Adopting the Bundle Format

Jeffrey Yasskin
WPACK — IETF 110
2021-03-12

draft-yasskin-wpack-bundled-exchanges-04
Goal

Agree to adopt draft-yasskin-wpack-bundled-exchanges as a starting point for further design.
Structure of draft-yasskin-wpack-bundled-exchanges

● Semantics
● Performance expectations
● Format w/o algorithms
  ○ Invariant bit + Sections + trailing length
  ○ Core section definitions
● Security considerations
Semantics

"A bundle is logically a set of HTTP representations, themselves represented by HTTP response messages."

Representations are named by their URLs + content negotiation information (but see a later slide).
Performance

Random Access

- $O(\text{index size})$ before starting to return resource bytes

Streaming

- Sender needs all resource sizes before starting.
- Receiver returns a resource $O(1)$ bytes behind what it has received.
Overall format

CBOR

1. Magic number
2. Version number
3. Primary/fallback URL (but see a later slide)
4. Section index
5. Sections
6. Total size
Core Sections

- Index: Response name -> byte range in Responses
- Manifest: App manifest, EPUB Package Document, etc. (but see a later slide)
- Critical: "Reject the bundle if you don't support these extension sections"
- Responses

There's an IANA registry for extension sections.
A few open PRs

Hum to accept or reject them for now?

We can change our minds later.
Motivation for these changes

Focus the initially-adopted I-D on what's needed for all use cases.

Plan to adopt more-focused extension I-Ds to support particular use cases.

The subresource loading use case seems to need the fewest features, so it drives these proposals.
Move the primary/fallback URL to a section (#617)

- Designed to allow a client to drop compatibility with old versions.
- Not all use cases have a primary URL
  - Representing a whole web page or iframe does.
  - Representing a group of subresources doesn't.
- If the bundle version isn't supported, much more efficient for the server to avoid sending a bundle in the first place.
Remove content negotiation (#618)

- **Online**
  - Online use cases will probably content-negotiate for the bundle as a whole.
  - Including multiple representations for 1 resource would waste space.

- **Offline**
  - Offline use cases might use this to support multiple languages.
  - Format negotiation would probably use a least-common-denominator format instead.

- **CDNs and caches** might like content negotiation info.
Move the manifest section out of the core (#619)

- Top-level document refers to its manifest.
- Manifest refers to the top-level document.
- Naming both at the top level saves parsing one in cases the client needs the other.
- Maybe doesn't pay for the complexity.
Remove the critical section (\#619)

- Better to negotiate for a bundle the client understands than reject it after the fact.
- Hard to add this later.