Extensible Priorities

Lucas Pardue and Kazuho Oku

draft-ietf-httpbis-priorities

Refresher 1

Priority has

- **urgency level** - u (sh-integer)
- **incremental** flag - i (sh-boolean)

Examples:

- priority: u=2
- priority: u=3, i
- priority: i

Uptake

Implementations with some level of extensible priority:

Chrome

quicly/h2o

ngtcp2/nghttp3

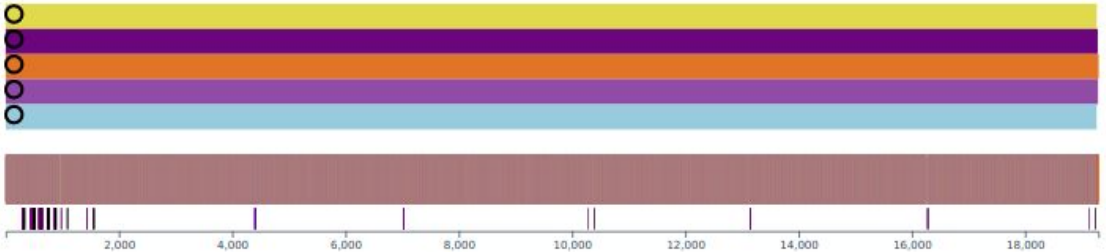
quiche (WIP)

mvfst? others?

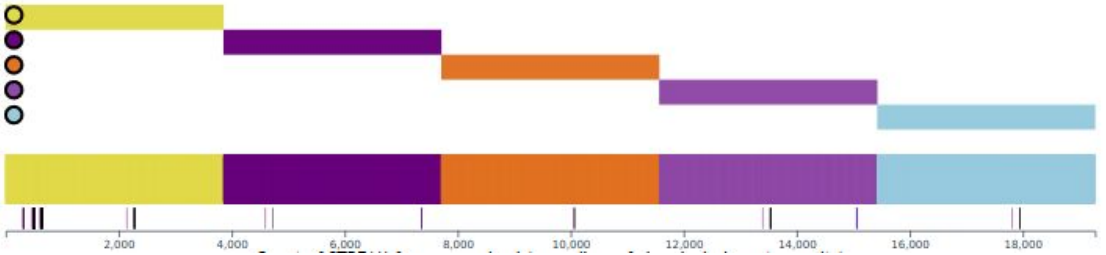
Impact

5 concurrent transfers of 5 MB, all urgency=1

quiche (master)
round-robin



quiche (wip)
FIFO



Overall shape of interop

Some client implementations sending signals.

Server implementations always did scheduling, are they consuming extensible priorities?

Interop activity could help test and measure. Define some test cases to exercise core functionality?

Tatsuhiro Tsujikawa suggested a query string syntax for interop (to bypass signalling difficulties) e.g. https://example.com/image.jpg?**u=3**&**i=1**

Changes in draft 00

Urgency level no longer have strict semantics:

- ~~0 - prerequisite~~
- ~~1 - default~~
- ~~2 through 6 - supplementary~~
- ~~7 - background~~

Use the range how you want, expect server transmits in order, low to high:

- 0 through 7
- Default is ~~still 1~~ 3 (oops!)
- Background is 7

https://tools.ietf.org/rfcdiff?url2=draft-ietf-httpbis-priority-00.txt

Remaining issue: Headers vs Frames

How to signal initial priority?

Discussion on https://github.com/httpwg/http-extensions/issues/1021

Headers

Frames

Current text says

Resources have an initial Priority, signalled in headers.

Resources can be reprioritized using a PRIORITY_UPDATE frame sent on control stream (in HTTP/2 that's stream 0).

PRIORITY_UPDATE carries:

- Stream ID of thing to reprioritize (in HTTP/3 may be a Push ID).
- New priority value, ASCII text encoded using Structured Headers.

Why not have both?

Nobody wants to chop their hand off.

Already have the PRIORITY_UPDATE frame for reprioritization. So just send that before a request?

In practice, Chrome is doing this already, in spite of the text that is says otherwise.

In practice, Chrome’s behaviour has to be accommodated anyway. HTTP/3 has no ordering guarantees between control stream and request streams, a server has to entertain the possibility of receiving a signal before the thing it refers to.

PR 1167

- Formalize that PRIORITY_UPDATE is allowed for initial priority.
  - The name is now a little odd, please put suggestions on a postcard.
- Clients can send PRIORITY_UPDATE.
  - MAY omit header field.
  - Can send PRIORITY_UPDATE first, and then header field. Not much different to a reprioritization event.
- Servers do not send PRIORITY_UPDATE.
- Servers SHOULD buffer PRIORITY_UPDATE and apply it when the element it refers to arrives.
- Ultimately PRIORITY_UPDATE trumps priority header because there is no way to distinguish it from reprioritization.

Desired outcome from this meeting

Proposal to support Headers **and** Frames was put to the list on April 30.

Feedback so far seems supportive. Suggestions for improvements have been incorporated, feels like it is getting ready to merge in readiness for a draft 01.

Some minor things remaining to highlight and discuss:

1. HTTP/3 frame format changes / versioning across priorities drafts.
2. A question for the WG about diagramming HTTP/2 **and** HTTP/3 frames in the same document.

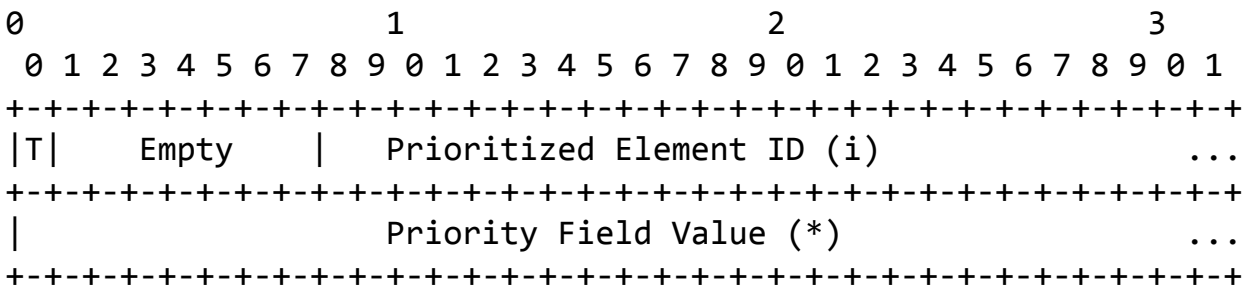
Discuss 1: Frame format change

Issue 1096 - consider two HTTP/3 frames rather than a bit field

Old frame, type **0xF**

T bit distinguishes between prioritizing a Request or Push ID

Priority Field Value is ASCII encoded Structured Headers



New frame proposal - HTTP/3

Remove bitfield, keep ID and value, frame type relates to element type:

- Type 0x1CCB8BBF1F0700 applies to requests
- Type 0x1CCB8BBF1F0701 applies to pushes

```
HTTP/3 PRIORITY_UPDATE Frame {
  Type (i) = 0x1CCB8BBF1F0700..0x1CCB8BBF1F0701,
  Length (i),
  Prioritized Element ID (i),
  Priority Field Value (..),
}
```

Why change the frame type to something horrible?

Breaking change in the frame format.

No way to signal what version of the priorities draft we are using.

Therefore, a risk that endpoints generate and parse the frame with different expectations. Parsing error is a MUST connection error.

- Option 1: Tie this change to a HTTP/3 draft version.
  - Downside is that we may need to iterate priorities faster than we iterate HTTP/3
- Option 2 (this proposal): Pick types for each priorities draft, revert back to 0xF and 0x1 when close to done.

HTTP/2 frame, no-op

The diagram has changed for consistency, has already caused some discussion

```
HTTP/2 PRIORITY_UPDATE Frame {
  Length (24),
  Type (8) = 0xF,
  Flags (8),
  Reserved (1),
  Stream Identifier (31),
  R (1),
  Prioritized Stream ID (31),
  Priority Field Value (..),
}
```

Discuss 2: ASCII vs QUIC-style vs McGuffin

Priorities draft defines frames in HTTP/2 and HTTP/3.

HTTP/3 spec dropped ASCII diagrams in favor of new “QUIC-style” format defining the **entire frame** including **frame header**. This might surprise HTTP/2 people.

Should we:

1. Mix the diagram diagram styles in one document - ASCII **and** QUIC-style
2. Pick one style - ASCII **or** QUIC-style
3. Define only the frame payload (Mike Bishop’s suggestion)
4. Something else / don’t care

Overall shape of the draft

Beyond these slides, some remaining minor issues

#1056: The default priority of a push.

Probably the only discussion-worthy issue, if time permits

If Time permits - Default priority of a push #1056

Server Push: the request and response headers are created by the server (or in an intermediary case, the origin). If either of the two contains a priority signal, resolve the merge as normal.

The question is what to do when there is **no priority signal**.

If Time permits - Default priority of a push #1056

Omitting priority signal in **normal** request/response: urgency=**3**, incremental=**false**

RFC 7540, Section 5.3.5 "*pushes initially depend on their associated stream … default weight of 16*"

Extensible priorities has no **dependency**, so what are a push's default urgency and incremental?

If Time permits - Default priority of a push #1056

Possible push defaults:

1) Same as the associated stream (aka “request that triggered the push”).
2) “An urgency one less than the associated stream”.
3) No default.

Considerations for picking:

- No single best value.
- Server can not use PRIORITY_UPDATE to signal push priority (as defined now)
- Performance benefits are hard to measure.
- Performance degradation is very possible if wrong value is picked
  - Especially if pushed resources priority cause bad scheduling of other resources

Questions?