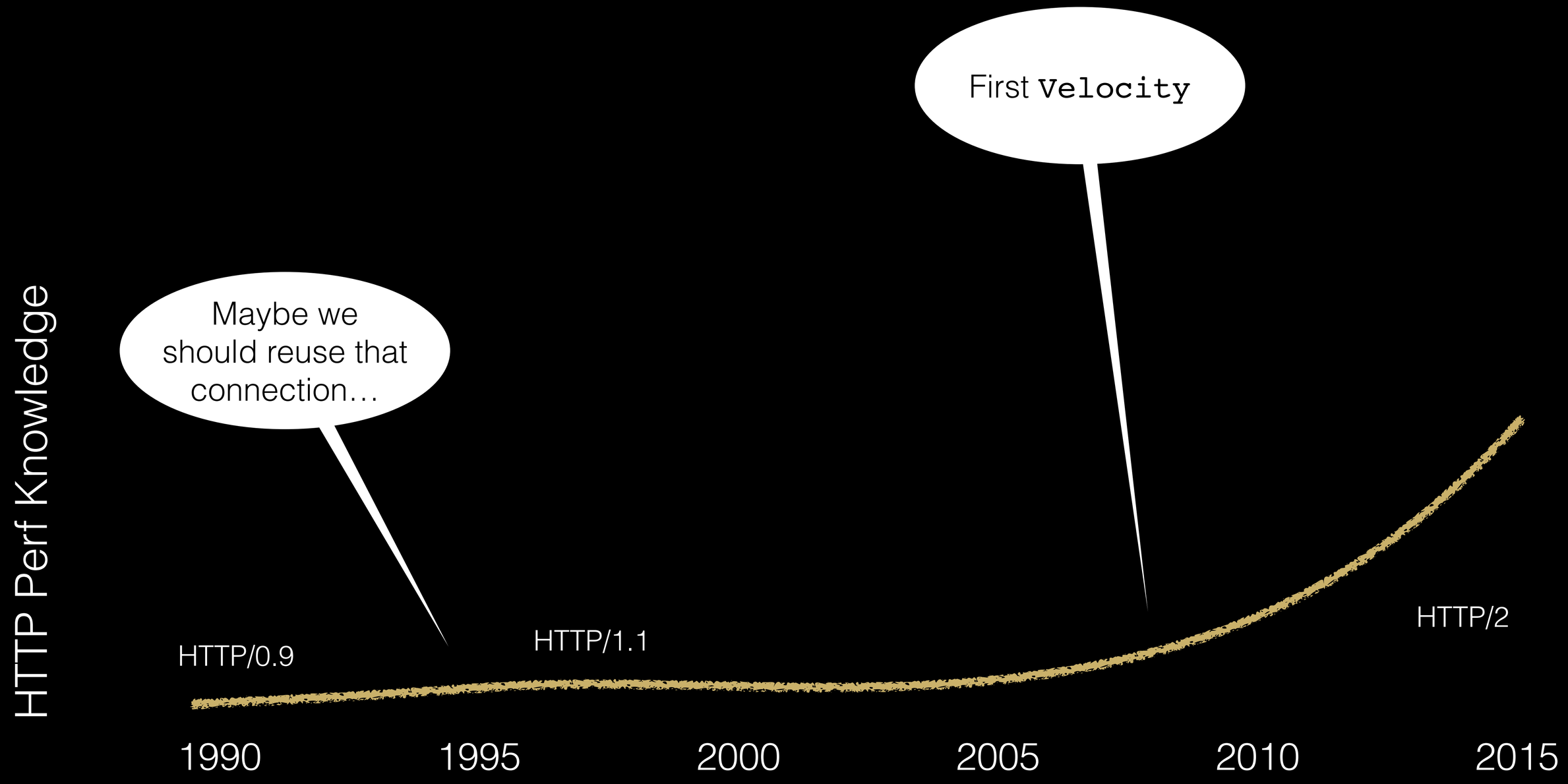
HTTP/2: Operable and Performant

Mark Nottingham @mnot (@akamai)

This talk may be disappointing.

Donald Rumsfeld

"As we know, there are **known knowns**; there are things we know we know. We also know there are **known unknowns**; that is to say we know there are some things we do not know. But there are also **unknown unknowns** — the ones we don't know we don't know."


HTTP Perf Knowledge
Maybe we should reuse that connection…
First Velocity
HTTP/0.9
HTTP/1.1
HTTP/2
1990
1995
2000
2005
2010
2015

HTTP/2 is almost here!

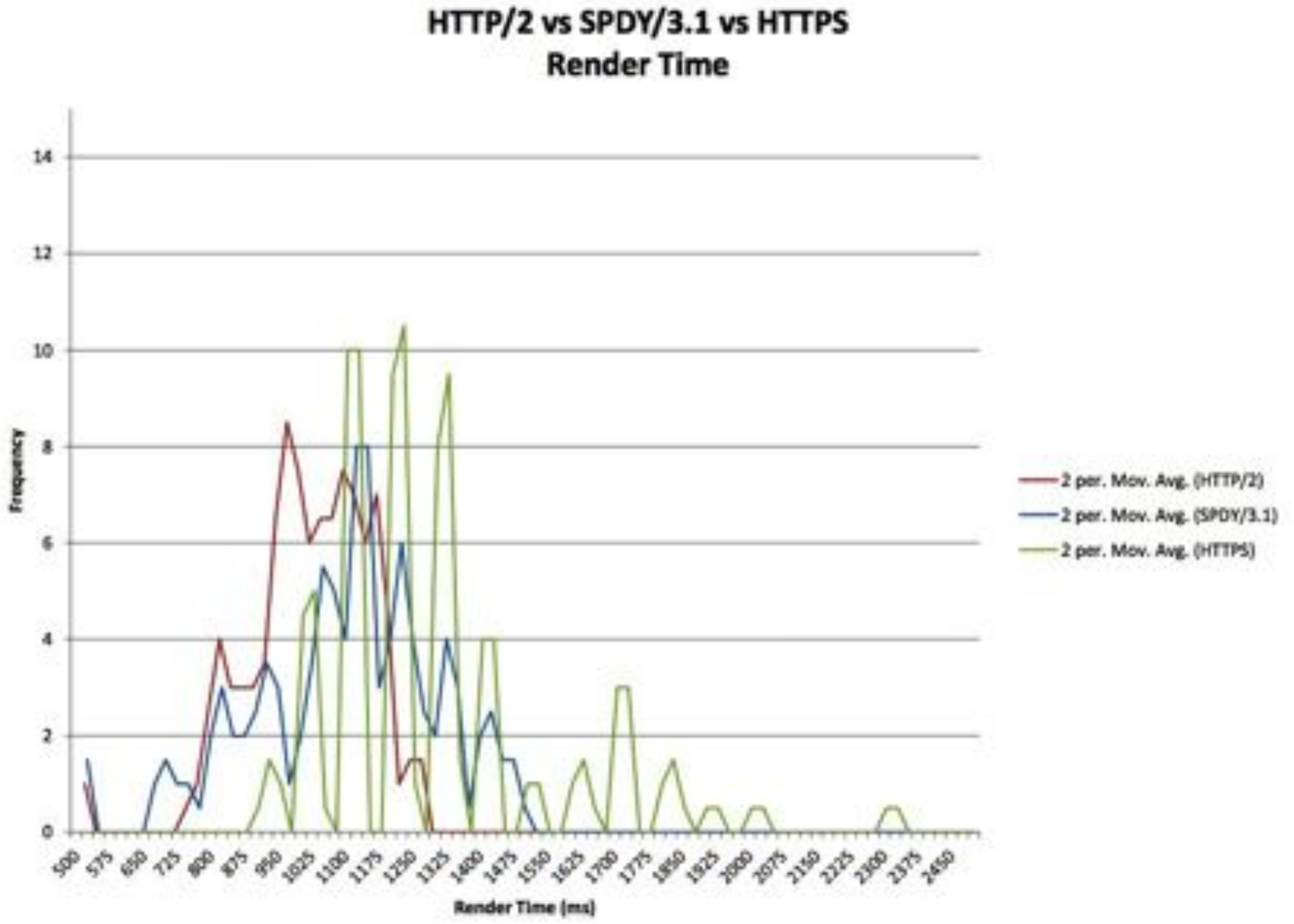
… and it's a lot like SPDY

HTTP/2 in One Slide

- Multiplexing + Header Compression + Server Push
- Goal: one connection from browser
- Post-WGLC
- Coming in Firefox, Chrome, IE, others very, **very** soon

Florian Bender 2014-09-15 14:46:18 PDT

```
Release Note Request (optional, but appreciated)
[Why is this notable]: The next era in Web tech has begun. Seriously.
[Suggested wording]: Implemented HTTP/2 and APLN.
```

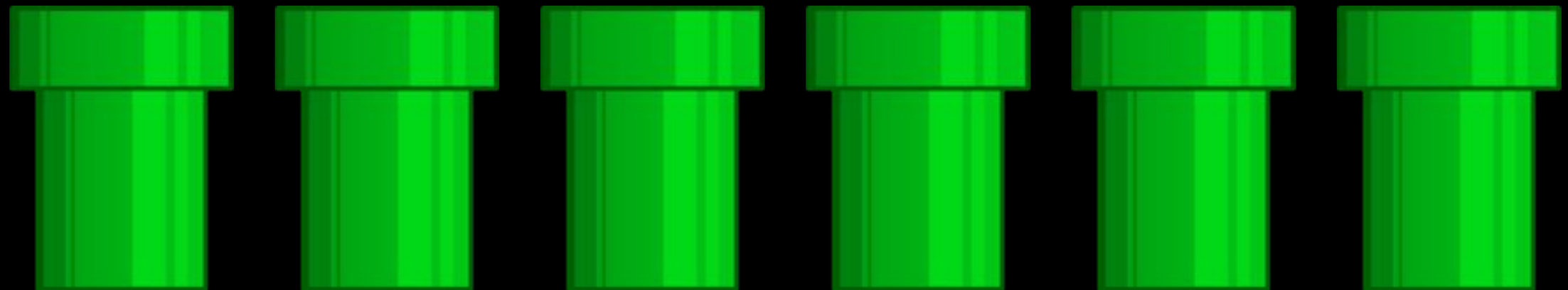

HTTP/2 vs SPDY/3.1 vs HTTPS
Render Time
Frequency
0
2
4
6
8
10
12
14
500
575
650
725
800
875
950
1025
1100
1175
1250
1325
1400
1475
1550
1625
1700
1775
1850
1925
2000
2075
2150
2225
2300
2375
2450
Render Time (ms)
2 per. Mov. Avg. (HTTP/2)
2 per. Mov. Avg. (SPDY/3.1)
2 per. Mov. Avg. (HTTPS)

{HTTP2, SPDY} and Performance

- Prioritisation
- Server Push
- Header Compression
- TLS and TCP

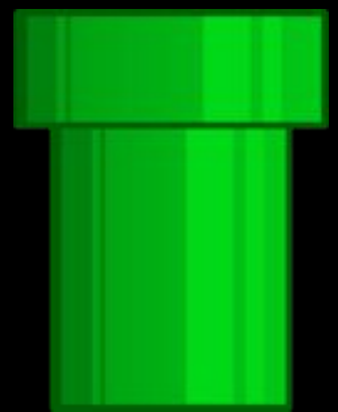
1. Prioritisation

HTTP/1



http://en.m.wikipedia.org/wiki/Harvard_University

GET Harvard_University	200 OK	en.m.wikipedia.org	329.7 KB	1.13s
GET common.css?version=120120111215S4	200 OK	bits.wikimedia.org	3.6 KB	367ms
GET android.css?version=120120111207T0	200 OK	bits.wikimedia.org	4.4 KB	390ms
GET jquery-1.7.1.min.js	200 OK	bits.wikimedia.org	91.7 KB	671ms
GET application.js?version=0108201112091S	200 OK	bits.wikimedia.org	3.3 KB	195ms
GET opensearch.js?version=1214201112943T	200 OK	bits.wikimedia.org	4.5 KB	203ms
GET W_logo_for_Mobile_Frontend.gif	200 OK	upload.wikimedia.org	379 B	342ms
GET close-button.png	200 OK	bits.wikimedia.org	1.9 KB	154ms
GET s-ehdpi.png	404 Not Found	bits.wikimedia.org	9 B	221ms
GET 20px-Padlock-silver.svg.png	200 OK	upload.wikimedia.org	856 B	245ms
GET 220px-Harvard_Wreath_Logo_1.svg.png	200 OK	upload.wikimedia.org	46.5 KB	938ms
GET Harvard_University_logo.PNG	200 OK	upload.wikimedia.org	2.7 KB	548ms
GET 220px-A_Westerly_View_of_the_Colledges_	200 OK	upload.wikimedia.org	8.2 KB	557ms
GET 296px-Rummell,_Richard_Harvard_Univers	404 Not Found	upload.wikimedia.org	9 B	492ms
GET 220px-Harvard_square_harvard_yard.JPG	200 OK	upload.wikimedia.org	18.5 KB	807ms
GET 170px-Harvard_University_map_(order,_dat	404 Not Found	upload.wikimedia.org	9 B	561ms
GET 220px-John_Harvard_statue_at_Harvard_Un	200 OK	upload.wikimedia.org	13.5 KB	906ms
GET 220px-Massachusetts_Hall,_Harvard_Unive	404 Not Found	upload.wikimedia.org	9 B	739ms
GET 220px-Harvard_yard_winter_2009j.JPG	200 OK	upload.wikimedia.org	14.2 KB	1.14s
GET 170px-Harvard_architects.png	200 OK	upload.wikimedia.org	88.4 KB	1.8s
GET 170px-Harvard_memorial_church_winter_2(	200 OK	upload.wikimedia.org	18 KB	1.29s
GET 220px-Widener_library_2009.JPG	200 OK	upload.wikimedia.org	10.6 KB	1.13s
GET 170px-University_Museum_at_Harvard_001.	200 OK	upload.wikimedia.org	7.8 KB	1.19s
GET 220px-Henry_Moore_sculpture,_Harvard_Un	404 Not Found	upload.wikimedia.org	9 B	1.28s
GET 220px-Harvard_stadium_2009h.JPG	200 OK	upload.wikimedia.org	8.1 KB	1.4s
GET 220px-Harvard_crimsons_v_brown_2009.JP	200 OK	upload.wikimedia.org	16.2 KB	1.62s
GET 20px-Wiki_letter_w_cropped.svg.png	200 OK	upload.wikimedia.org	916 B	1.5s
GET 32px-Flag_of_Boston.svg.png	200 OK	upload.wikimedia.org	1.1 KB	1.54s
GET 23px-Flaticonpainting.jpg	200 OK	upload.wikimedia.org	877 B	1.56s
GET 38px-Wikisource-logo.svg.png	200 OK	upload.wikimedia.org	2.3 KB	1.63s
GET 12px-Commons-logo.svg.png	200 OK	upload.wikimedia.org	545 B	1.66s
GET 110px-BostonTrip-91.jpg	200 OK	upload.wikimedia.org	4 KB	1.74s
GET 80px-Flag_of_Massachusetts.svg.png	200 OK	upload.wikimedia.org	1.7 KB	1.78s
GET 125px-P9091102.jpg	200 OK	upload.wikimedia.org	4.4 KB	1.77s
GET 125px-Gillette_Stadium.jpg	200 OK	upload.wikimedia.org	5.1 KB	1.82s
33 Requests			711.8 KB	4.42s (onload: 4.47s)



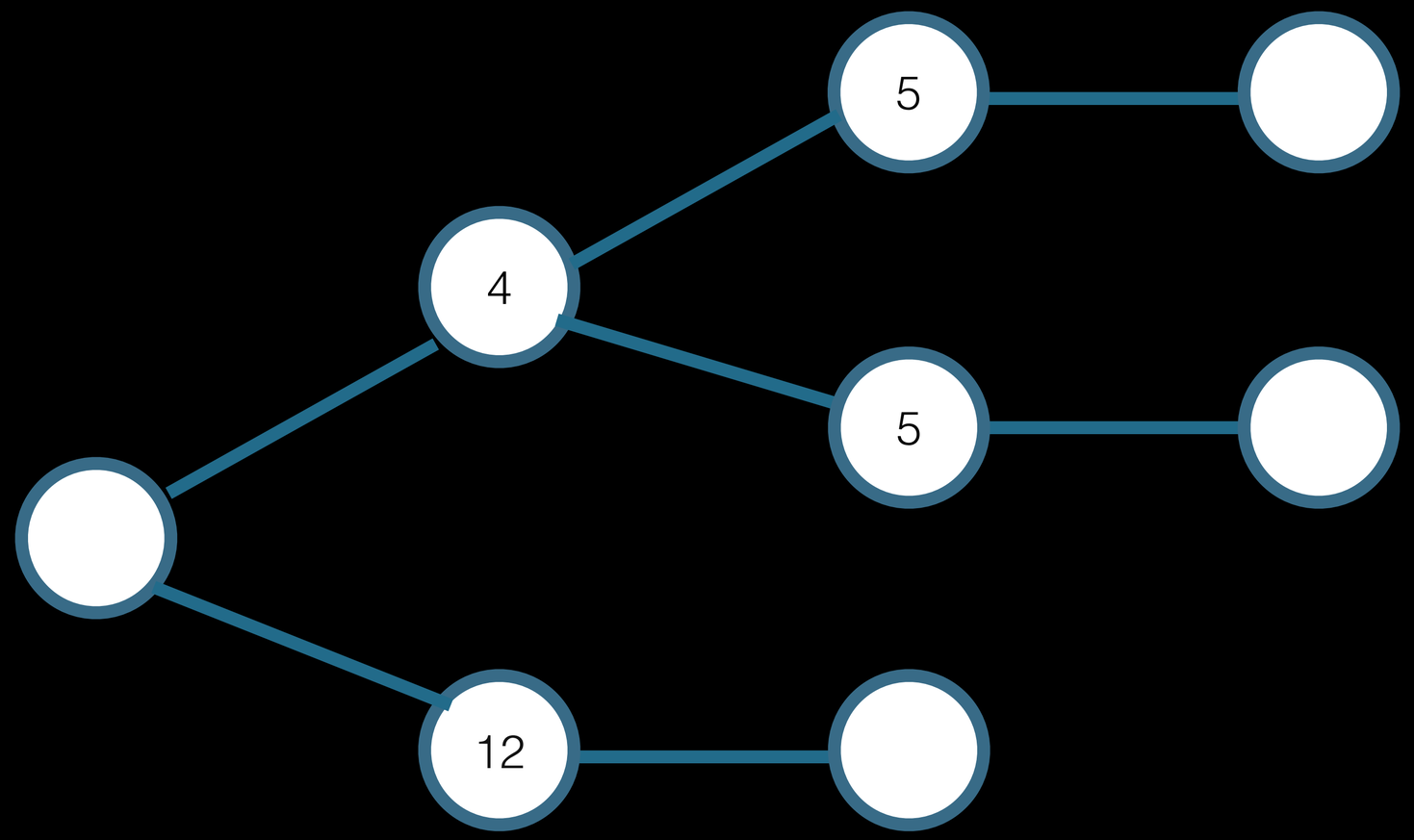
HTTP/2

http://en.m.wikipedia.org/wiki/Harvard_University				
GET Harvard_University	200 OK	en.m.wikipedia.org	329.7 KB	1.25s
GET common.css?version=12012011121954	200 OK	bits.wikimedia.org	3.6 KB	361ms
GET android.css?version=12012011120715	200 OK	bits.wikimedia.org	4.4 KB	360ms
GET jquery-1.7.1.min.js	200 OK	bits.wikimedia.org	91.7 KB	844ms
GET application.js?version=01052011120915	200 OK	bits.wikimedia.org	3.3 KB	170ms
GET opensearch.js?version=12142011129437	200 OK	bits.wikimedia.org	4.5 KB	175ms
GET W_logo_for_Mobile_Frontend.gif	200 OK	upload.wikimedia.org	379 B	162ms
GET close-button.png	200 OK	bits.wikimedia.org	1.9 KB	178ms
GET s-xhdpi.png	404 Not Found	bits.wikimedia.org	9 B	487ms
GET 20px-Padlock-silver.svg.png	200 OK	upload.wikimedia.org	898 B	228ms
GET 220px-Harvard_Wreath_Logo_1.svg.png	200 OK	upload.wikimedia.org	46.5 KB	416ms
GET Harvard_University_logo.PNG	200 OK	upload.wikimedia.org	2.7 KB	300ms
GET 220px-A_Westerly_View_of_the_Colledges_	200 OK	upload.wikimedia.org	8.2 KB	436ms
GET 290px-Rummelt,_Richard_Harvard_Univers	404 Not Found	upload.wikimedia.org	9 B	478ms
GET 220px-Harvard_square_harvard_yard.JPG	200 OK	upload.wikimedia.org	16.5 KB	537ms
GET 170px-Harvard_University_map_(older,_dat	404 Not Found	upload.wikimedia.org	9 B	528ms
GET 220px-John_Harvard_statue_at_Harvard_Un	200 OK	upload.wikimedia.org	13.5 KB	578ms
GET 220px-Massachusetts_Hall,_Harvard_Unive	404 Not Found	upload.wikimedia.org	9 B	496ms
GET 220px-Harvard_yard_winter_2009j.JPG	200 OK	upload.wikimedia.org	14.2 KB	504ms
GET 170px-Harvard_architects.png	200 OK	upload.wikimedia.org	68.4 KB	1.03s
GET 170px-Harvard_memorial_church_winter_2(	200 OK	upload.wikimedia.org	18 KB	657ms
GET 220px-Widener_library_2009.JPG	200 OK	upload.wikimedia.org	10.6 KB	499ms
GET 170px-University_Museum_at_Harvard_001.	200 OK	upload.wikimedia.org	7.8 KB	402ms
GET 220px-Henry_Moore_sculpture,_Harvard_Un	404 Not Found	upload.wikimedia.org	9 B	413ms
GET 220px-Harvard_stadium_2009h.JPG	200 OK	upload.wikimedia.org	6.1 KB	507ms
GET 220px-Harvard_crimsons_v_brown_2009.JP	200 OK	upload.wikimedia.org	16.2 KB	547ms
GET 20px-Wiki_letter_w_cropped.svg.png	200 OK	upload.wikimedia.org	916 B	430ms
GET 32px-Flag_of_Boston.svg.png	200 OK	upload.wikimedia.org	1.1 KB	360ms
GET 21px-Flubaspainting.jpg	200 OK	upload.wikimedia.org	877 B	365ms
GET 38px-Wikisource-logo.svg.png	200 OK	upload.wikimedia.org	2.3 KB	376ms
GET 12px-Commons-logo.svg.png	200 OK	upload.wikimedia.org	545 B	360ms
GET 110px-BostonTrip-91.jpg	200 OK	upload.wikimedia.org	4 KB	420ms
GET 60px-Flag_of_Massachusetts.svg.png	200 OK	upload.wikimedia.org	1.7 KB	346ms
GET 125px-P9091163.jpg	200 OK	upload.wikimedia.org	4.4 KB	334ms
GET 125px-Gillette_Stadium.jpg	200 OK	upload.wikimedia.org	5.1 KB	342ms
35 Requests			711.8 KB	3.86s (onload: 3.09s)

- In HTTP/1, Prioritisation is a browser heuristic*
  - “CSS and JS first, then images…”
  - “This connection for that request…”
- In HTTP/2, it’s **hinted by the client**, **determined by the server**.

* a.k.a. “guessing”

HTTP/2 Priority Hinting



www.youtube.com/watch?v=i_mKY2CQ9Kk

Upload Sign In

Most Hilarious Videos Of Cats 2014 - Best Funny Kitten Compilation

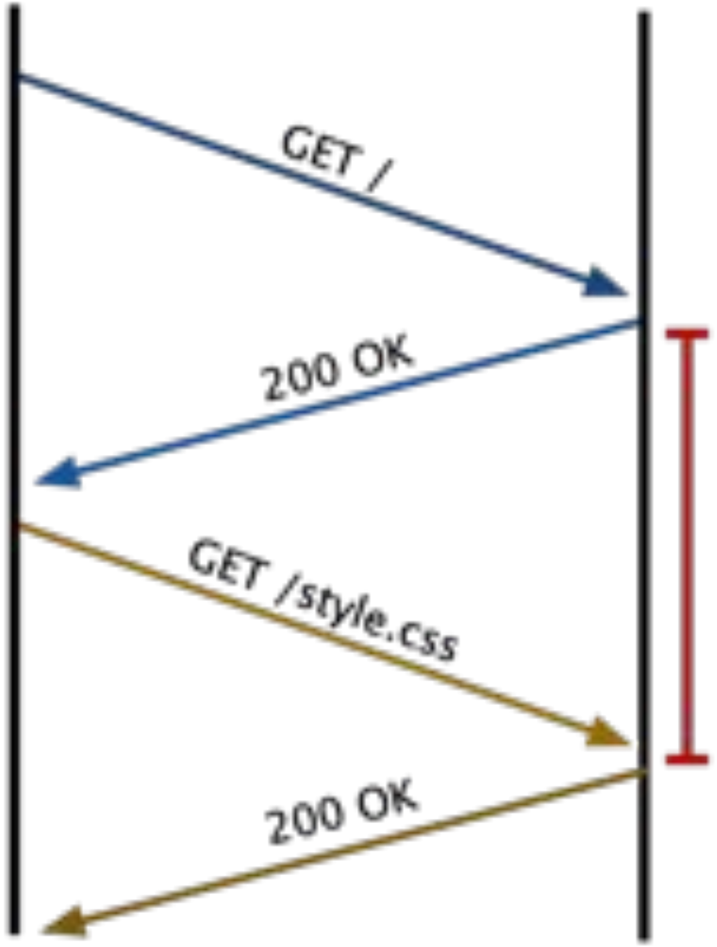
- Servers can do what they like with this information.
  - Only one of several potential sources!
  - Page analysis, RUM over time, etc.
- No prioritisation performs **worse** than HTTP/1.

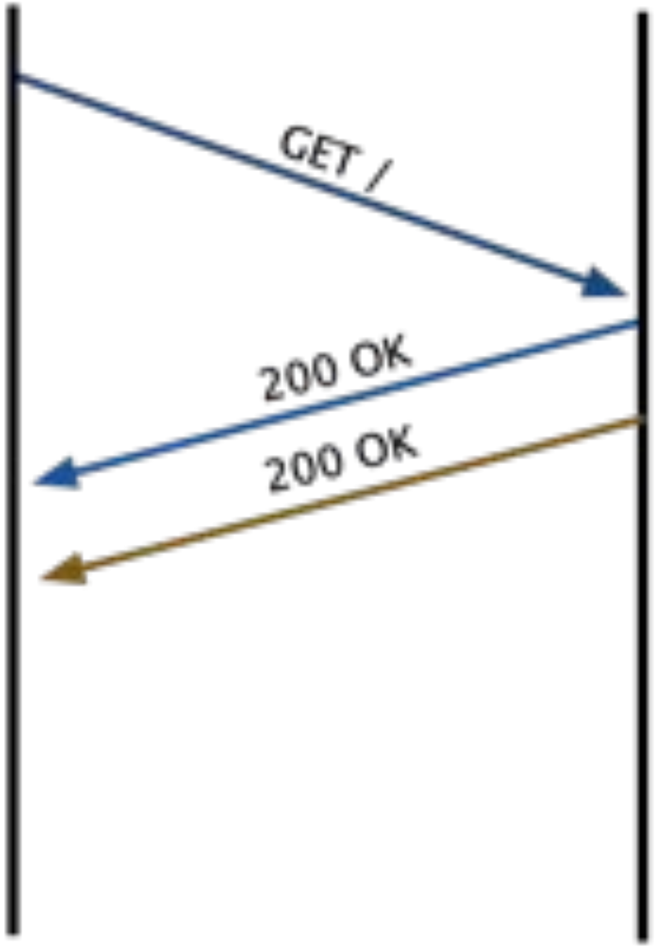
Ask Your Implementation

- How do you handle priority by default?
- What APIs do you expose for effecting prioritisation?
  - How do you handle reprioritisation?
- Do you queue in user space and use TCP_NOTSENT_LOWAT*?

* https://insouciant.org/tech/prioritization-only-works-when-theres-pending-data-to-prioritize/

2. Server Push


GET /
200 OK
GET /style.css
200 OK


GET /
200 OK
200 OK

Benefits of Push

- Avoid a RT without sacrificing **Resource Granularity**!
  - Better cache efficiency
  - Reduced parse / blocking
  - Load what you need
  - Modularity
- Client can refuse push with RST_STREAM

When do I Push?

- **Easy answer**: when you previously inlined / concatenated
- **Creative answer**: when you want to overload it for async data
- **Real answer**: we need research, metrics and tools!
  - *speculative push* is likely a very different beast
  - how will intermediaries handle server push?

3. Header Compression

HTTPbis Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 13, 2015

R. Peon
Google, Inc
H. Ruellan
Canon CRF
September 9, 2014

HPACK - Header Compression for HTTP/2

draft-ietf-httpbis-header-compression-latest

Abstract

This specification defines HPACK, a compression format for efficiently representing HTTP header fields, to be used in HTTP/2.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft takes place on the HTTPBIS working group mailing list (ietf-http-wg@w3.org), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/>.

Working Group information can be found at <http://tools.ietf.org/wg/httpbis/>; that specific to HTTP/2 are at <http://http2.github.io/>.

The changes in this draft are summarized in Appendix D.1.

Status of This Memo

```
GET / HTTP/1.1
Host: www.etsy.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2) AppleWebKit/536.26.14 (KHTML, like Gecko)
Version/6.0.1 Safari/536.26.14
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
DNT: 1
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Cookie: uaid=uaid%3DVdhk5W6sexG-_Y7ZBeQFa3cq7yMQ%26_now%3D1325204464%26_slt%3Ds_LCLVpU%26_kid%3D1%26_ver
%3D1%26_mac%3DlVnlM3hMdb3Cs3hqMVuk_dQEixsqQzUlNYCs9H_Kj8c.; user_prefs=1&2596706699&q0tPzMlJLaoEAA==
Connection: keep-alive
```

```
GET /assets/dist/js/etsy.recent-searches.20121001205006.js HTTP/1.1
Host: www.etsy.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2) AppleWebKit/536.26.14 (KHTML, like Gecko)
Version/6.0.1 Safari/536.26.14
Accept: */*
DNT: 1
Referer: http://www.etsy.com/
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Cookie: autosuggest_split=1; etala=111461200.1476767743.1349274889.1349274889.1349274889.1.0;
etalb=111461200.1.10.1349274889; last_browse_page=%2F; uaid=uaid%3DVdhk5W6sexG-_Y7ZBeQFa3cq7yMQ%26_now
%3D1325204464%26_slt%3Ds_LCLVpU%26_kid%3D1%26_ver%3D1%26_mac
%3DlVnlM3hMdb3Cs3hqMVuk_dQEixsqQzUlNYCs9H_Kj8c.; user_prefs=1&2596706699&q0tPzMlJLaoEAA==
Connection: keep-alive
```

- Simple replacement strategies work surprisingly well
  - e.g., LRU
- Tuning is more important when you mux clients
  - e.g., in a load balancer / reverse proxy
  - look at state commitment as well
- In the long run, **changing how we use HTTP headers will improve compression**

4. TLS and TCP

Is TLS Fast Yet?

https istlsfastyet.com

TLS has exactly one performance problem: it is not used widely enough.

Everything else can be optimized.

Data delivered over an unencrypted channel is insecure, untrustworthy, and trivially intercepted. We owe it to our users to protect the security, privacy, and integrity of their data — all data must be encrypted while in flight and at rest. Historically, concerns over performance have been the common excuse to avoid these obligations, but today that is a false dichotomy. Let's dispel some myths.

CPU & latency costs

The process of establishing and communicating over an encrypted channel introduces additional computational costs. First, there is the asymmetric (public key) encryption used during the TLS handshake. Then, once a shared secret is established,

```
# upgrade to latest
$> openssl version
OpenSSL 1.0.1i 6 Aug 2014

# run benchmarks
```

- initcwnd of 10
- Turn off tcp_slow_start_after_idle
- Experiment with congestion control algorithms

How does {SPDY, HTTP/2} affect Ops?

- TLS
- Load Balancing/Failover
- DoS
- Tools
- Metrics
- Transition Strategies

1. TLS

- HTTP/2 does **not** require https://
- However, Chrome and Firefox have said they won't do plaintext http://
  - … Firefox is experimenting with "Opportunistic Security" for http://
- If you want to get the most users onto HTTP/2, **it has to be https://** (for now)

- Minimum TLS 1.2
- SNI required
- Renegotiation during HTTP prohibited
- Ephemeral Key Exchange required (for forward secrecy)
- AEAD ciphers effectively required
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 w/ P256

2. Failover and Load Balancing

- HTTP/2 flows are **longer-lived**
- So, you can’t assume they’ll go away soon
- This changes load balancing and failover strategies

- **GOAWAY** allows you to gracefully shut down a connection
  - Existing requests drain

- **Alternative Services** acts like CNAME at the HTTP layer
- “Spool up a connection at host:port and talk to it like it’s this origin.”
- It’s graceful; should introduce no latency / pauses
- Load balancing, geo optimisation, taking server out of rotation
- Browser support coming (hopefully)

3. DoS

TCP Connections

- HTTP/2 uses **less connections** by a factor of 4x - 8x!
- Those connections are a **lot** more active, well-utilised
- If you think you’re under attack, you’ve got options
  - Reduce SETTINGS_MAX_CONCURRENT_STREAMS
  - Flow control them
  - GOAWAY

Memory

- **Header compression** = More state per connection
  - default header compression state = **4k**
  - state commitment can be tuned using SETTINGS_HEADER_TABLE_SIZE
- **Buffering** e.g., by intermediaries
  - can be controlled by flow control
    - (stream-level and connection-level)

CPU

- Binary format is **easier to parse**
- **HPACK** is very low overhead (less than gzip)
  - can be dialled down if need be with SETTINGS_HEADER_TABLE_SIZE

Intermediaries

- There are some cases where an intermediary can get in trouble
  - Never forward a frame before you have it all!
  - Never forward a header-bearing block before you have it all!
  - … unless you **really** trust the sender.

4. Tools

packet-h2-14.pcapng [Wireshark 1.99.0 (v1.99.0-rc1-1239-gf67bd96 from unknown)]

Analyze Statistics Telephony Tools Internals Help

Filter: http2 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Data Rate	Info
4	0.00037500	::1	::1	HTTP2	101		SETTINGS
6	0.00061900	::1	::1	HTTP2	197		Magic, SETTINGS, HEADERS
8	0.00073800	::1	::1	HTTP2	95		SETTINGS
10	0.00138800	::1	::1	HTTP2	21974		SETTINGS, HEADERS, DATA, DATA, D
11	0.00141200	::1	::1	HTTP2	984		DATA
13	0.00275600	::1	::1	HTTP2	155		SETTINGS, HEADERS
14	0.00425900	::1	::1	HTTP2	277		HEADERS, HEADERS, HEADERS, HEADE
16	0.00928100	::1	::1	HTTP2	43365		SETTINGS, HEADERS, HEADERS, HEAD
17	0.01328600	::1	::1	HTTP2	99		WINDOW_UPDATE

```
[Header Length: 208]
▸ Header: :status: 200
▸ Header: server: nghttpd nghttp2/0.5.2-DEV
▸ Header: content-length: 22617
▸ Header: cache-control: max-age=3600
▸ Header: date: Sat, 02 Aug 2014 10:50:25 GMT
▸ Header: last-modified: Sat, 02 Aug 2014 07:58:59 GMT
  Padding: <MISSING>
▸ Stream: DATA, Stream ID: 1, Length 4096
▸ Stream: DATA, Stream ID: 1, Length 4096
▸ Stream: DATA, Stream ID: 1, Length 4096
▸ Stream: DATA, Stream ID: 1, Length 4096
▸ Stream: DATA, Stream ID: 1, Length 4096
```

```
0010  00 00 55 a0 06 40 00 00  00 00 00 00
0020  00 00 00 00 00 01 00 00  00 00 00 00
0030  00 00 00 00 00 01 0b b8  dc bc d6 8f
0040  23 fa 80 10 01 56 55 a8  00 00 01 01
0050  aa 63 00 89 aa 62 00 00  00 04 01 00
0060  00 58 01 04 00 00 00 01  88 76 93 aa
0070  52 a9 a7 4a 6b 13 00 5d  b5 c4 b5 fc
0080  84 10 9c 0b bf 58 89 a4  7e 56 1c c5
0090  61 96 dc 34 fd 28 01 29  0d b3 28 20
00a0  8d 82 e0 9b 53 16 8d ff  6c 96 dc 34
00b0  0d b3 28 20 05 a5 00 ed  c6 de b8 db
00c0  00 10 00 00 00 00 00 00  01 0a 0a 3c
00d0  54 59 50 45 20 68 74 6d  6c 3e 0a 3c
```

Frame (21974 bytes) | Decompressed Header (208 bytes)

Header (http2.header), 39 bytes ... Profile: CAPWAP

cURL

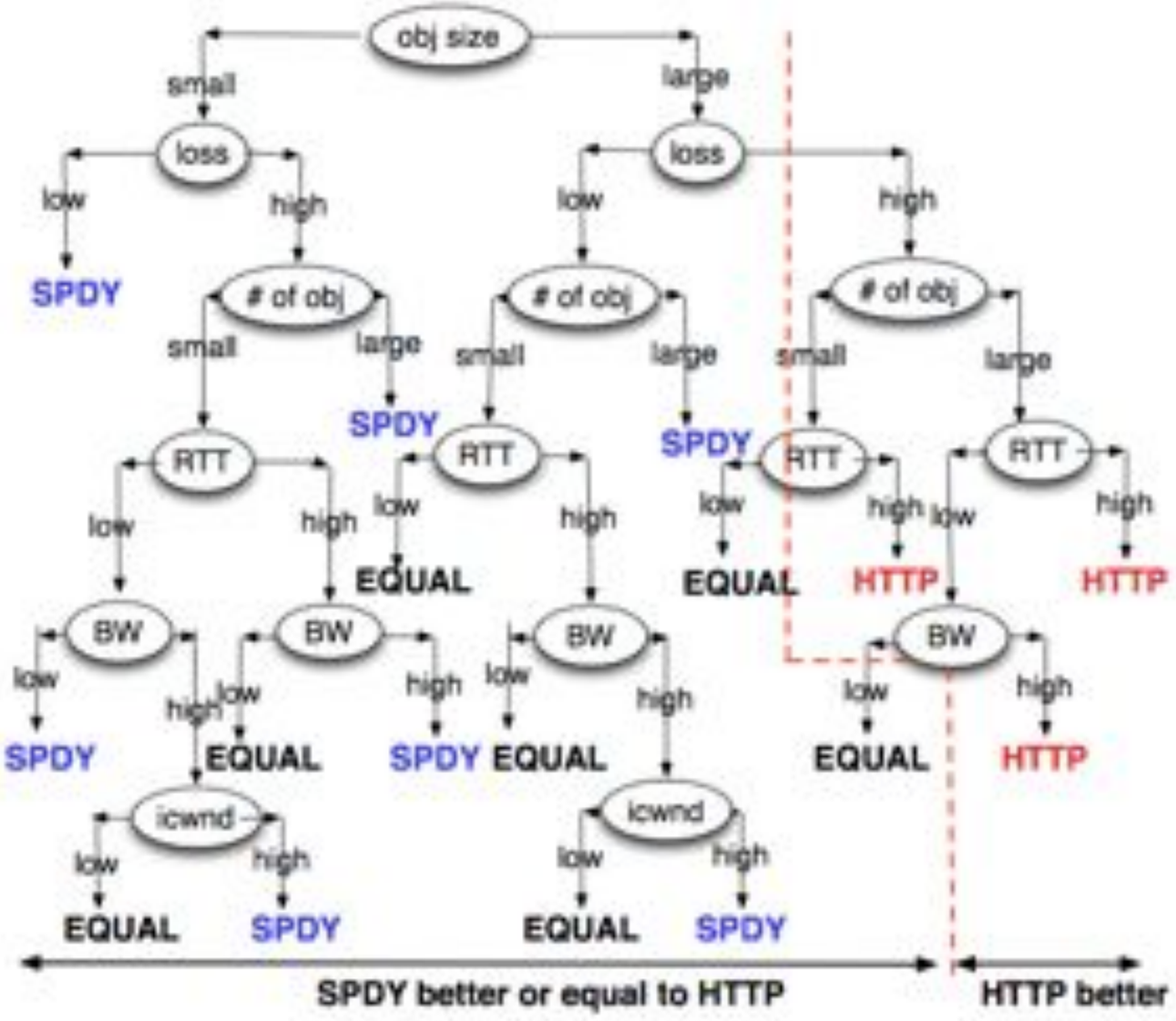
gets those URLs

?

5. Metrics

- **Connection Utilisation**
  - idle periods, concurrency, reset streams, protocol errors
- **HPACK efficiency**
  - hit rates, memory consumption, lost opportunities
  - both directions!
- RUM RUM **RUM**!

6. Transition Strategies



“How Speedy is SPDY?” — Wang, Balasubramanian, Krishnamurthy and Wetherall

Un-Hacking Your Site

Spriting ☞ Separate Resources

Inlining ☞ Server Push

Sharding ☞ Single Host

Concatenation ☞ Separate Resources

Why Un-Hack?

- CSS Spriting delays image downloads (**indirection**)
- Concatenation / inlining / spriting reduce **cache efficiency**
- Concatenation / inlining / spriting encourages **wasted download**
- JS concatenation increases **parse/load overhead**
- Sharding **reduces header compression efficiency, tcp flow gains**

When to Un-Hack?

- Lots of choices:
  - When x% of your traffic supports HTTP/2
  - Dynamically hack per connection
  - Gradual rollout
  - Just Do It

- Switch to HTTPS first
- De-[sprite, concatenate, shard, inline]
- Browser rollout period is a unique opportunity
- Collect metrics!

THE TASMANIAN GOVERNMENT

HYDRO-ELECTRIC POWER

The Master-key to Industry

GREATEST
HYDRO-ELECTRIC
STATE
IN THE
COMMONWEALTH

TASMANIA—
Ideal Climate—
Deep Water Ports—
Cheap Factory Sites—
Abundant Fresh Water Supply

TASMANIA offers the greatest opportunities in Australia for Industrial Enterprise with unlimited power for all purposes

TASMANIA


Questions?