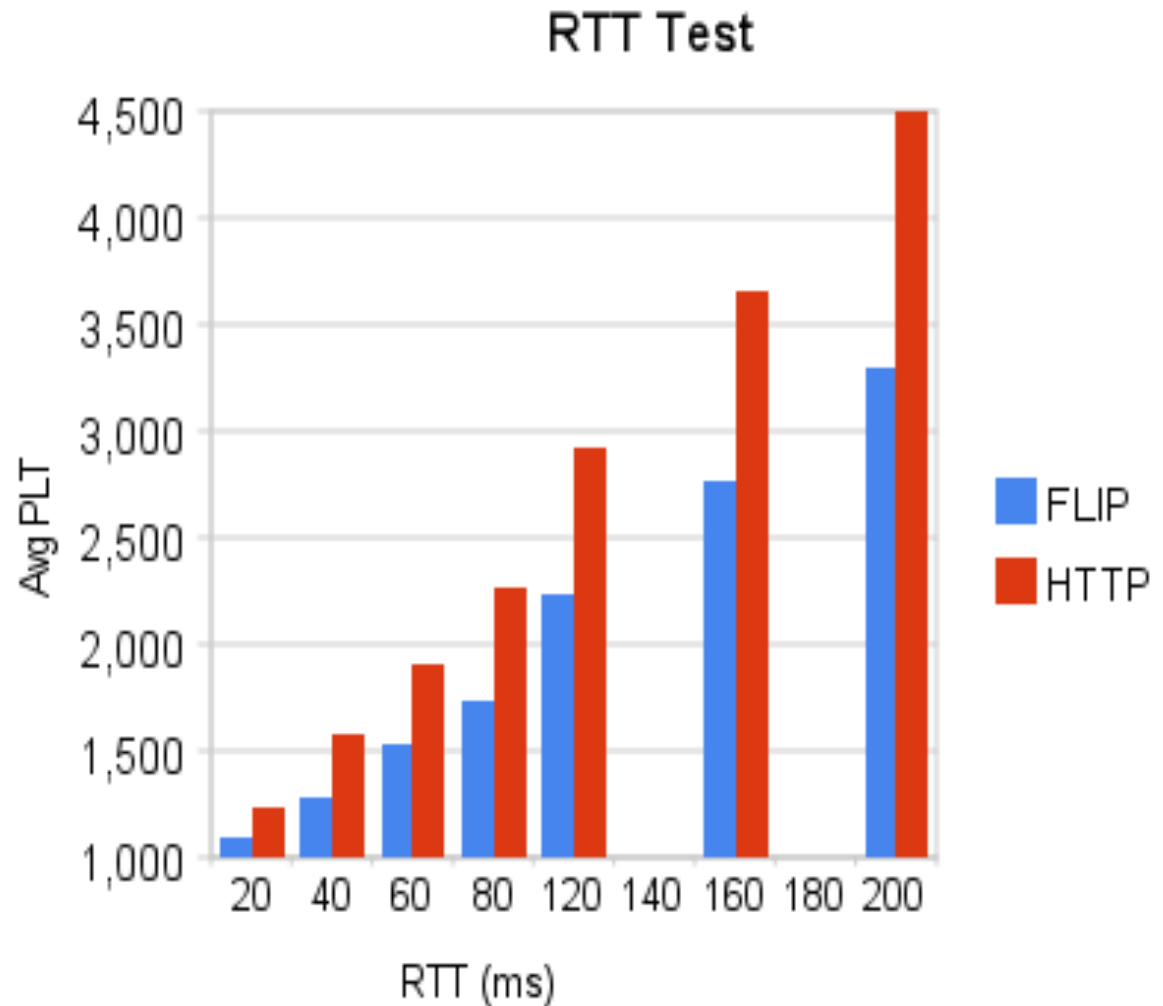


# Results: RTT

Average RTT is  
~110ms.

Fast RTTs are ~40ms.

Typical US is 50-  
100ms.

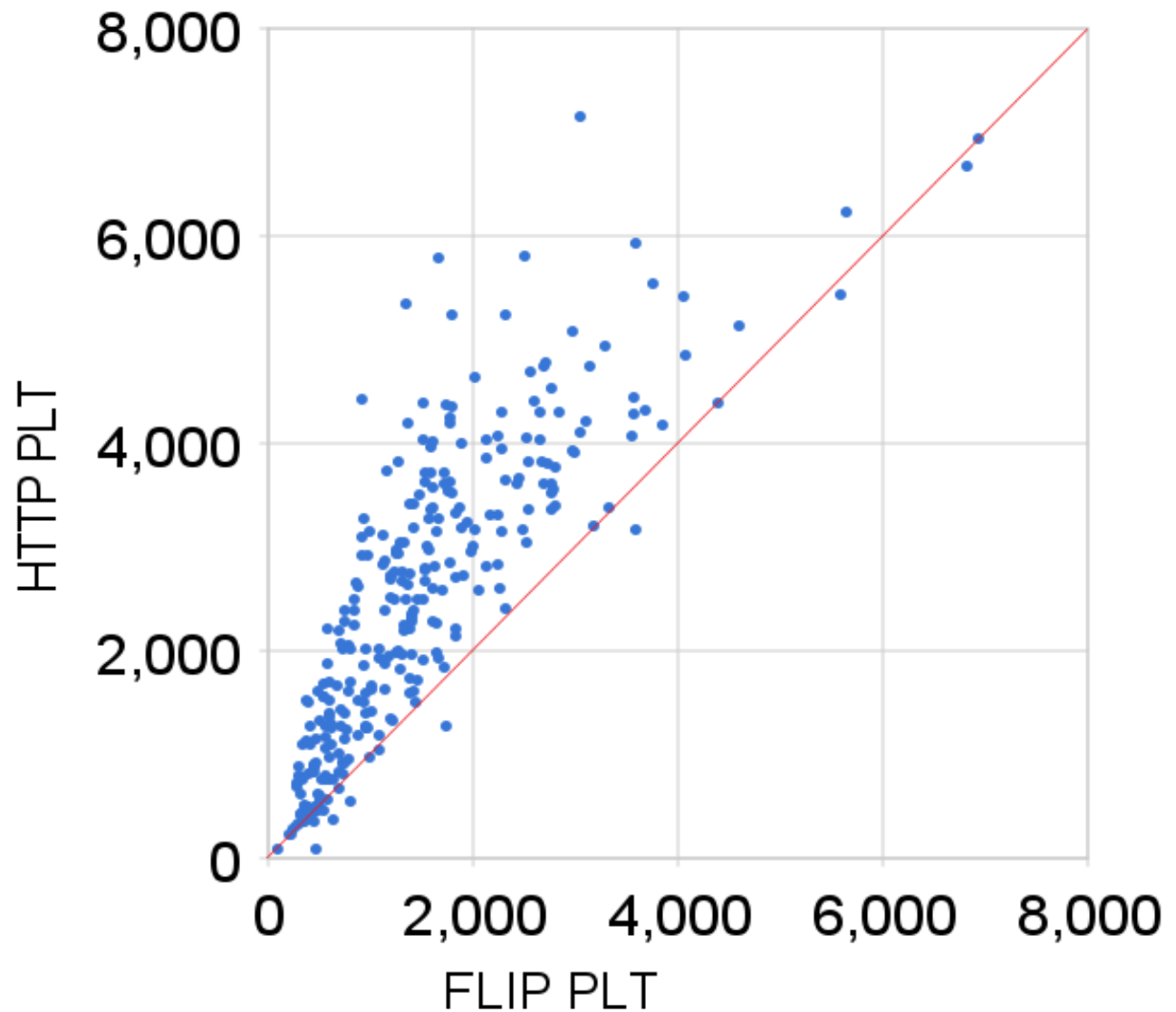


# Top 300 Content

Question: Do highly optimized websites get less benefit?

Randomly picked 300 sites from the Alexa Top-1000.

Overall page load improvement average was 40%.



# Other protocols

- HTTP over SCTP
  - Provides multiplexed streams.
  - Stream-aware congestion control.
  - SCTP replaces the transport.
  - Difficult to deploy a new transport.
- SST
  - New protocol
  - Lightweight streams over a single connection
  - Runs as a transport or over UDP

# Security

- A protocol for the next 20 years should be secure.
  - Encrypted data.  
Why is it still possible to eavesdrop?
  - Server-authentication.  
When connecting to your bank, it should be verifiably your bank, and not a 3rd party in the middle.
- Doesn't security work against reduced latency?  
Yes - round trips are costly. But, can we optimize this too?
- Optimizations:
  - Cut-through
  - Client-side tickets
  - Server Name Indication (SNI)
- Is Scalability an issue?  
Maybe. We don't think so. (Think Moore's law).  
We're measuring to find out.



# Deployment

- HTTP is Treacherous for new Protocol Deployments
  - Transparent proxies, anti-virus, and other software make incorrect assumptions about the stream
  - Examples: pipelining, web sockets, SDCH
  - WebSockets data indicates 10-15% of users are unable to leverage HTTP Upgrade headers.
- But, SSL provides a "clean" tunnel.
- SPDY can be negotiated as part of the SSL handshake.