HTTP Working Group
Internet-Draft
Intended status: Experimental
Expires: October 20, 2017

I. Grigorik
Google

April 18, 2017

HTTP Client Hints
draft-ietf-httpbis-client-hints-04

Abstract

An increasing diversity of Web-connected devices and software capabilities has created a need to deliver optimized content for each device.

This specification defines a set of HTTP request header fields, colloquially known as Client Hints, to address this. They are intended to be used as input to proactive content negotiation; just as the Accept header field allows user agents to indicate what formats they prefer, Client Hints allow user agents to indicate device and agent specific preferences.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/ .

Working Group information can be found at http://httpwg.github.io/ ; source code and issues list for this draft can be found at https://github.com/httpwg/http-extensions/labels/client-hints .

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 20, 2017.

Table of Contents

1. Introduction

There are thousands of different devices accessing the web, each with different device capabilities and preference information. These device capabilities include hardware and software characteristics, as well as dynamic user and client preferences.

One way to infer some of these capabilities is through User-Agent (Section 5.5.3 of [RFC7231]) header field detection against an established database of client signatures. However, this technique requires acquiring such a database, integrating it into the serving path, and keeping it up to date. However, even once this infrastructure is deployed, user agent sniffing has numerous limitations:

- User agent detection cannot reliably identify all static variables
- User agent detection cannot infer any dynamic client preferences
- User agent detection requires an external device database
- User agent detection is not cache friendly

A popular alternative strategy is to use HTTP cookies ([RFC6265]) to communicate some information about the user agent. However, this approach is also not cache friendly, bound by same origin policy, and imposes additional client-side latency by requiring JavaScript execution to create and manage HTTP cookies.

This document defines a set of new request header fields that allow user agent to perform proactive content negotiation (Section 3.4.1 of [RFC7231]) by indicating device and agent specific preferences, through a mechanism similar to the Accept header field which is used to indicate preferred response formats.

Client Hints does not supersede or replace the User-Agent header field. Existing device detection mechanisms can continue to use both mechanisms if necessary. By advertising its capabilities within a request header field, Client Hints allows for cache friendly and proactive content negotiation.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234] with the list rule extension defined in [RFC7230], Appendix B. It includes by reference the DIGIT rule from [RFC5234] and the OWS and field-name rules from [RFC7230].

2. Client Hint Request Header Fields

A Client Hint request header field is a HTTP header field that is used by HTTP clients to indicate configuration data that can be used by the server to select an appropriate response. Each one conveys client preferences that the server can use to adapt and optimize the response.

2.1. Sending Client Hints

Clients control which Client Hints are sent in requests, based on their default settings, user configuration and/or preferences. Implementers might provide user choice mechanisms so that users may balance privacy concerns with bandwidth limitations. Implementations specific to certain use cases or threat models might avoid transmitting these headers altogether, or limit them to secure contexts or authenticated sessions. Implementers should be aware that explaining the privacy implications of passive fingerprinting or network information disclosure may be challenging.

The client and server, or an intermediate proxy, can use an opt-in mechanism to negotiate which fields should be reported to allow for efficient content adaption.

2.2. Server Processing of Client Hints

When presented with a request that contains one or more client hint headers, servers can optimize the response based upon the information in them. When doing so, and if the resource is cacheable, the server MUST also generate a Vary response header field (Section 7.1.4 of [RFC7231]), and optionally Key ([I-D.ietf-httpbis-key]), to indicate which hints can affect the selected response and whether the selected response is appropriate for a later request.

Further, depending on the hint used, the server can generate additional response header fields to convey related values to aid client processing. For example, this specification defines "Content-

DPR" response header field that needs to be returned by the server when the "DPR" hint is used to select the response.

2.2.1. Advertising Support via Accept-CH header field

Servers can advertise support for Client Hints using the Accept-CH header field or an equivalent HTML meta element with http-equiv attribute ([W3C.REC-html5-20141028]).

```
Accept-CH = #field-name
```

For example:

```
Accept-CH: DPR, Width, Viewport-Width
```

When a client receives Accept-CH, or if it is capable of processing the HTML response and finds an equivalent HTML meta element, it can treat it as a signal that the application is interested in receiving specified request header fields that match the advertised field-values; subresource requests initiated as a result of processing the response from the server that includes the Accept-CH opt-in can include the request header fields that match the advertised field-values.

For example, based on Accept-CH example above, a user agent could append DPR, Width, and Viewport-Width header fields to all subresource requests initiated by the page constructed from the response.

2.2.2. The Accept-CH-Lifetime header field

Servers can ask the client to remember an origin-wide Accept-CH preference for a specified period of time to enable delivery of Client Hints on all subsequent requests to the origin, and on subresource requests initiated as a result of processing a response from the origin.

```
Accept-CH-Lifetime = #delta-seconds
```

The field-value indicates that the Accept-CH preference should be considered stale after its age is greater than the specified number of seconds.

```
Accept-CH: DPR, Viewport-Width
Accept-CH-Lifetime: 86400
```

For example, based on the Accept-CH and Accept-CH-Lifetime example above, a user agent could persist an origin-wide Accept-CH preference

for up to 86400 seconds (1 day). Then, if a request is initiated to the same origin before the preference is stale (e.g. as a result of a navigation to the origin, or fetching a subresource from the origin) the client could append the requested header fields (DPR and Viewport-Width in this example) to the request and any subresource requests initiated as a result of processing a response from same origin.

2.2.3. Interaction with Caches

When selecting an optimized response based on one or more Client Hints, and if the resource is cacheable, the server needs to generate a Vary response header field ([RFC7234]) to indicate which hints can affect the selected response and whether the selected response is appropriate for a later request.

```
Vary: DPR
```

Above example indicates that the cache key needs to include the DPR header field.

```
Vary: DPR, Width, Downlink
```

Above example indicates that the cache key needs to include the DPR, Width, and Downlink header fields.

Client Hints MAY be combined with Key ([I-D.ietf-httpbis-key]) to enable fine-grained control of the cache key for improved cache efficiency. For example, the server can return the following set of instructions:

```
Key: DPR;partition=1.5:2.5:4.0
```

Above example indicates that the cache key needs to include the value of the DPR header field with three segments: less than 1.5, 1.5 to less than 2.5, and 4.0 or greater.

```
Key: Width;div=320
```

Above example indicates that the cache key needs to include the value of the Width header field and be partitioned into groups of 320: 0-320, 320-640, and so on.

```
Key: Downlink;partition=0.5:1.0:3.0:5.0:10
```

Above example indicates that the cache key needs to include the (Mbps) value of the Downlink header field with six segments: less

than 0.5, 0.5 to less than 1.0, 1.0 to less than 3.0, 3.0 to less than 5.0, 5.0 to less than 10; 10 or higher.

3. Client Hints

3.1. The DPR header field

The "DPR" request header field is a number that indicates the client's current Device Pixel Ratio (DPR), which is the ratio of physical pixels over CSS px (Section 5.2 of [W3C.CR-css-values-3-20160929]) of the layout viewport (Section 9.1.1 of [CSS2]) on the device.

```
DPR = 1*DIGIT [ "." 1*DIGIT ]
```

If DPR occurs in a message more than once, the last value overrides all previous occurrences.

3.1.1. Confirming Selected DPR

The "Content-DPR" response header field is a number that indicates the ratio between physical pixels over CSS px of the selected image response.

```
Content-DPR = 1*DIGIT [ "." 1*DIGIT ]
```

DPR ratio affects the calculation of intrinsic size of image resources on the client - i.e. typically, the client automatically scales the natural size of the image by the DPR ratio to derive its display dimensions. As a result, the server MUST explicitly indicate the DPR of the selected image response whenever the DPR hint is used, and the client MUST use the DPR value returned by the server to perform its calculations. In case the server returned Content-DPR value contradicts previous client-side DPR indication, the server returned value MUST take precedence.

Note that DPR confirmation is only required for image responses, and the server does not need to confirm the resource width as this value can be derived from the resource itself once it is decoded by the client.

If Content-DPR occurs in a message more than once, the last value overrides all previous occurrences.

3.2. The Width header field

The "Width" request header field is a number that indicates the desired resource width in physical px (i.e. intrinsic size of an image). The provided physical px value is a number rounded to the smallest following integer (i.e. ceiling value).

```
Width = 1*DIGIT
```

If the desired resource width is not known at the time of the request or the resource does not have a display width, the Width header field can be omitted. If Width occurs in a message more than once, the last value overrides all previous occurrences.

3.3. The Viewport-Width header field

The "Viewport-Width" request header field is a number that indicates the layout viewport width in CSS px. The provided CSS px value is a number rounded to the smallest following integer (i.e. ceiling value).

```
Viewport-Width = 1*DIGIT
```

If Viewport-Width occurs in a message more than once, the last value overrides all previous occurrences.

3.4. The Downlink header field

The "Downlink" request header field is a number that indicates the client's maximum downlink speed in megabits per second (Mbps).

```
Downlink = 1*DIGIT [ "." 1*DIGIT ]
```

If Downlink occurs in a message more than once, the minimum value should be used to override other occurrences.

3.5. The Save-Data header field

The "Save-Data" request header field consists of one or more tokens that indicate client's preference for reduced data usage, due to high transfer costs, slow connection speeds, or other reasons.

```
Save-Data = sd-token *( OWS ";" OWS [sd-token] )
sd-token = token
```

This document defines the "on" sd-token value, which is used as a signal indicating explicit user opt-in into a reduced data usage mode on the client, and when communicated to origins allows them to

deliver alternate content honoring such preference - e.g. smaller image and video resources, alternate markup, and so on. New token and extension token values can be defined by updates to this specification.

4. Examples

For example, given the following request header fields:

```
DPR: 2.0
Width: 320
Viewport-Width: 320
```

The server knows that the device pixel ratio is 2.0, that the intended display width of the requested resource is 160 CSS px (320 physical pixels at 2x resolution), and that the viewport width is 320 CSS px.

If the server uses above hints to perform resource selection for an image asset, it must confirm its selection via the Content-DPR response header to allow the client to calculate the appropriate intrinsic size of the image response. The server does not need to confirm resource width, only the ratio between physical pixels and CSS px of the selected image resource:

```
Content-DPR: 1.0
```

The Content-DPR response header field indicates to the client that the server has selected resource with DPR ratio of 1.0. The client can use this information to perform additional processing on the resource - for example, calculate the appropriate intrinsic size of the image resource such that it is displayed at the correct resolution.

Alternatively, the server could select an alternate resource based on the maximum downlink speed advertised in the request header fields:

```
Downlink: 0.384
```

The server knows that the client's maximum downlink speed is 0.384Mbps (GPRS EDGE), and it can use this information to select an optimized resource - for example, an alternate image asset, stylesheet, HTML document, media stream, and so on.

5. Security Considerations

The request header fields defined in this specification expose information that is already available to Web applications in the browser runtime itself (e.g., using JavaScript and CSS). For example, the application can obtain viewport width, image display width, and device pixel ratio via JavaScript, or through the use of CSS media queries and unique resource URLs even if JavaScript is disabled. However, servers that gather this information through such mechanisms are typically observable (e.g., you can see that they're using JavaScript to gather it), whereas servers' use of the header fields introduced by this specification is not observable. Section 2.1 discusses potential mitigations.

For example, sending Client Hints on all requests can make information about the user's environment available to origins that otherwise did not have access to this data, which may or may not be the desired outcome - e.g. this may enable an image optimization service to deliver a tailored asset, and it may reveal same information about the user to other origins that may not have had access to it before. Similarly, sending highly granular data, such as image and viewport width may help identify users across multiple requests. Restricting such field values to an enumerated range, where the user agent advertises a threshold value that is close but is not an exact representation of the current value, can help mitigate the risk of such fingerprinting.

Implementers ought to provide mechanisms and policies to control how and when such hints are advertised. For example, they could require origin opt-in via Accept-CH; clear remembered opt-in, as set by Accept-CH-Lifetime, when site data, browsing history, browsing cache, or similar, are cleared; restrict delivery to same origin subrequests; limit delivery to requests that already carry identifying information (e.g. cookies); modify delivery policy when in an "incognito" or a similar privacy mode; enable user configuration and opt in, and so on.

6. IANA Considerations

This document defines the "Accept-CH", "DPR", "Width", and "Downlink" HTTP request fields, "Content-DPR" HTTP response field, and registers them in the Permanent Message Header Fields registry.

6.1. Accept-CH

- Header field name: Accept-CH
- Applicable protocol: HTTP
- Status: standard

- Author/Change controller: IETF
- Specification document(s): Section 2.2.1 of this document
- Related information: for Client Hints

6.2. Accept-CH-Lifetime

- Header field name: Accept-CH-Lifetime
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 2.2.2 of this document
- Related information: for Client Hints

6.3. Content-DPR

- Header field name: Content-DPR
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.1.1 of this document
- Related information: for Client Hints

6.4. Downlink

- Header field name: Downlink
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.4 of this document
- Related information: for Client Hints

6.5. DPR

- Header field name: DPR
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.1 of this document
- Related information: for Client Hints

6.6. Save-Data

- Header field name: Save-Data
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.5 of this document
- Related information: for Client Hints

6.7. Viewport-Width

- Header field name: Viewport-Width
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.3 of this document
- Related information: for Client Hints

6.8. Width

- Header field name: Width
- Applicable protocol: HTTP
- Status: standard
- Author/Change controller: IETF
- Specification document(s): Section 3.2 of this document
- Related information: for Client Hints

7. References

7.1. Normative References

[CSS2] Bos, B., Celic, T., Hickson, I., and H. Lie, "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", W3C Recommendation REC-CSS2-20110607, June 2011, <http://www.w3.org/TR/2011/REC-CSS2-20110607>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

[W3C.CR-css-values-3-20160929] Atkins, T. and E. Etemad, "CSS Values and Units Module Level 3", World Wide Web Consortium CR CR-css-values-3-20160929, September 2016, <https://www.w3.org/TR/2016/CR-css-values-3-20160929>.

[W3C.REC-html5-20141028] Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, T., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014, <http://www.w3.org/TR/2014/REC-html5-20141028>.

7.2. Informative References

[I-D.ietf-httpbis-key] Fielding, R. and M. Nottingham, "The Key HTTP Response Header Field", draft-ietf-httpbis-key-01 (work in progress), March 2016.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <http://www.rfc-editor.org/info/rfc6265>.

Appendix A. Changes

A.1. Since -00

- Issue 168 (make Save-Data extensible) updated ABNF.
- Issue 163 (CH review feedback) editorial feedback from httpwg list.
- Issue 153 (NetInfo API citation) added normative reference.

A.2. Since -01

- Issue 200: Moved Key reference to informative.
- Issue 215: Extended passive fingerprinting and mitigation considerations.
- Changed document status to experimental.

A.3. Since -02

- Issue 239: Updated reference to CR-css-values-3
- Issue 240: Updated reference for Network Information API
- Issue 241: Consistency in IANA considerations
- Issue 250: Clarified Accept-CH

A.4. Since -03

- Issue 284: Extended guidance for Accept-CH
- Issue 308: Editorial cleanup
- Issue 306: Define Accept-CH-Lifetime

A.5. Since -04

- None

Author's Address

Ilya Grigorik
Google

Email: ilya@igvita.com
URI: https://www.igvita.com/