Extensible Prioritization Scheme for HTTP

Lucas Pardue and Kazuho Oku

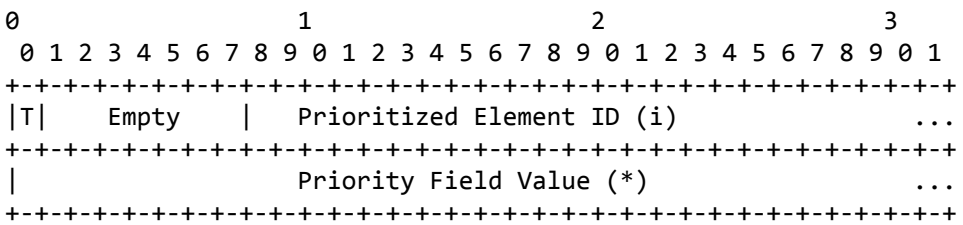
draft-ietf-httpbis-priority


HEADERS & FRAMES
HEADERS
FRAMES

New in -02: H3 PRIORITY_UPDATE breaking change

-01

- Type **0xF**
- **T** bit - Element ID is Request or Push
- Priority Field Value is ASCII encoded Structured Headers



-02

- Replace bitfield with frame type
- **0xF0700** applies to Request
- **0xF0701** applies to Push

```
HTTP/3 PRIORITY_UPDATE Frame {
  Type (i) = 0xF0700..0xF0701,
  Length (i),
  Prioritized Element ID (i),
  Priority Field Value (..),
}
```

New in -02

1. Add consideration for server scheduling (Section 9)
2. Remove specific instruction about intermediary fairness (Section 10.2)
3. Describe considerations when clients use PRIORITY_UPDATE (Section 6)

   Problem: Race condition where frame arrives before request

   Solution: Servers SHOULD buffer **most recent frame** and apply it once steam is opened.

   Need to add some further text on how servers can limit the state commitment.

Open issues

- "Within the stream limit" in HTTP/2? #1261
- The default priority of a pushed request #1056

#1261 - "Within the stream limit" in HTTP/2? (1)

For HTTP/2 PRIORITY_UPDATE frame we say:

“The Prioritized Stream ID MUST be within the stream limit.”

Mike Bishop points out that

*“*HTTP/2 *specifies a maximum number of concurrently active streams, with streams which are either "open" or "half-open" counting toward the limit regardless of the ID”*

Are we attempting to redefine HTTP/2? No (if we can avoid it).

#1261 - "Within the stream limit" in HTTP/2? (2)

Kazuho suggests we rephrase the requirement to concurrency window size.

And while we’re fixing this we can probably add more text about buffering and limits in Section 6.

Implementation experience can help make this better.

Default priority of a push #1056 (1)

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**.

Default priority of a push #1056 (2)

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?

Default priority of a push #1056 (3)

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 cannot use PRIORITY_UPDATE to signal push priority
- 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?