Considerations for HTTP/2 Prioritization

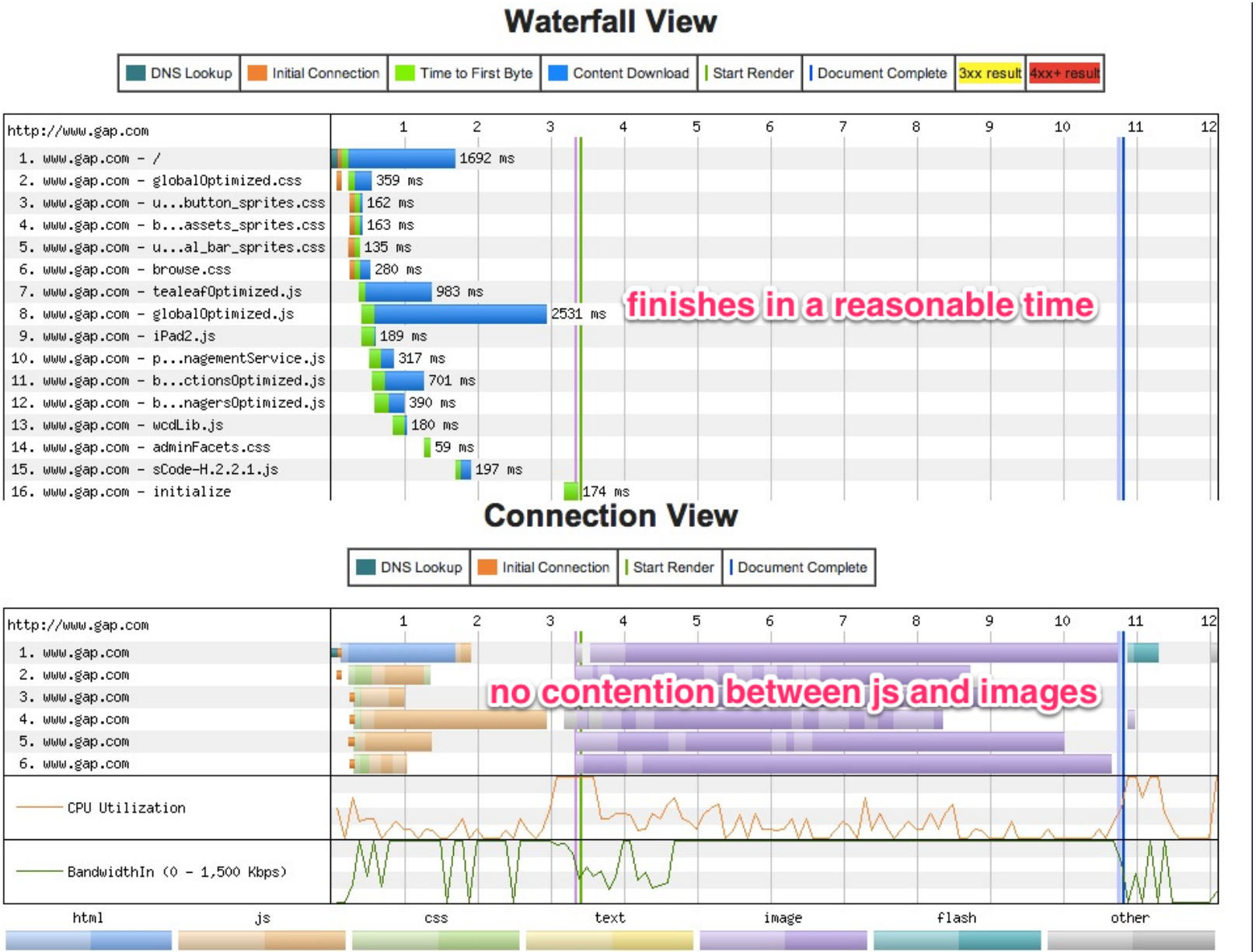
(primarily from a browser perspective)

William Chan
willchan@chromium.org

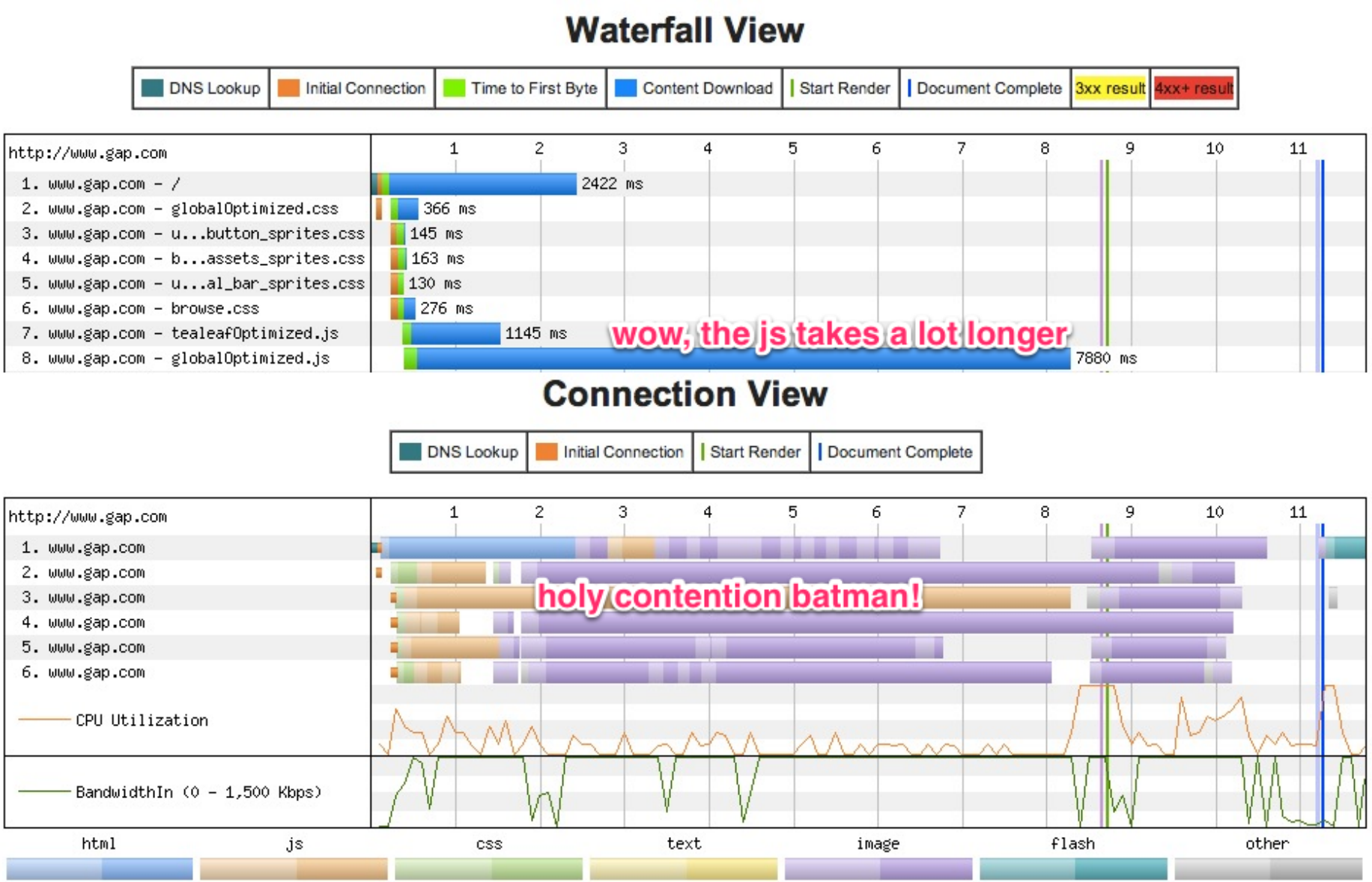
Why is prioritization important?

- Parallelization without prioritization introduces contention
- For a page load, some resources are more important than others
- Browsers are forced to make tradeoffs between link utilitization and contention. For example, most browsers withhold issuing HTTP requests for low priority subresources (e.g. images) until first paint (blocked on stylesheets to prevent FOUC)

Example with Resource Scheduling On



Example with Resource Scheduling Off



Page Load Considerations for Resource Scheduling

- Certain resource types should be loaded earlier than others. Roughly speaking, documents > script/stylesheets > images
- Resources earlier in the document should generally be loaded earlier
- Many resources can only be processed as a whole (scripts, stylesheets, etc). Better to finish a resource before sending another.
- Many resources can be processed in a streaming fashion (documents, images). Sometimes it's good to interleave (e.g. progressive images)

Other Considerations for Stream Prioritization

- Sometimes it's desirable to pipeline. Issue requests for video frames 1, 2, ..., X. Expect that the server responds in order (best effort).
- How to interleave tabs within a browser?
- What happens with the user switches foreground tab? What happens when the user scrolls the viewport of the page?
- How to interleave streams from different user agents at a proxy in a "fair" way?

SPDY/3 Prioritization

- Uses a few bits to assign a static **advisory** priority level to a stream
- Advisory is important. It's better to fill the pipe rather than let it go idle in order to maintain strict ordering
- "The sender and recipient SHOULD use best-effort to process streams in the order of highest priority to lowest priority."
- Does not specify behavior within the same priority level, up to server

Problems with SPDY/3 Prioritization

- No support for re-prioritization
  - Switching foreground tab
  - Rest of resource may not be visible (e.g. a huge document, mostly outside of viewport)
  - User may scroll / change viewport, or rendering engine may discover resources are invisible (e.g. display: none), etc.
- Insufficient mechanism for ordering streams
  - Cannot cleanly handle deep pipelines that exceed # of priorities.
  - Cannot request resource ordering beyond the # of priorities, even if it's better to order than to interleave

Potential solutions

- Introduce a concept of stream "dependencies"
  - Create a dependency graph?
  - Dependency tree where siblings are interleaved?
  - Dependency chain where a bit indicates if an edge is dominant or equal?
  - Have multiple graphs/trees/chains, where the root/head is given a weight for weighted scheduling? Problem: how much of graph/tree/chain must be kept around to maintain dependency info?
- Introduce reprioritization of streams

Potential Problem

- Advisory semantics are unenforceable
  - Since they're advisory, how do you know they're respected?
  - Many existing SPDY/2 and SPDY/3 implementations are known not to use the stream priority field at all.
- If advisory semantics aren't implemented, they cannot be relied upon. It's important for user agent implementions to rely on the semantics and push on servers/intermediaries to support them properly, rather than workaround missing support

Extra Reading Material :)

- https://insouciant.org/tech/resource-prioritization-in-chromium/
- https://insouciant.org/tech/throttling-subresources-before-first-paint/
- https://insouciant.org/tech/prioritization-is-critical-to-spdy/