mozilla

# H2 Checkpoint.

After 1 ½ years - a gut check from Firefox's Point of View

# On the Rise

- **H2 is 31% of all Transactions (up from 18% last summer)**
- **44% of all HTTPS Transactions**
- **HTTPS itself is up from 55% to 72% over same period.**

GTFO

- **SPDY successfully being deprecated**
- **Last summer H2:Spdy was 1:1**
- **This summer H2:Spdy is 20:1**

GTFO

- **NPN vs ALPN is harder**
- **Last Summer NPN was 20% of handshakes**
- **This Summer it is 10%**
- **Why?**

# GTFO

- **I wanted one more GTFO slide.**
- **It turns out the only reasons ever given on the Internet for GO_AWAY are NONE  PROTOCOL_ERROR and INTERNAL_ERROR**

# The HTTP/2 Areas of Focus

# The Latency Story

Delay

| Percentile | Desktop RTT (ms) | Mobile RTT(ms) |
|---|---|---|
| 5 | 1 | 11 |
| 25 | 20 | 44 |
| 50 | 79 | 94 |
| 75 | 194 | 184 |
| 95 | 800 | 913 |

# TCP Connection Management

HMM.

- **Median Transactions per connection H2: 8+ .. H1 is 1+**
- **95th Percentile H2: 40 .. H1 is 4**
- **Better than 6X - RST_CANCEL and Coalescing**
- **Only 20% carry more than 6 simultaneous streams**

# Queueing Delays


- **H2 and H1 both have a median delay of 0!**
- **80th percentile wait - H1 100ms, H2 2ms**
- **95th percentile wait - H1 2000ms, H2 16ms**
- **Wait over 100ms - H1 20%, H2 3%**

# Header Compression Matters for Multiplexing


- **Median is great - 90% reduction**
- **80th percentile also good - 75% reduction**
- **Tail is poor. 90th percentile is just 10% reduction**
- **Median request header is 85 bytes.**
- **(Response median size is 42%)**

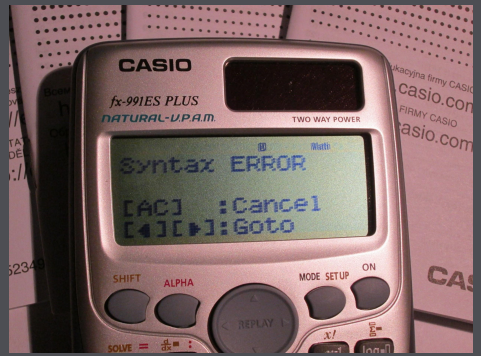
FEELING OKAY?

**So far the checkup in okay. But there are concerns. There are always concerns.**

# Interop Pain - But Getting Better

Framing

Header Syntax




Connection Coalescing

# About Priority; I've got Questions.

# Bad Answers.

99designs

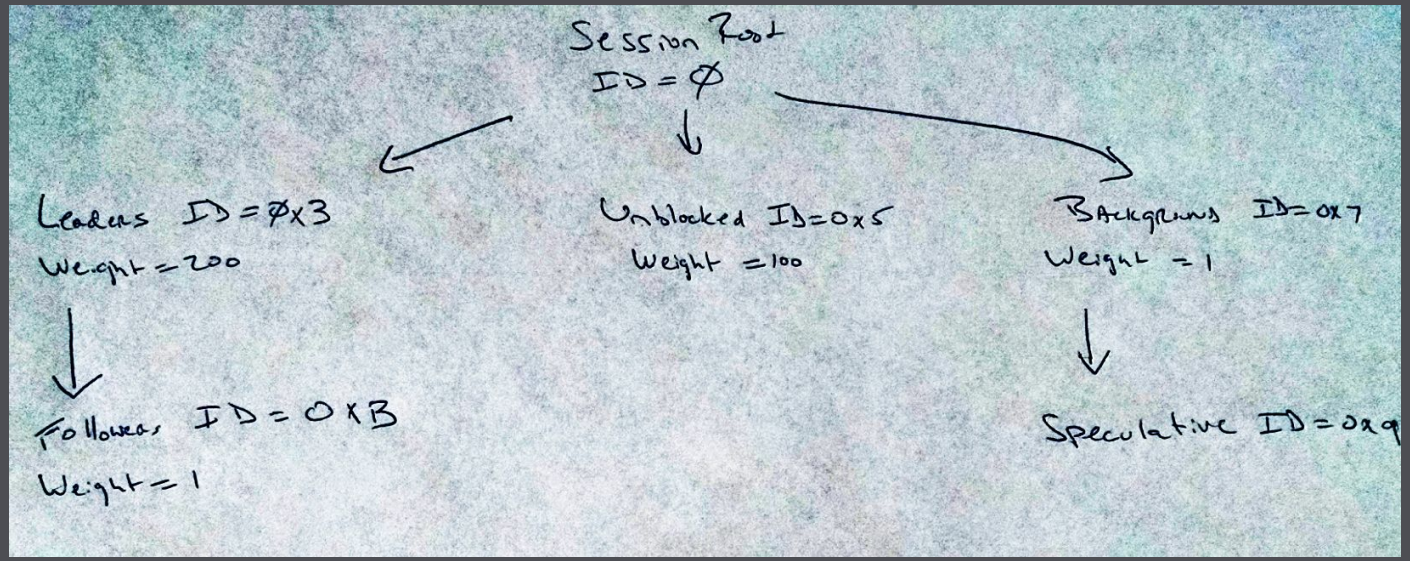
Engineering Blog

## Real–world HTTP/2: 400gb of images per day

**When using HTTP/2, our bandwidth-bound pages take significantly longer to reach visual completion despite loading faster. Why is this?**

HTTP/2 changed the landscape for resource prioritization — the responsibility is now shared between the browser and the server. The browser gives the server hints about priority but it's the server that's in charge of what order the bytes are delivered.

# We Run The Risk of Making Priority Non-Deployable and you cannot do MUX without it



# Speaking of things that are not widely deployed

PUSH

- **Pushes do happen; but rarely. ~0% of connections see one.**
- **Not a lot of consistency are cache management, HEAD, etc..**
- **A W3C webfacing API remains a need. Internal APIs exist.**
- **Can it be proven to be more than a 1-RTT Trick worth the complexity?**

# Speaking of things that are not widely deployed

ALTSVC

- **1000 times less likely to be routed via ALT-SVC than via an explicit proxy. (Does not consider bootstrapping quic).**
- **Roughly same odds of receiving a 308. (Sorry Julian.)**
- **CDNs were big proponents of this for load shedding during standardization - where are we at?**
- **5% of small Alt-Svc in Firefox involves OE.**

THE
END?

# The world changes with your POV. Does this match your view?