# Alt-Svc-bis fix+feature candidates

Author: Erik Nygren <erik+ietf@nygren.org>
Date: May 2021
Contributors: Lucas Pardue, Martin Thompson, Mike Bishop (add your name here), ...

# Overview

This brainstorms a set of potential fix+feature candidates for Alt-Svc BIS. In the spirit of brainstorming, some of the things in this list may be bad ideas. One of the points of this document is to judge interest in each of these items to determine what potential authors may be interested in including in a draft.

A number of these items were deferred out of the SVCB/HTTPS draft. Others are based on regrets from implementation experience with AltSvc. A few are potentially valuable features.

It is not yet defined whether this should be a new header or can be done via extensions to Alt-Svc as it exists today.

A number of major clients have yet to implement Alt-Svc to other hostnames fully, and some of this is due to concerns that they have with the current specification.

It may make sense to split this list into batches, but another option would be to try and get all of these done at once even if as separate but cooperating drafts.

# Potential Scope Items

## Incorporating errata and Editorial improvements

(Hopefully this is non-controversial.)

Includes:

- Incorporate errata
- Reference *RFC* 8336 "*ORIGIN Frames"* (regarding disabling connection coalescing)
- ...

## Fix ALPN handling

The ALPN semantics in {{AltSvc}} are ambiguous, and problematic in some interpretations. We should update {{AltSvc}} to give it well-defined semantics that match HTTPS RRs. For example,

specify that the alpn negotiated via the TLS handshake does not need to be the same as the alpn indicated in the AltSvc.

(From https://github.com/MikeBishop/dns-alt-svc/issues/246 and other discussion threads during HTTPS RR. David Benjamin also has strong opinions on this topic.)

One option would be to pull in the text we landed on for the HTTPS/SVCB draft. See 6.1: https://tools.ietf.org/html/draft-ietf-dnsop-svcb-https-02#section-6.1

## Address concerns about Alt-Svc lifetime bounding

Some people have expressed concerns that Alt-Svc allows a compromised Origin to hold onto clients forever by continuing to offer updated Alt-Svc entries. There may be ways to reduce the vulnerability exposure here. This is of particular concern when an Alt-Svc record has a much longer lifetime than an HTTPS RR.

## Support ECH

The HTTPS RR spec puts Alt-Svc out-of-scope. Two options (and there may be more) include:

1. Add an ech= parameter to Alt-Svc
2. Defining some better integration between Alt-Svc and HTTPS RRs that meets a set of requirements. For example, allow an AltSvc server name to be treated as an “AliasMode” reference to an HTTPS record.

## Better interactions with HTTPS record

This was deferred out of the HTTPS RR draft. There are a number of design options here, but requirements and pros/cons will want to be discussed in-detail before proposing designs. Supporting ECH and Alt-Svc together is a primary goal.

An important item here is which takes precedence, providing safe and time-bounded ways to allow Alt-Svc to take precedence over HTTPS records.

Providing a way for Alt-Svc to act as AliasMode references to HTTPS SvcMode records seems like one clean way for interaction in that it avoids needing to duplicate SVCB in Alt-Svc. We would still need to address time-bounding and trust considerations.

## Dont-Be-Clear

This minor issue should get fixed:
https://www.ietf.org/archive/id/draft-pardue-httpbis-dont-be-clear-00.txt

## QUIC frame definition (or deprecate HTTP/2 frame)

See: https://datatracker.ietf.org/doc/html/draft-bishop-httpbis-altsvc-quic-01 and https://datatracker.ietf.org/doc/html/draft-bishop-httpbis-altsvc-quic-01

***Decision point: we could also deprecate the HTTP/2 frame instead.***

## Accept-Alt-Svc request header

There is significant variation in client support for the Alt-Svc specification, including with some clients only implementing a subset of the specification as described elsewhere here. Having an Accept-Alt-Svc request header that lists a set of supported Alt-Svc features allows for extension of Alt-Svc but also allows for deprecation.

If we don't deprecate the frames, we'll also need a SETTINGS equivalent.

There are potential client fingerprinting concerns here so we'll want to not go too far with this.

## Improved/replaced Alt-Used request header

There is limited implementation support for Alt-Used out of privacy concerns. It also only sends a subset of the Alt-Svc record being used, and there are unclear interactions between Alt-Used and HTTPS RRs.

Daniel Stenberg points out:

> *Alt-Used (RFC 7838 section 5) is a request header that only sends host name + port number, with no hint if that port number is TCP or UDP (or ALPN name), which makes at least one large HTTP/3 deployment trigger its Alt-Svc loop detection when only switching protocols to h3.*

It is proposed that we replace or redefine Alt-Used and also define how it interacts with SVCB. Note that hostile origins have many knobs for getting this information (eg, encoding in ports or IPv6 addresses) so a goal would be to allow non-hostile origins to get information on which Alt-Svc or SVCB record is being used in a way that doesn't make things worse from a privacy perspective.

Some options include:

1. Just send the whole Alt-Svc or SVCB binding used in the header

2. Have a param encoding an N-bit or N-character value for the record-id. This value would be sent as Alt-Used. How many bits to use is an open question.
    - Allowing for a dynamic length where the client chooses how many bits or characters to include based on privacy budget is one attractive but complicated option. Server implementers would put the most important info into earlier bits/characters.

The goal here is to allow for virtual hosting of alternative services, allowing the server to know which alternative service was used (eg, for load feedback, diagnostics/debugging, loop detection, and other operational purposes), but without hacks like separate ports or IP addresses that leak information to passive network observers.

See some discussion and other options here:
https://github.com/MikeBishop/dns-alt-svc/issues/107

One corner-case is that SNI-based cluster load balancing (fairly common) can't see any Alt-Svc/Alt-Used information but may need it for making decisions. We want to keep this out of IP addresses and port numbers and anything else in-the-clear where possible. One idea is that if ECH is in-use that we could include Alt-Used information in the inner (encrypted) ClientHello in a new TLS extension. This would of course require ECH and Alt-Svc to work well together, but might be a motivator.

## Path/object scoped Alt-Svc

The largest-scope, most disruptive, and perhaps most controversial item would be to allow Alt-Svc to be scoped to URL paths with a way to indicate that transitions to use the Alt-Svc should be done synchronously.

This is desired for use-cases of large content libraries where an Origin would like to have clients use different server names for different objects but while sharing a single Origin. This would also likely need negotiation.

For example:

- Client indicates via a request header (eg, Accept-*) or a SETTING that it supports this feature
- Server could indicate that the path="/movies/MurderOnTheExampleExpress/" should be accessed by a particular alternative service and that the client should synchronously switch to use it. Server returns a new 3xx response header along with the Alt-Svc.
- On connecting to the indicated alternative service, the client sends an Alt-Used (or alt-used-bis) header to indicate that it is now using this alternative service.

Example:

```
To www.example.com:
```

```
            GET /foo/bar.mp4
            Accept-Alt-Svc: sync, path, altused2
            Host: www.example.com

From www.example.com:

            HTTP/1.1 3xx Alternative Server
            Alt-Svc: h2="foo.example.net:443"; ma=7200; sync; path="/foo/"

To foo.example.net:
            GET /foo/bar.mp4
            Accept-Alt-Svc: path, altused2
            Host: www.example.com
            Alt-Used-v2: alt-svc;h2="foo.example.net:443";path

From foo.example.net:
            HTTP/1.1 200 OK
            Alt-Svc: h2="foo.example.net:443"; refresh; ma=7200; path="/foo/"
            [Content-Body]
```

## Persist & other caching concerns

RFC 7838 defines the persist parameter.

> Alternative services that are intended to be longer lived (such as those that are not specific to the client access network) can carry the "persist" parameter with a value "1" as a hint that the service is potentially useful beyond a network configuration change.
>
> When alternative services are used to send a client to the most optimal server, a change in network configuration can result in cached values becoming suboptimal. Therefore, clients SHOULD remove from cache all alternative services that lack the "persist" flag with the value "1" when they detect such a change, when information about network state is available.

We talked about this somewhere (maybe the QUIC slack?). For clients like curl detecting network changes is very tricky. For clients like Chrome, IIRC they default to behaving like persist=1 for all alternatives.

The RFC text today seems to imply that server's factor in client network properties when deciding what to advertise. I'm not sure that is really true for the majority of deployments - it isn't for Cloudflare's H3 adverts. IMO the recommendation that a client invalidate Alt-Svc cache entries based on their own network state changes seems mistaken today, and the situation can

potentially get worse with protocol evolution (connection migration, multipath etc). Clients should aggresively manage their Alt-Svc caches, but it seems wiser to base this on active probing, or other heuristics not just network changes.

## Radical simplification

***Discussion/Decision Point:*** If we are able to reach wide deployment and use of the HTTPS record, it may supersede many use cases for Alt-Svc. We should reassess the needs from the new baseline to see whether Alt-Svc can be radically simplified. (Chrome never fully implemented Alt-Svc redirection anyway.)