

AMP Ads One Hop Fast Fetch for Custom Creative and 3p Ad Networks

Overview

Currently, fast-fetch ads directly deliver ads from an Ad server, such as DFP. For marketplaces and exchanges to support Fast Fetch, we need a mechanism for forwarding the creative content request to a different Ad server on a different domain. Since, historically, repeated passbacks or redirects are a large source of latency and a poor end user experience, we will limit the interaction to a single redirect.

Link to Google Doc:

<https://docs.google.com/document/d/1Nz8DwDjw1tgAAU0yUti39Zv1f6c6p4PYhIRbcXsEKLI/edit#heading=h.tkd5ljsqrb7n>

Reference:

<https://docs.google.com/document/d/1thFF613oMI7OdDddWKGZT7iRZhpYNYSRO7kcEYHJJE/edit#heading=h.mu3ylcl65dnI>

Concerns

- Needs to minimize latency in the end user experience. (One of the original drivers for AMP Ads in the first place).
- Needs to respect amp source origin mechanism, to appropriately attribute the ad request to the original page domain containing the ad.
- Needs to respect the Fast-Fetch signature process.
- Needs to support cookies (tracking/attribution) back to both 3rd parties.

Misconceptions

- **Ultimately 1 hop has a worse user experience than 0 hop. The ad appears late to the user, what about that?**

- This is not always the case and in several cases this can be the opposite. When the ad networks use large edge network the initial html might get to the browser slightly faster with 1 hop (not even guaranteed if ad network servers don't have optimized connection with Google), but the net time for entire ad to load can be slower with 0 hops since DFP doesn't do any optimizations for delivery of the creative. If a user in Germany does 1 hop from DFP and the resulting server it hits is in Germany and sometimes as close as ISP then an edge network can parse the html from ad network as it goes through and the creative assets can be re-written and preloaded onto the edge server and the client can re-use the http2 connection and have them server pushed. DFP could clearly do some of these same optimizations but it isn't really their core business. There are numerous other optimizations that are additive (TLS session resumptions, image optimization, minification, etc). This become more pronounced for video ads and can significantly reduce buffering.

- Additionally doing this server side is still the same number of net calls. It has the advantage of not being made from a mobile device, but has overhead of rendering server side.

- This is also assuming that all the entities would switch to using 0 hop. The overhead and startup costs for networks that typically don't do server to server integrations can be prohibitive if they need to sync cookies. We believe (to be proved in a lab) that the benefits of having more A4A ads in the world far outweighs the preferential treatment benefits of A4A ads just on AMP pages because of the number of AMP pages relative to the web.

- $1 > 0 > 20$

Proposed Implementation

In the amp-a4a codepath, we extend the meaning of extractCreativeAndSignature to include a signal for a redirect. If the redirect property is returned as the creativeParts, it will contain the AMP markup for a single amp-ad. This amp-ad tag will have to match the width/height of the original amp-ad or elide the size parameters.

Example with Macros:[\[a\]](#)[\[b\]](#)[\[c\]](#)

```
<amp-ad width=%%WIDTH%% height=%%HEIGHT%% type=cloudflare data-cf-network=cloudflare src="https://firebolt.cloudflaredemo.com/a4a-ad.html[d][e]"></amp-ad>
```

The a4a code will parse and process the amp-ad tag, potentially loading the second hop's extension and retrieve the redirect URL through the tag's getAdUrl() call. Then it will run one more cycle of fetch/extractCreativeAndSignature to get the final creative and signature to validate the creative. This creative is then processed through the usual A4A verification/replacement mechanisms. Additional macros with impression pixels or click urls can be added to the amp-ad or as a header such as x-ampanalytics[\[f\]](#). The network specific implementations of the redirect could initiate this single-hop redirect through either a custom header, like 'Amp-Fast-Fetch-Redirect', or just by recognition of the content containing a single amp-ad tag. This would require proper CORS headers to be setup by ad server.

We believe this is important step to increase AMP Ad adoption and make for a better Ad experience on AMP. Even if DFP choose not to participate this would still be needed by ad servers such as OpenX, Appnexus, and Adzerk.

Appendix: Considered Alternatives

302 Redirect

Ideally, we would like to leverage existing web standards for redirects (like the HTTP status 302 with Location header), but due to inconsistent implementations of CORS and cookie interactions, this may not be feasible. According to the CORS spec, this should work, and would not require any code changes. (This would involve using a 302 redirect with properly set Access-Control-Allow-Origin and Access-Control-Allow-Credentials response headers), but there is anecdotal evidence that all browsers may not handle this properly. Set-Cookie response headers should also work for both domains. There would be some amount of logic to ensure that the __amp_source_origin parameter is correctly passed to the redirected resource. Additionally, creation of the URL for the next hop becomes the responsibility of the routing server, which prevents leveraging runtime implementations of tags to simplify the URL generation.

Server to Server Chaining with Proxying and Render

Another solution is to force the ad exchange/marketplace to fully proxy calls to ad servers. This has the benefit of potentially reducing the bandwidth (and possibly initial latency for amp ad html) burden from the client device, since the routing of the ad happens completely between servers. The biggest disadvantage is that the target Ad server loses any ability to leverage client cookies to their domain without cookie syncing.

[\[a\]](#) Could you provide an example of what exactly is going into the response from the publisher ad server?

[\[b\]](#) I guess my main question is: Could this go into the JSON that is attached to the creative? I suppose it doesn't matter too much for performance, so maybe this is better.

[\[c\]](#) There is definitely a case for it to be JSON. Especially once ad servers support single requests. The question is also how to make sure Ad Server will want to make sure their click url or other analytics like active view would get inserted. It could happen at multiple levels. IN JS, as it pass through us, at publisher or ad network ad server. .

[\[d\]](#) Can the response contain another redirect?

If no, we need to make a good case for why we support 1-hop but not N-hop.

[\[e\]](#) Maybe we shouldn't think of this hops but overall speed of ads and "cost" on performance. That is really what the user care about.

[\[f\]](#) DFP currently using:

```
x-ampanalytics:"url":"https://pagead2.googlesyndication.com/activeview?
id=ampim\u0026avi=B8B3qwlVSWMOvN4zxBbq7hWAA97WqjOsEAAQAQTgByAEByAMK0ggFCIABEAHCeWYY9Mf70AM\u0026cid=CAASFerOHd7DLi5Oe
```