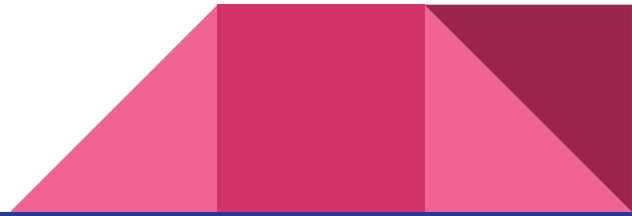
Client Hint Reliability

David Benjamin (davidben@google.com)
HTTPWG 2020 Interim

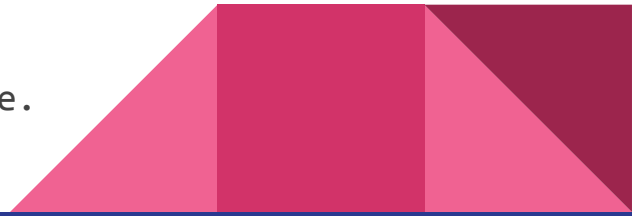
Client Hints

- https://tools.ietf.org/html/draft-ietf-httpbis-client-hints-15
  - In AUTH48 state
- Moves HTTP content negotiation from passive to active
  - Server declares request headers it is interested in
  - Client sends request headers it is willing to send
  - Client maintains a cache of server preferences
- Bandwidth and privacy improvements
  - No wasting bytes on unused request headers
  - Sites need to request fingerprintable surfaces (easier to measure, monitor, budget, etc.)



Inconsistent Behavior

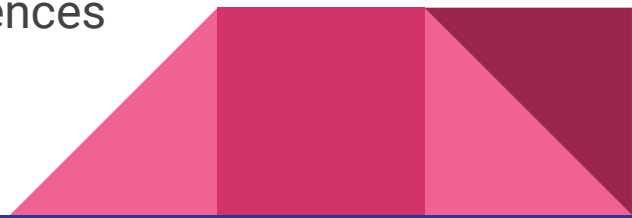
```
GET /first-page HTTP/1.1
                    HTTP/1.1 200 OK
                    Accept-CH: Device-Memory
                    Vary: Device-Memory

                    Here’s the default version of the page.
                    <a href=“/second-page”>Next page</a>


GET /second-page HTTP/1.1
Device-Memory: 0.5
                    HTTP/1.1 200 OK
                    Accept-CH: Device-Memory
                    Vary: Device-Memory

                    Here’s the low-memory version of the page.
```



The Reliability Problem

- Server preferences are delayed by one request
  - Hints are missing on first page visit
  - Changes in server preferences apply late
- Okay for optimizations, not for meaningful content variations
- Clearing Accept-CH cache can break pages
- Example: User-Agent Client Hints
  - https://wicg.github.io/ua-client-hints/

Goal: The client should reliably incorporate server preferences into request. (It may still decline to send the hint!)



Critical-CH

- Server does not know if client would have sent header
- Client does not know if content variation is meaningful
- Critical-CH header
  - Contains a list of client hints that would meaningfully change *this resource*
  - Like Vary, but tells the client this is worth an RTT hit
  - Client updates Accept-CH cache and decides if it would have sent a listed header
  - If so, cancel the old stream and retry the request
  - Otherwise, use the response as is
- Accept-CH cache can now be freely cleared

Critical-CH Example

The client initially sends no hints.

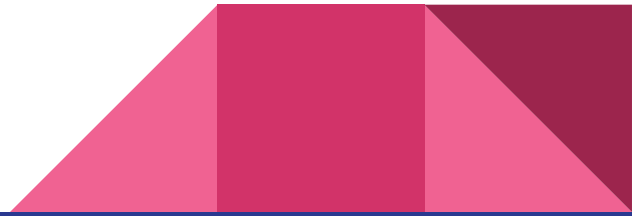
```
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Accept-CH: Sec-CH-Example, Sec-CH-Example-2
Vary: Sec-CH-Example
Critical-CH: Sec-CH-Example

Here’s the default version of the page.
```

If the client would have sent the hints, it retries. Otherwise, it uses the resource as-is.

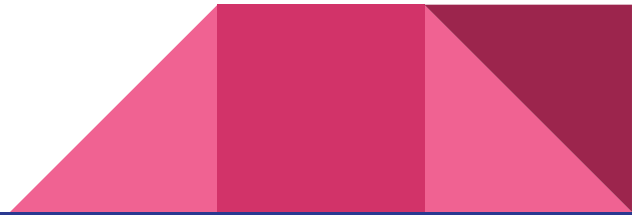
```
GET / HTTP/1.1
Sec-CH-Example: 1
Sec-CH-Example-2: 2
```

```
HTTP/1.1 200 OK
Accept-CH: Sec-CH-Example, Sec-CH-Example-2
Vary: Sec-CH-Example
Critical-CH: Sec-CH-Example

Here’s a more specific version of the page.
```



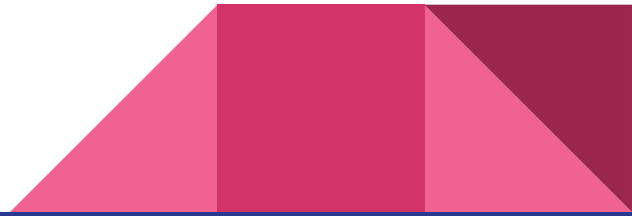
ACCEPT_CH and ALPS

- Retries cost a round-trip
- TLS 1.3 establishes encryption earlier
- Send server preferences in ACCEPT_CH frame, alongside SETTINGS
  - But SETTINGS are not reliable.
- Application Layer Protocol Settings (ALPS)
  - draft-vvv-tls-alps and draft-vvv-httpbis-alps
  - Protocol-specific data sent in TLS EncryptedExtensions
  - Like ALPN, available before application data
  - Rationalizes SETTINGS, NewSessionTicket, and 0-RTT interaction
    - https://github.com/quicwg/base-drafts/issues/3086
  - Some H2/H3 settings otherwise not possible
    - https://github.com/quicwg/base-drafts/issues/3622



Alternatives Considered

- Only have ACCEPT_CH?
  - Not reliable in edge cases, so just an optimization
    - Cross-name connection reuse
    - Long-lived connections
    - Older server software
- Only have Critical-CH?
  - RTT hit on all first page loads is prohibitive
- Vary instead of Critical-CH?
  - Client cannot distinguish between optimizations and meaningful differences
- SETTINGS_ACCEPT_CH instead of new frame?
  - HTTP/2 settings can only be integer-valued



Open Questions

- Layering between Client Hints, HTTP, and TLS
  - ACCEPT_CH and ALPS are one way to layer things
  - draft-bishop-httpbis-extended-settings-01?
  - TLS 1.3 half-RTT data?
    - This one is less practical than it sounds.
- Layering between HTTP intermediary and origin
  - .well-known resource?
  - Origin Policy?
- Service Workers

Questions?

- https://tools.ietf.org/html/draft-davidben-http-client-hint-reliability-01
- https://tools.ietf.org/html/draft-vvv-httpbis-alps-00
- https://tools.ietf.org/html/draft-vvv-tls-alps-01