http workshop 2019

Overall the workshop was extremely useful. I learned a huge amount and came away much better informed about the current state of HTTP and the direction things are heading in.

These notes should be read in conjunction with the notes (day 1, day 2 and day 3) and the list of Conversation Starters. The following notes use the same order as the list of conversation starters.

Security

H2 - the important bits

Implementations have max stream values that range from 30 to 256 with most using 100. We currently use 200.

Action: ~~Consider reducing default max streams to 100~~. Done

Implementations have max header list sizes that range from 16kB to 4GB. Most use 16kB or 32kB. We use 4GB.

Action: ~~Consider reducing default max header list size.~~ Done

Other values were consistent with typical implementations.

Connection coalescence.

Action: Consider implementing RFC 8336

Action: Review HTTP/2 extension for secondary certificates

Signed Exchanges

Could be handled with a dedicated Servlet. Possibly the Default Servlet in a similar manner to compression.

Might need changes to mapping. Or at least how users think about mapping. One way to route correctly would be to have the default host handle all the signed exchanges.

No action required from us at this point.

Traffic Analysis

No action required from us at this point.

Embedding decryption keys in PCAP

OpenSSL has a callback for writing the TLS key log for debug purposes.

Action: We should add support for this to Tomcat-Native.

Generic overlay networks

No action required from us at this point.

Intercepting QUIC

Testing QUIC / HTTP/3 implementations is going to be tricky.

No action required from us at this point.

HTTP State Tokens

Use an opaque token on the client to reference state held on the server.

Sort of equivalent to a cookie that only holds a session ID.

Would require servers and proxies to maintain much more state.

There are several new(ish) cookie features we aren't using and should consider. See https://scotthelme.co.uk/tough-cookies/

Action: Review new(ish) cookie features

Breaking out of Request/Response

Prefer-Push

Server push has minimal benefit for browsers.

Neater solution to transclusion.

Likely to require application level implementation since you need to know which resources (if any) to push.

No action required from us at this point.

WebSocket

I was surprised at the level of dislike of WebSocket.

Most people in the room wanted to see it go.

No action required from us at this point.

hx URIs

No action required from us at this point.

Extending h2 for bi-directional messaging

This is a potential replacement for WebSocket

Proxy may interfere with HTTP/2 push

Would require Tomcat changes to implement.

No action required from us at this point.

Partial POST replay

Handles failover for POST requests that take longer than the standard drain time.

Would require Tomcat changes to implement.

No action required from us at this point.

HTTP retries

Aims to provide better signalling.

Would require Tomcat changes to implement.

May require Servlet spec changes.

May need to relax HTTP spec.

No action required from us at this point.

Headers

Sec-Fetch-*

Aim to protect authenticated client session

Should be possible to implement in an application. Not sure if a general (e.g. filter) implementation would be possible.

No action required from us at this point.

Origin Policy

Aims to headers that apply site-wide from every response. e.g. CSP, CORS (4k for Twitter!)

Would be static files in a known location for Tomcat. Could also be configuration driven and served by a standard servlet.

No action required from us at this point.

Structured Headers

Actively being worked on in the HTTP working group

Would require a new header parser for Tomcat.

No action required from us at this point.

Web-compatible header values

Standard approach to header parsing.

No action required from us at this point.

Client Hints

Only on secure connections.

Reduce client fingerprinting.

Replace UA and accept-* headers

Could make use of / be replaced by origin policy

May impact on Tomcat's current content negotiation.

No action required from us at this point.

Other

Partially Reliable HTTP

Used video as an example.

If some TCP frames are lost it isn't always worth trying to recover them.

Jitter is worse than lost data.

Would need Servlet API changes to support this.

No action required from us at this point.

SEARCH method

c.f. POST retries

Many POSTs are for searches and could be cached and could be safely re-tried

Is a new method (SEARCH) required.

Depends on being able to normalize the search query.

No action required from us at this point.

VTest

Possible platform for common test cases

Little varnish focused at the moment.

No action required from us at this point.

CDNs

Proxy status header is being discussed.

There is a caching test suite we could use.

Action: Run the caching test suite against Tomcat

EOS detection in H2

Consider the following sequence:

- client req then EOS
- server response then EOS
- server shutsdown
- client sends window update
- TCP reset
- reponse truncated

Send a ping and wait for ACK before shutting down server.

Half-close didn't cover all use cases.

Implies TCP reset packages "jump" the queue.

Action: Check Tomcat's close behaviour

Response sources

pre-fetch - new document

pre-load - resource in document

Need to be aware of browser behaviour for push. See links in day2 notes for additional reading.

No action required from us at this point.

Early hints

103 status code

RFC 8297

Implementation would require Tomcat changes.

No action required from us at this point.

Additional notes

HTTP/2 Priority

Includes the browser's view of prioritisation.

Audio and video add a time factor that isn't currently taken into account.

Action: Test Tomcat with the prioritisation test page

The larger the send buffer, the harder it is to do prioritisation.

Action: Review Tomcat's buffering. Committed buffers make (re-)prioritisation harder.

QUIC & HTTP/3

Consensus among those I spoke to was that Tomcat should wait as things are still in flux. It may be possible to use an external QUIC implementation (like we do OpenSSL).

Action: Investigate TCP fast open. Can Tomcat make use of it? What is involved?

Action: Investigate TLS 1.3 early data. Can Tomcat make use of it? What is involved?

Alt-Src

Discussions continue about different channels for passing this to clients.

It will be needed to HTTP/3.

No action needed until we start HTTP/3 implementation.