HTTP/2 basics
Status
Deploy
Future
Daniel Stenberg

Email: daniel@haxx.se
Twitter: @bagder
Web: daniel.haxx.se
Blog: daniel.haxx.se/blog
about the protocol
Why HTTP/2

HTTP/1 ping-pong
... makes it latency sensitive
TCP connection fatigue
HTTP Pipelining failed

Deliver more data earlier!
The world is still big
+ slower through fiber
+ never the shortest distance
+ buffer (bloat)
+ radio networks =
Several hundred milliseconds
HTTP/2
Multiplexed streams
Compressed headers
Server push
Maintains HTTP paradigms
Multiplexed and compressed

- no gaps
- more requests
- earlier responses
Many TCP vs one

HTTP/1.1

HTTP/2
HTTP paradigms stay

Header and body

HTTP:// and HTTPS://

Most client-side apps won't notice
Most server-side apps won't notice
browsers
Browsers are HTTPS-only

Firefox: 17% HTTP/2

30% of HTTPS

HTTP/2 in 70% of browsers
51% of HTTPS contents over HTTP/2

29% of SSL sites within top-1000 use SPDY or HTTP/2

Googlebot groks HTTP/2 early 2016
deploy
Poking at it

Apache, NGINX, H2O, ATS, Caddy, Litespeed

nghttp2

curl

wireshark

h2i

SPDY Indicator
Challenges for you

h2 is straight-forward, but ...

HTTPS!

OpenSSL / other TLS-lib versions and ALPN

Mixed content / ads

Certs and Let's Encrypt

Let's Encrypt
HTTP/2 – what to expect for your site

It depends

20% - 60% faster is common

Server push makes a difference

Priorities is key

Remember: HTTPS

Shorten dependency chains!
the HTTP future
Improving what we have

h2 server push improvements
h2 extensions have not taken off
h2 client certs?
(slightly) improved cookies
Guide to TCP when writing HTTP
More HTTPS, blind caches?
Better h2 tools, more h2 comparisons
Beyond HTTP/2

Time to drop HTTP/1 legacies

HTTP/3 will happen faster

QUIC and the OSI model crash
Thank you!

Doing good is part of our code