SPDY Status

mbelshe IETF80 http-bis
What is SPDY?

- **Goal: Reduce Web Page Load time.**
- **Multiplexing**
  - Get the data off the client
- **Compression**
  - HTTP headers are excessive
  - Uplink bandwidth is limited
- **Prioritization**
  - Today the browser holds back
  - Priorities enable multiplexing
- **Encrypted & Authenticated**
  - Eavesdropping at the Cafe must be stopped
- **Server Push**
  - Websites do some of this today with data URLs
Deployment Status @ Google

- On by default since Chrome 6
  - Currently at 90%; 10% holdback is for A/B testing.
- On for all Google SSL traffic
- SPDY HTTP->SPDY proxy used externally some
- SPDY Proxy

In other words, yes, you can really use it now.

But SPDY is:
- experimental
- research
- not standardized (yet)
- going to change (and you can help guide it!)
Results
Less is More - Conns, Bytes, Packets

Connections Per Web Page

HTTP

SPDY

# of connections

SPDY vs HTTP Total Packets (Top-45 pages)

HTTP

SPDY

# packets

SPDY vs HTTP Upload KB Sent (Top-45 pages)

HTTP

SPDY

KB

51% reduction

SPDY vs HTTP Download KB (Top-45 pages)

HTTP

SPDY

KB

19% reduction

4% reduction
Latency

- Overall very good!
  - $\text{avg} > 7.5\%$
- No SPDY specific optimizations
- This is Google's optimized content, SSL only.
- Unknown diff between mac and win so far. User bias?
Not Too Shabby WebSocket

- docs.google.com has a "hanging get" for every doc open
- how to scale beyond 6 connections per domain?
  - docs[1-N].google.com
- but, gets expensive and is horribly inefficient
- switched to spdy and much happier
- Header compression mitigates the inefficiency of a hanging get
Next steps
1. Make SSL Faster
TLS - False Start

SSL vs SSL FalseStart performance

PLT (ms)

- SSL
- SSL/False...
Snap Start

- 0-RTT SSL
- We built it.
  - Don't like it.
- Doesn't do perfect forward secrecy
- Changes too complex when retrofitted atop existing SSL
2. Make the Transport Faster
Transport inefficiencies

- Single connection throttle
- Data in the syn packet
- Protocol layering of security atop transport doesn't work well
- Inability for kernels to change will push transports into user space, even if it is just TCP atop UDP.
Standardization
Be a squeaky wheel...

- Everything is open now
- We welcome help
- Others are already implementing and testing
- More people asking for SPDY standardization will motivate me.