

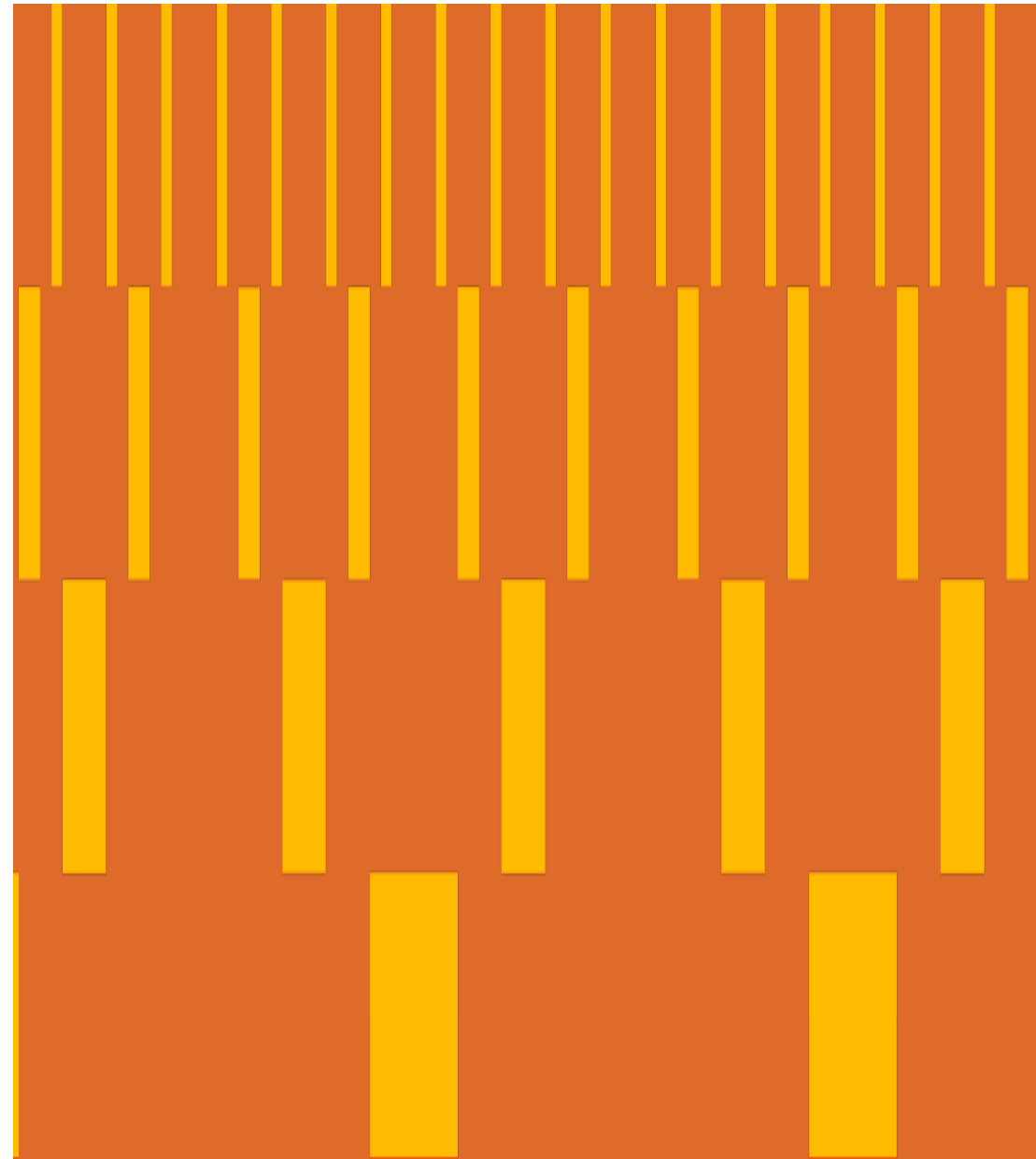
BBC Research & Development

Network metrics/info/hints

Piers O'Hanlon

W3C Web & Networks IG

30 November 2021





Overview: Network metrics

- **Metrics: What are they and where do they come from?**
- **Motivation: Why are they useful?**
- **Server/Edge based**
- **Client based**
- **Privacy/Fingerprinting**
- **Discussion**



Network metrics

- **Transport state/timing information from IP packet transfers**
 - Collected [for use] by congestion/flow control algorithms
- **Provide a view of current network state from sender and/or receiver perspective**
- **Kernel level**
 - TCP Congestion state info (e.g. `TCP_INFO` structure: `cwnd`, `rtt`, `mss`)
 - Delivery Rate Estimation (BBR related: `draft-cheng-iccr-g-delivery-rate-estimation`)
- **Application/Service level**
 - QUIC Transport Info API (e.g. `Mvfst` (Facebook), `Quiche` (Cloudflare), etc)
 - WebRTC/RTP/SCTP
 - Application/Service state



Motivation: Use-Cases

- **Assist or drive the media quality selection algorithms for streaming media**
- **Inform initial [media] rate selection**
- **Provide better bandwidth estimates**
 - Especially for shorter requests (e.g. gRPC, audio) and aggregating/chunked streams as they're harder to measure
- **Control load balancing/scheduling of flows (e.g. in Envoy/Traefik)**
- **Improve latency sensitive applications and reliability of estimates using RTT values**
- **Inform client/browser media/data caching strategies**
- **Traffic analytics and monitoring**
 - Intermediate node information also useful



Server metrics

- **Provide information only known to server/origin/proxy**
 - E.g. sender cwnd, rate estimates, cache
- **Provide information not directly available to client**
 - E.g. rtt,
- **Network transport information accessible on servers/proxies**
 - E.g. Nginx, Apache Traffic Server, Varnish, etc
- **Metrics inserted in HTTP response headers or as data**

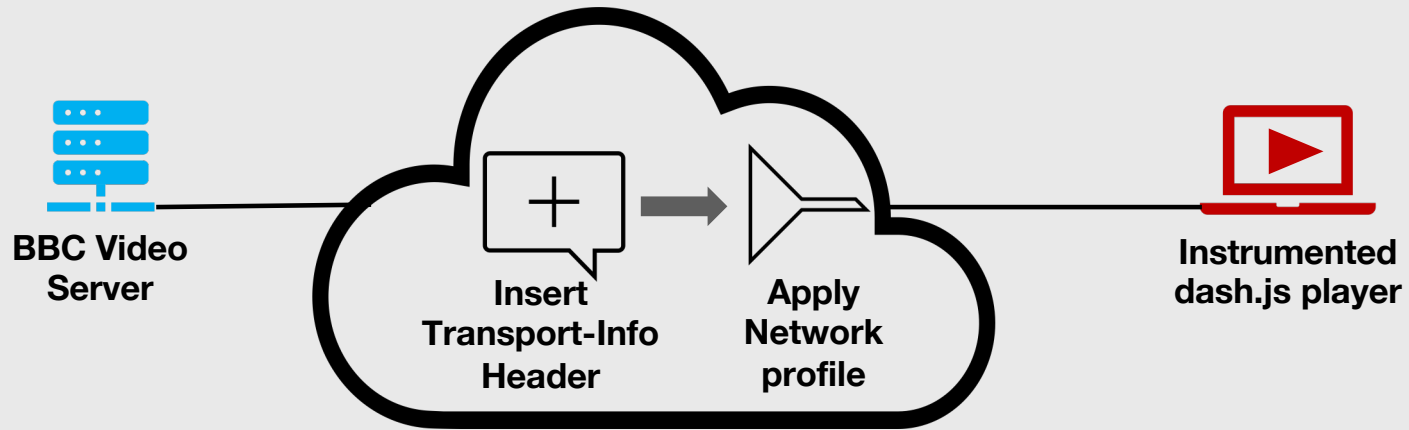
SERVER METRICS

Proposed Schemes

- **IETF**
 - draft-ietf-httpbis-cache-header
 - Cache-Status header: cache metrics/info (e.g. hit/miss, ttl etc)
 - draft-ohanlon-transport-info-header:
 - Transport-Info Header: Network Transport metric delivery (e.g. rtt, cwnd, send_rate)
- **W3C: Server Timing**
 - Timing of request processing
- **CTA-WAVE: Common Media Server Data (CMSD) [CTA-WAVE]**
 - CMSD-Dynamic (e.g. thruput estimates) & Static (e.g. media info)

SERVER METRICS

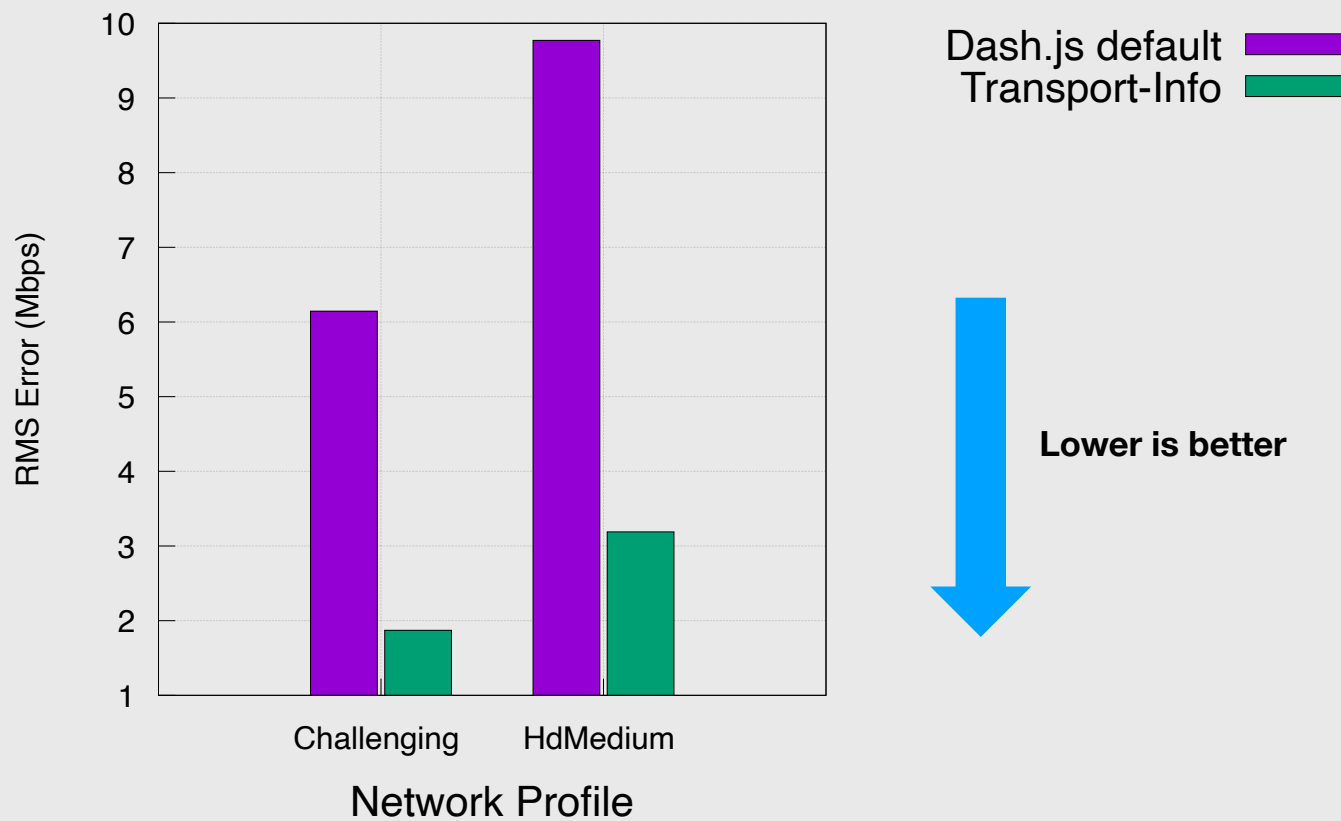
Experimental setup: Testing Transport-Info Header





SERVER METRICS: EXPERIMENTAL RESULTS

Bandwidth estimation error comparison



SERVER METRICS

Issues

- **How to know that client is directly connected**
 - => Some kind of metric validation
- **Coalesced/tunnelled flows**
 - H2/3 allow for different flows to same origin in one transport connection
- **Flow (re)prioritisation**
 - H2/HTTP based priorities
- **Where is the best accuracy-utility trade-off point**
 - Depends on/limits purpose



Client metrics

- Browser/Web APIs

- NetInfo API (e.g. Client side Transport metrics)
 - Rebooted: Proposed just metered and sustainedSpeed
 - But still unpopular with some browsers vendors
 - Very granular estimates
- Resource Timing API
 - Timing of network phases (e.g. DNS, TLS, TTFB, etc)
- WebRTC API – e.g. getStats()
- Others from Web Performance Working Group

- Javascript: Measure and estimate

- Dependent upon CPU resources, increasingly high resolution clocks, and suitable APIs



Addressing Privacy/Fingerprinting issues

- **Data minimisation**
 - Only information required for the purposes of the application be shared.
- **Quantisation and adding noise to metrics [to those deemed sensitive]**
- **Temporal update frequency limits**
- **Explicit opt in**
 - Potentially using HTTP Client Hints (RFC8942)
- **Metrics should only be sent over an encrypted connection**
- **Data already available via other means could be considered less sensitive?**
 - E.g. Information via APIs: NetInfo, Resource Timing, WebRTC



Discussion

- **Potential API improvements**
 - Streams API: Provide data buffer arrival timestamping
 - Improve client based measurements and estimates
 - Resource Timing API: Improve utility for media streaming measurements
- **Where is the right balance between (Where X=Bandwidth, rtt, etc)?**
 - Measuring X client side in browser
 - Accessing local transport X
 - Obtaining a server-side X estimate
- **Further analysis needed on transport metrics benefits/uses**



BBC Research & Development

Q & A