Corrected timing for low-lead-time prefetches

Notes about bug 337199386[1]

Jeremy Roman <jbroman@chromium.org>

What is changing?

Navigational prefetches initiated using speculation rules (i.e., those which result in a deliveryType of `"navigational-prefetch"`) may see the values of some timestamps change – in particular, the values of timestamps after `fetchStart`, up to and including `responseStart`, may change.

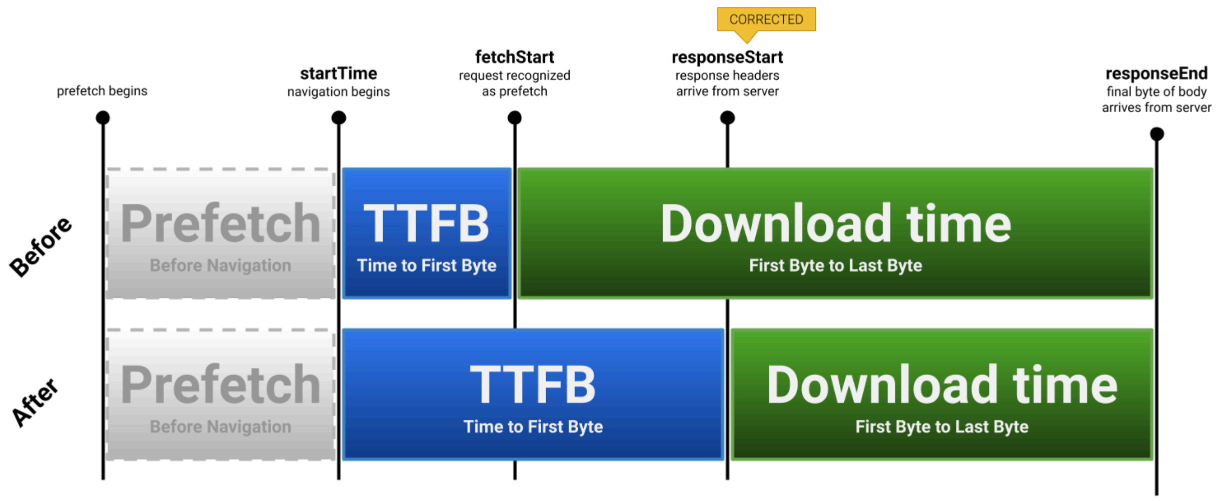
Previously these values took on the value of `fetchStart` for such prefetches; now they will only do so if they occur before `fetchStart`.

For **prefetches which had not yet reached `responseStart` by the time a navigation occurs** (e.g., because it is triggered when the user presses the mouse button and the response headers have not arrived by the time the user releases the mouse button), this will manifest as:

- **an increase in time to first byte (TTFB)**, which was often (incorrectly) very close to zero
- **a corresponding decrease in any loading time metric which measures from `responseStart`** (or, less commonly, an earlier timestamp such as connectEnd)

Metrics which measure from the navigation start or before to an event after the response start, such as largest contentful paint (LCP), should not be affected.

[1] Special thanks to Mateusz Krzeszowiak for an excellent bug report and reproduction case.



For **prefetches which occur long enough in advance of the navigation** for response headers to arrive, significant metrics shifts are not expected.

When is this changing?

This change is expected to be released in **Chrome 126**, and in other Chromium browsers according to their respective release cycles.

What are these timestamps?

These timestamps are part of the Navigation Timing specification, and allow web pages to measure the timing of navigation in detail. The typical flow of these timestamps and the corresponding activity is as follows:

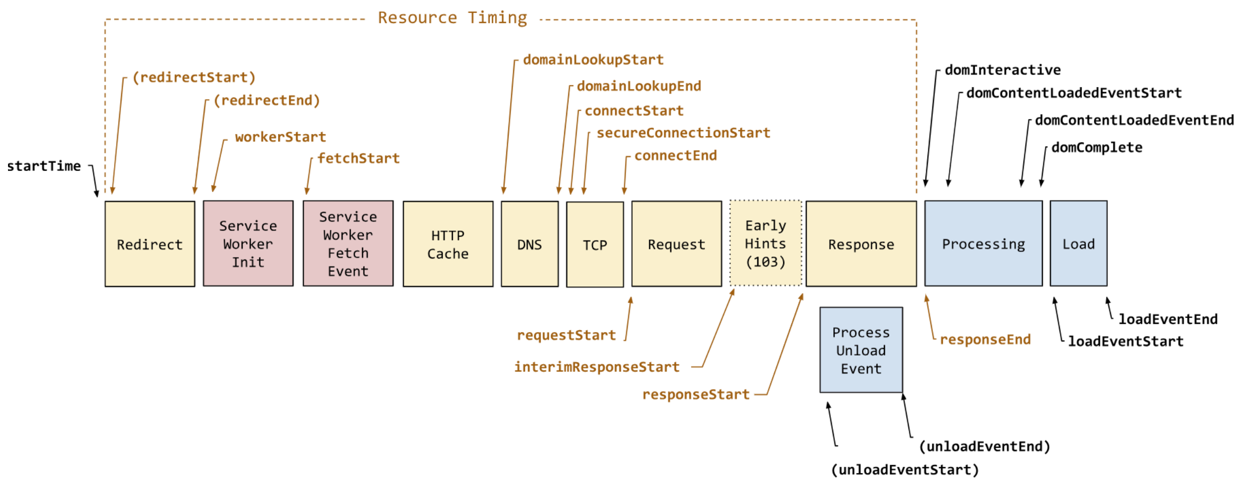


Image from MDN: PerformanceNavigationTiming

These values are often used by performance tooling to compute derived metrics, such as time to first byte (TTFB).

Why did this bug occur?

This section is mostly of interest to Chromium developers.

In early revisions, prefetches could not be served until the response head had arrived, and any navigation before that time would not wait for the prefetch but issue a new request. Consequently, it sufficed to allow these values to duplicate the previous timestamp, since these always logically occur before navigation. See this bug and this doc (Google-internal, sorry).

Since Chromium's PerformanceResourceTiming and PerformanceNavigationTiming implementations do this automatically in the absence of valid timestamps, this was accomplished by preventing load timing information from being included with a prefetched response.

Instead, we should pass along the essential timestamps for computing typical metrics, but simply adjust those that occur before `fetchStart` to occur at the same time as it, as though they had completed instantaneously (since, from the perspective of the user experience, it's as though they had).

Is this fix causing a problem for a site?

This section is mostly of interest to Chromium developers or web developers who suspect this change is affecting their site.

The CL to change this behavior includes a Finch feature parameter intended to replicate the previous behavior, just in case. The following command-line switches toggle between the two behaviors:

- Previous behavior:
  ```
  --enable-features='AdjustNavigationalPrefetchTiming:adjust_navigational_prefetch_timing_behavior/remove_load_timing'
  ```

- New behavior:
  ```
  --enable-features='AdjustNavigationalPrefetchTiming:adjust_navigational_prefetch_timing_behavior/clamp_to_fetch_start'
  ```