

HTTP Client Hints Reliability

victortan@

HTTP Client Hints

- [RFC 8942](#)
- Proactive content negotiation
- HTTP headers about the client's device, network, and user preferences
- Performance - only send headers actually going to be used
- Privacy - shift from passive fingerprinting to active requests
 - Not allow to identify individual users easily
 - Address relevant user cases

Why Client Hints

- Organize the information better
- Minimize the passive fingerprinting - request only need instead of broadcasting
- Low and high entropy client hints - reduce default entropy
- Easy to measure and control total entropy - privacy focused approach

Why Low and High Entropy Client hints

- Fingerprinting best practice - avoid unnecessary increases to fingerprinting surface
- Browser implements control - limited entropy shared over to server
 - Transparent and permission based models
- Provides a framework better privacy protection
 - Permission policy
 - Avoid abuse usage

The Problem

- Initial requests lack server preferred Client Hints
- Server cannot optimize content on the first visit
- Easy solution: send all but with privacy concern

Solution

- Critical-CH : [draft-victortan-httpbis-chr-critical-ch-00](#)
 - Straightforward
 - Performance regression
- ACCEPT_CH frame: [draft-victortan-httpbis-chr-accept-ch-frame-00](#)
 - draft-bishop-httpbis-extended-settings-01
 - Send the ACCEPT_CH frame as early as possible
 - Improve performance on delivering client hints
 - Connection using TLS: ALPS ([draft-vvv-tls-alps-01](#))
 - Connection-level optimization

Next Steps

- Does the WG think the problem should be resolved?
- Google has an implementation of ACCEPT_CH frame with ALPS, any interests to expand the solution?
- Is there enough interest to adopt the drafts and resume further discussion?