Alt-Svc SNI
and
DNS ALT SVC

Mike Bishop and Ben Schwartz
HTTPBIS, IETF 102
Alt-Svc SNI in a nutshell

Alt-Svc:
```
h2="www.example.com:443";ma=2635200;persist=true;sni=www.example.com
```

“For queries to this domain in the next 30 days, on any network, open a socket to www.example.com:443, ask for www.example.com, and use HTTP/2.”

- Alt-Svc is used for protocol upgrades and load balancing
- Currently (without Alt-Svc SNI), the client maintains the original SNI when contacting the new server
- If the first certificate doesn’t match the original host, use Secondary Certificate Authentication to get the right certificate
Certificate Validation Rules (new and improved!)

Goal: Maximize flexibility while ensuring defense against an active adversary

The server MUST return a valid certificate which covers at least one of the following:

- The hostname indicated in the SNI extension
- The hostname of the origin that published the alternative
- The hostname used for connecting to the alternative

The client MUST validate the certificate in the handshake for authenticity according to [RFC2818] and ensure that it is valid for at least one of these names. Clients SHOULD NOT accept certificates issued to the IP address of the alternative unless the alternative is specified as an IP literal.
DNS ALTSVC in a nutshell

_443._https.www.example.com. 30D IN ALTSVC
"h2=":443";persist=true;sni=innocence.example"

“When connecting to https://www.example.com:443 in the next 30 days, use this value for Alt-Svc.”

- Alt-Svc value is human-readable but opaque to the DNS
- No change to Alt-Svc syntax or semantics
- Waiting for an ALTSVC response is optional
- Alt-host can be an IP address, empty (no change), or a name (needs lookup)
- **New**: Multiple RRs for load-balancing, multiple hosts per RR for fallback
<table>
<thead>
<tr>
<th>Alt-Svc SNI + DNS ALTSVC</th>
<th>ESNI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can accelerate Alt-Svc</td>
<td>Never improves performance (currently)</td>
</tr>
<tr>
<td>Adds a roundtrip when the alt-host is not in the initial certificate</td>
<td>Only adds a roundtrip in exception cases</td>
</tr>
<tr>
<td>Still helps without DNS, using in-band Alt-Svc</td>
<td>Depends on non-address DNS every TTL</td>
</tr>
<tr>
<td>Enables Opportunistic Encryption from the start</td>
<td>Has no effect on plain-text HTTP</td>
</tr>
<tr>
<td>Only for HTTP and HTTPS (so far)</td>
<td>Naturally applies to any use of TLS</td>
</tr>
<tr>
<td>No change to wire image</td>
<td>The TLS extension is publicly visible</td>
</tr>
<tr>
<td>DNS lookups are rare, responses are small</td>
<td>ESNI values are large and have short TTL</td>
</tr>
<tr>
<td>DNS entries are human-readable and should not require frequent maintenance</td>
<td>DNS entries are opaque and must be updated periodically by the TLS frontend</td>
</tr>
<tr>
<td>Each frontend can be configured independently (e.g. multi-CDN)</td>
<td>All frontends must hold the same ESNI private key</td>
</tr>
<tr>
<td>Enables new load balancing and DDOS defenses</td>
<td>Strictly a privacy measure</td>
</tr>
</tbody>
</table>