DNSOP Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

B. Schwartz
Google
M. Bishop
E. Nygren
Akamai Technologies

November 4, 2019

Service binding and parameter specification via the DNS (DNS SVCB and HTTPSSVC)
draft-ietf-dnsop-svcb-httpssvc-01

Abstract

This document specifies the "SVCB" and "HTTPSSVC" DNS resource record types to facilitate the lookup of information needed to make connections for origin resources, such as for HTTPS URLs. SVCB records allow an origin to be served from multiple network locations, each with associated parameters (such as transport protocol configuration and keying material for encrypting TLS SNI). They also enable aliasing of apex domains, which is not possible with CNAME. The HTTPSSVC DNS RR is a variation of SVCB for HTTPS and HTTP origins. By providing more information to the client before it attempts to establish a connection, these records offer potential benefits to both performance and privacy.

TO BE REMOVED: This proposal is inspired by and based on recent DNS usage proposals such as ALTSVC, ANAME, and ESNIKEYS (as well as long standing desires to have SRV or a functional equivalent implemented for HTTP). These proposals each provide an important function but are potentially incompatible with each other, such as when an origin is load-balanced across multiple hosting providers (multi-CDN). Furthermore, these each add potential cases for adding additional record lookups in-addition to AAAA/A lookups. This design attempts to provide a unified framework that encompasses the key functionality of these proposals, as well as providing some extensibility for addressing similar future challenges.

TO BE REMOVED: The specific name for this RR type is an open topic for discussion. "SVCB" and "HTTPSSVC" are meant as placeholders as they are easy to replace. Other names might include "B", "SRV2", "SVCHTTPS", "HTTPS", and "ALTSVC".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction

The SVCB and HTTPSSVC RRs provide clients with complete instructions for access to an origin. This information enables improved performance and privacy by avoiding transient connections to a sub-optimal default server, negotiating a preferred protocol, and providing relevant public keys.

For example, when clients need to make a connection to fetch resources associated with an HTTPS URI, they currently resolve only A and/or AAAA records for the origin hostname. This is adequate for services that use basic HTTPS (fixed port, no QUIC, no [ESNI]). Going beyond basic HTTPS confers privacy, performance, and operational advantages, but it requires the client to learn additional information, and it is highly desirable to minimize the number of round-trip and lookups required to learn this additional information.

The SVCB and HTTPSSVC RRs also help when the operator of an origin wishes to delegate operational control to one or more other domains, e.g. delegating the origin resource "https://example.com" to a service operator endpoint at "svc.example.net". While this case can sometimes be handled by a CNAME, that does not cover all use-cases. CNAME is also inadequate when the service operator needs to provide a bound collection of consistent configuration parameters through the DNS (such as network location, protocol, and keying information).

This document first describes the SVCB RR as a general-purpose resource record that can be applied directly and efficiently to a wide range of services. As HTTPS is a primary use-case and has special requirements, the HTTPSSVC RR is also defined within this document as a special case of SVCB. Services wishing to avoid the need for an [Attrleaf] label with SVCB may follow the pattern of HTTPSSVC and assign their own SVCB-compatible RR types.

All behaviors described as applying to the SVCB RR also apply to the HTTPSSVC RR unless explicitly stated otherwise. Section 7 describes additional behaviors specific to the HTTPSSVC record. Apart from Section 7 and introductory examples, much of this document refers only to the SVCB RR, but those references should be taken to apply to SVCB, HTTPSSVC, and any future SVCB-compatible RR types.

The SVCB RR has two forms: 1) the "Alias Form" simply delegates operational control for a resource; 2) the "Service Form" binds together configuration information for a service endpoint. The Service Form provides additional key=value parameters within each RDATA set.

TO BE REMOVED: If we use this for providing configuration for DNS authorities, it is likely we'd specify a distinct "NS2" RR type that is an instantiation of SVCB for authoritative nameserver delegation and parameter specification, similar to HTTPSSVC.

TO BE REMOVED: Another open question is whether SVCB records should be self-descriptive and include the service name (eg, "https") in the RDATA section to avoid ambiguity. Perhaps this could be included as a svc="baz" parameter for protocols that are not the default for the RR type? Current inclination is to not do so.

1.1. Introductory Example

As an introductory example for an HTTPS origin resource, a set of example HTTPSSVC and associated A+AAAA records might be:

```
www.example.com.  7200  IN CNAME    svc.example.net.
; AliasForm
example.com.      7200  IN HTTPSSVC 0 svc.example.net.
; ServiceForm
svc.example.net.  7200  IN HTTPSSVC 2 svc3.example.net. ( alpn=h3
                                      port=8003 esniconfig="..." )
svc.example.net.  7200  IN HTTPSSVC 3 svc2.example.net. ( alpn=h2
                                      port=8002 esniconfig="..." )
svc2.example.net. 300   IN A        192.0.2.2
svc2.example.net. 300   IN AAAA     2001:db8::2
svc3.example.net. 300   IN A        192.0.2.3
svc3.example.net. 300   IN AAAA     2001:db8::3
; Compatibility records for non-HTTPSSVC-aware clients
example.com.      300   IN A        192.0.2.1
example.com.      300   IN AAAA     2001:db8::1
svc.example.net.  300   IN A        192.0.2.1
svc.example.net.  300   IN AAAA     2001:db8::1
```

In the preceding example, both of the "example.com" and "www.example.com" origin names are aliased to use alternative service endpoints offered as "svc.example.net" (with "www.example.com" continuing to use a CNAME alias). HTTP/2 is available on a cluster of machines located at svc2.example.net with TCP port 8002 and HTTP/3 is available on a cluster of machines located at svc3.example.net with UDP port 8003. The client can use the specified ESNI keys to encrypt the SNI values of "example.com" and "www.example.com" in the handshake with these alternative service endpoints. When connecting, clients will continue to treat the authoritative origins as "https://example.com" and "https://www.example.com", respectively.

For services other than HTTPS (as well as for HTTPS origins with non-default ports), the SVCB RR and an [Attrleaf] label will be used.

For example, to reach an example resource of "baz://api.example.com:8765", the following Alias Form SVCB record would be used to delegate to "svc4-baz.example.net." which in-turn could return AAAA/A records and/or SVCB records in ServiceForm.

```
_8765._baz.api.example.com. 7200 IN SVCB 0 svc4-baz.example.net.
```

1.2. Goals of the SVCB RR

The goal of the SVCB RR is to allow clients to resolve a single additional DNS RR in a way that:

- Provides service endpoints authoritative for the service, along with parameters associated with each of these endpoints.
- Does not assume that all alternative service endpoints have the same parameters or capabilities, or are even operated by the same entity. This is important as DNS does not provide any way to tie together multiple RRs for the same name. For example, if www.example.com is a CNAME alias that switches between one of three CDNs or hosting environments, successive queries for that name may return records that correspond to different environments.
- Enables CNAME-like functionality at a zone apex (such as "example.com") for participating protocols, and generally enables delegation of operational authority for an origin within the DNS to an alternate name.

Additional goals specific to HTTPSSVC and the HTTPS use-case include:

- Connect directly to [HTTP3] (QUIC transport) alternative service endpoints
- Obtain the [ESNI] keys associated with an alternative service endpoint
- Support non-default TCP and UDP ports
- Address a set of long-standing issues due to HTTP(S) clients not implementing support for SRV records, as well as due to a limitation that a DNS name can not have both CNAME and NS RRs (as is the case for zone apex names)
- Provide an HSTS-like indication signaling for the duration of the DNS RR TTL that the HTTPS scheme should be used instead of HTTP (see Section 7.4).

1.3. Overview of the SVCB RR

This subsection briefly describes the SVCB RR in a non-normative manner. (As mentioned above, this all applies equally to the HTTPSSVC RR which shares the same encoding, format, and high-level semantics.)

The SVCB RR has two forms: AliasForm and ServiceForm. SVCB RR entries with two non-empty fields are in AliasForm. When more fields are present, this indicates that the SVCB RR is in ServiceForm. The fields are:

1. SvcFieldPriority: The priority of this record (relative to others, with lower values preferred). Applicable for the ServiceForm, and otherwise has value "0". (Described in Appendix A.1.)

2. SvcDomainName: The domain name of either the alias target (for AliasForm) or the alternative service endpoint (for ServiceForm).

3. SvcFieldValue: A list of key=value pairs describing the alternative service endpoint for the domain name specified in SvcDomainName (only for ServiceForm and otherwise empty). Described in Section 2.1.

Cooperating DNS recursive resolvers will perform subsequent record resolution (for SVCB, A, and AAAA records) and return them in the Additional Section of the response. Clients must either use responses included in the additional section returned by the recursive resolver or perform necessary SVCB, A, and AAAA record resolutions. DNS authoritative servers may attach in-bailiwick SVCB, A, AAAA, and CNAME records in the Additional Section to responses for an SVCB query.

When in the ServiceForm, the SvcFieldValue of the SVCB RR provides an extensible data model for describing network endpoints that are authoritative for the origin, along with parameters associated with each of these endpoints.

For the HTTPS use-case, the HTTPSSVC RR enables many of the benefits of [AltSvc], without waiting for a full HTTP connection initiation (multiple roundtrips) before learning of the preferred alternative, and without necessarily revealing the user's intended destination to all entities along the network path.

1.4. Parameter for ESNI

This document also defines a parameter for Encrypted SNI [ESNI] keys, both as a general SVCB parameter and also as a corresponding Alt-Svc parameter. See Section 8.

1.5. Terminology

For consistency with [AltSvc], we adopt the following definitions:

- An "origin" is an information source as in [RFC6454]. For services other than HTTPS, the exact definition will need to be provided by the document mapping that service onto the SVCB RR.

- The "origin server" is the server that the client would reach when accessing the origin in the absence of the SVCB record or an HTTPS Alt-Svc.

- An "alternative service" is a different server that can serve the origin over a specified protocol.

For example within HTTPS, the origin consists of a scheme (typically "https"), a host name, and a port (typically "443").

Additional DNS terminology intends to be consistent with [DNSTerm].

SVCB is a contraction of "service binding". HTTPSSVC is a contraction of "HTTPS service". SVCB, HTTPSSVC, and future RR types that share SVCB's format and registry are collectively known as SVCB-compatible RR types.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The SVCB record type

The SVCB DNS resource record (RR) type (RR type ???) is used to locate endpoints that can service an origin. There is special handling for the case of "https" origins. The presentation format of the record is:

Name TTL IN SVCB SvcFieldPriority SvcDomainName SvcFieldValue

The SVCB record is defined specifically within the Internet ("IN") Class ([RFC1035]). SvcFieldPriority is a number in the range

0-65535, SvcDomainName is a domain name, and SvcFieldValue is a set of key=value pairs present for the ServiceForm. The SvcFieldValue is empty for the AliasForm.

The algorithm for resolving SVCB records and associated address records is specified in Section 3.

2.1. Parameter specification via ServiceFieldValue

In ServiceForm, the SvcFieldValue contains key=value pairs. Keys are IANA-registered SvcParamKeys (Section 11.1) with both a case-insensitive string representation and a numeric representation in the range 0-65535. Registered key names should only contain characters from the ranges "a"-"z", "0"-"9", and "-". In ABNF [RFC5234],

```
ALPHA_LC    = %x61-7A   ;  a-z
key         = ALPHA_LC / DIGIT / "-"
display-key = ALPHA / DIGIT / "-"
```

Values are in a format specific to the SvcParamKey. Their definition should specify both their presentation format and wire encoding (e.g., domain names, binary data, or numeric values).

The SVCB format preserves the order of values and can encode multiple values for the same parameter. However, clients MUST consider only the first appearance of a parameter unless its specification explicitly allows multiple values.

2.1.1. Presentation format

The presentation format for SvcFieldValue is a whitespace-separated list of the key=value pairs. Each pair is presented in the following form:

```
; basic-visible is VCHAR minus DQUOTE, ";", and "\"
basic-visible = %x21 / %x23-3A / %x3C-5B / %x5D-7E
escaped-char  = "\" (VCHAR / WSP)
contiguous    = *(basic-visible / escaped-char)
quoted-string = DQUOTE *(contiguous / WSP) DQUOTE
value         = quoted-string / contiguous
pair          = display-key "=" value
```

The value format is intended to match the definition of <character-string> in [RFC1035] Section 5.1. (Unlike <character-string>, the length of a value is not limited to 255 characters.)

Unrecognized keys are represented in presentation format as "keyNNNNN" where NNNNN is the numeric value of the key type without

leading zeros. In presentation format, values of unrecognized keys should be represented in wire format, using decimal escape codes (e.g. \255) when necessary.

2.2. SVCB RDATA Wire Format

The RDATA for the SVCB RR consists of:

- a 2 octet field for SvcFieldPriority as an integer in network byte order.
- the uncompressed SvcDomainName, represented as a sequence of length-prefixed labels as in Section 3.1 of [RFC1035].
- the SvcFieldValue byte string, consuming the remainder of the record (so smaller than 65535 octets and constrained by the RDATA and DNS message sizes).

AliasForm is defined by SvcFieldPriority being 0.

When SvcFieldValue is non-empty (ServiceForm), it contains a list of SvcParamKey=SvcParamValue pairs with length-prefixes for the SvcParamValues, each of which contains:

- a 2 octet field containing the SvcParamKey as an integer in network byte order.
- a 2 octet field containing the length of the SvcParamValue as an integer between 0 and 65535 in network byte order (but constrained by the RDATA and DNS message sizes).
- an octet string of the length defined by the previous field.

If the parser reaches the end of the RDATA while parsing a SvcFieldValue, the RR is invalid and MUST be discarded.

TODO: decide if we want special handling for any SvcParamKey ranges? For example: range for greasing; experimental range; range-of-mandatory-to-use-the-RR vs range of ignore-just-param-if-unknown.

2.3. SVCB owner names

When querying the SVCB RR, an origin is typically translated into a QNAME by prefixing the port and scheme with "_", then concatenating them with the host name, resulting in a domain name like "_8004._examplescheme.api.example.com.".

Protocol mappings for SVCB MAY remove the port or replace it with other protocol-specific information, but MUST retain the scheme in the QNAME. RR types other than SVCB can define additional behavior for translating origins to QNAMEs. See Section 7.1 for the HTTPSSVC behavior.

When a prior CNAME or SVCB record has aliased to an SVCB record, each RR shall be returned under its own owner name.

Note that none of these forms alter the origin or authority for validation purposes. For example, clients MUST continue to validate TLS certificate hostnames based on the origin host.

As an example:

```
_8443._foo.api.example.com. 7200 IN SVCB 0 svc4.example.net.
svc4.example.net.  7200  IN SVCB 3 ( svc4.example.net. alpn="bar"
                                    port="8004" esniconfig="..." )
```

would indicate that "foo://api.example.com:8443" is aliased to use ALPN protocol "bar" service endpoints offered at "svc4.example.net" on port 8004.

2.4. SvcRecordType

The SvcRecordType is indicated by the SvcFieldPriority, and defines the form of the SVCB RR. When SvcFieldPriority is 0, the SVCB SvcRecordType is defined to be in AliasForm. Otherwise, the SVCB SvcRecordType is defined to be in ServiceForm.

Within an SVCB RRSet, all RRs should have the same SvcRecordType. If an RRSet contains a record in AliasForm, the client MUST ignore any records in the set with ServiceForm.

2.5. SVCB records: AliasForm

When SvcRecordType is AliasForm, the SVCB record is to be treated similar to a CNAME alias pointing to SvcDomainName. SVCB RRSets SHOULD only have a single resource record in this form. If multiple are present, clients or recursive resolvers SHOULD pick one at random.

The AliasForm's primary purpose is to allow aliasing at the zone apex, where CNAME is not allowed. For example, if an operator of https://example.com wanted to point HTTPS requests to a service operating at svc.example.net, they would publish a record such as:

```
example.com. 3600 IN SVCB 0 svc.example.net.
```

The SvcDomainName MUST point to a domain name that contains another SVCB record, address (AAAA and/or A) records, or both address records and a ServiceForm SVCB record.

Note that the SVCB record's owner name MAY be the canonical name of a CNAME record, and the SvcDomainName MAY be the owner of a CNAME record. Clients and recursive resolvers MUST follow CNAMEs as normal.

Due to the risk of loops, clients and recursive resolvers MUST implement loop detection. Chains of consecutive SVCB and CNAME records SHOULD be limited to (8?) prior to reaching terminal address records.

As legacy clients will not know to use this record, service operators will likely need to retain fallback AAAA and A records alongside this SVCB record, although in a common case the target of the SVCB record might offer better performance, and therefore would be preferable for clients implementing this specification to use.

Note that SVCB AliasForm RRs do not alias to RR types other than address records (AAAA and A), CNAMEs, and ServiceForm SVCB records. For example, an AliasForm SVCB record does not alias to an HTTPSSVC record, nor vice-versa.

2.6. SVCB records: ServiceForm

When SvcRecordType is the ServiceForm, the combination of SvcDomainName and SvcFieldValue parameters within each resource record associates an alternative service location with its connection parameters.

Each protocol scheme that uses SVCB MUST define a protocol mapping that explains how SvcFieldValues are applied for connections of that scheme. Appendix A defines a limited mapping between Alt-Svc ([AltSvc]) values and the SVCB ServiceForm. Protocols using SVCB may use this Alt-Svc mapping if they also use Alt-Svc. Unless specified otherwise by the protocol mapping, clients MUST ignore SvcFieldValue parameters that they do not recognize.

2.6.1. Special handling of "." for SvcDomainName in ServiceForm

For ServiceForm SVCB RRs, if SvcDomainName has the value ".", then the owner name of this record MUST be used as the effective SvcDomainName. (The SvcDomainName of an SVCB RR in AliasForm MUST NOT have this value.)

For example, in the following example "svc2.example.net" is the effective SvcDomainName:

```
www.example.com.  7200  IN HTTPSSVC svc.example.net.
svc.example.net.  7200  IN CNAME    svc2.example.net.
svc2.example.net. 7200  IN HTTPSSVC 1 . ( alpn=h2
                                      port=8002 esniconfig="..." )
svc2.example.net. 300   IN A        192.0.2.2
svc2.example.net. 300   IN AAAA     2001:db8::2
```

2.6.2. SvcFieldPriority

As RRs within an RRSet are explicitly unordered collections, the SvcFieldPriority value serves to indicate priority. SVCB RRs with a smaller SvcFieldPriority value SHOULD be given preference over RRs with a larger SvcFieldPriority value.

When receiving an RRSet containing multiple SVCB records with the same SvcFieldPriority value, clients SHOULD apply a random shuffle within a priority level to the records before using them, to ensure uniform load-balancing.

3. Client behavior

An SVCB-aware client resolves an origin HOST by attempting to determine the preferred SvcFieldValue and IP addresses for its service, using the following procedure:

1. Issue parallel AAAA/A and SVCB queries for the name HOST. The answers for these may or may not include CNAME pointers before reaching one or more of these records.

2. If an SVCB record of AliasForm SvcRecordType is returned for HOST, clients MUST loop back to step 1 replacing HOST with SvcDomainName, subject to loop detection heuristics.

3. If one or more SVCB records of ServiceForm SvcRecordType are returned for HOST, clients should select the highest-priority option with acceptable parameters, and resolve AAAA and/or A records for its SvcDomainName if they are not already available. These are the preferred SvcFieldValue and IP addresses. If the connection fails, the client MAY try to connect using values from a lower-priority record. If none of the options succeed, the client SHOULD connect to the origin server directly.

4. If an SVCB record for HOST does not exist, the received AAAA and/or A records are the preferred IP addresses and there is no SvcFieldValue.

This procedure does not rely on any recursive or authoritative server to comply with this specification or have any awareness of SVCB.

When selecting between AAAA and A records to use, clients may use an approach such as [HappyEyeballsV2].

Some important optimizations are discussed in Section 5 to avoid additional latency in comparison to ordinary AAAA/A lookups.

3.1. Clients using a Proxy

Clients using a domain-oriented transport proxy like HTTP CONNECT ([RFC7231] Section 4.3.6) or SOCKS5 ([RFC1928]) SHOULD disable SVCB support if performing SVCB queries would violate the client's privacy intent.

If the client can safely perform SVCB queries (e.g. via the proxy or an affiliated resolver), the client SHOULD follow the standard SVCB resolution process, selecting the highest priority option that is compatible with the client and the proxy. The client SHOULD provide the final SvcDomainName and port (if present) to the proxy as the destination host and port.

Providing the proxy with the final SvcDomainName has several benefits:

- It allows the client to use the SvcFieldValue, if present, which is only usable with a specific SvcDomainName. The SvcFieldValue may include information that enhances performance (e.g. alpn) and privacy (e.g. esniconfig).

- It allows the origin to delegate the apex domain.

- It allows the proxy to select between IPv4 and IPv6 addresses for the server according to its configuration, and receive addresses based on its network geolocation.

4. DNS Server Behavior

When replying to an SVCB query, authoritative DNS servers SHOULD return A, AAAA, and SVCB records (as well as any relevant CNAME records) in the Additional Section for any in-bailiwick SvcDomainNames.

Recursive resolvers that are aware of SVCB SHOULD ensure that the client can execute the procedure in Section 3 without issuing a second round of queries, by following this procedure while constructing a response to a stub resolver for an SVCB record query:

1. When processing an SVCB response from an authoritative server, add it to the Additional section (unless it is the Answer).

2. If all records are in ServiceForm, resolve A and AAAA records for each SvcDomainName (or for the owner name if the SvcDomainName is "."), and include all the results in the Additional section.

3. Otherwise, select an AliasForm record at random, and resolve A, AAAA, and SVCB records for the SvcDomainName. If the SVCB record does not exist, add the A and AAAA records to the Additional section. Otherwise, go to step 1, subject to loop detection heuristics.

All DNS servers SHOULD treat the SvcParam portion of the SVCB RR as opaque and SHOULD NOT try to alter their behavior based on its contents.

When responding to a query that includes the DNSSEC OK bit ([RFC3225]), DNSSEC-capable recursive and authoritative DNS servers MUST accompany each RRSet in the Additional section with the same DNSSEC-related records that it would send when providing that RRSet as an Answer.

5. Performance optimizations

For optimal performance (i.e. minimum connection setup time), clients SHOULD issue address (AAAA and/or A) and SVCB queries simultaneously, and SHOULD implement a client-side DNS cache. Responses in the Additional section of an SVCB response SHOULD be placed in cache before performing any followup queries. With these optimizations in place, and conforming DNS servers, using SVCB does not add network latency to connection setup.

5.1. Optimistic pre-connection and connection reuse

If an address response arrives before the corresponding SVCB response, the client MAY initiate a connection as if the SVCB query returned NODATA, but MUST NOT transmit any information that could be altered by the SVCB response until it arrives. For example, a TLS ClientHello can be altered by the "esniconfig" value of an SVCB response (Section 6.3). Clients implementing this optimization SHOULD wait for 50 milliseconds before starting optimistic pre-connection, as per the guidance in [HappyEyeballsV2].

An SVCB record is consistent with a connection if the client would attempt an equivalent connection when making use of that record. If an SVCB record is consistent with an active or in-progress connection C, the client MAY prefer that record and use C as its connection.

For example, suppose the client receives this SVCB RRSet for a protocol that uses TLS over TCP:

```
_1234._bar.example.com. 300 IN SVCB 1 svc1.example.net (
    esniconfig="111..." ipv6hint=2001:db8::1 port=1234 ... )
                               SVCB 2 svc2.example.net (
    esniconfig="222..." ipv6hint=2001:db8::2 port=1234 ... )
```

If the client has an in-progress TCP connection to "[2001:db8::2]:1234", it MAY proceed with TLS on that connection using "esniconfig="222..."", even though the other record in the RRSet has higher priority.

If none of the SVCB records are consistent with any active or in-progress connection, clients must proceed as described in Step 3 of the procedure in Section 3.

5.2. Preferring usable records

A nonconforming recursive resolver might not return all the information required to use all the records in an SVCB response. If some of the SVCB records in the response can be used without requiring additional DNS queries, the client MAY prefer those records, regardless of their priorities.

5.3. Structuring zones for performance

To avoid a delay for clients using a nonconforming recursive resolver, domain owners SHOULD use a single SVCB record whose SvcDomainName is in the origin hostname's CNAME chain if possible. This will ensure that the required address records are already present in the client's DNS cache as part of the responses to the address queries that were issued in parallel.

6. Initial SvcParamKeys

A few initial SvcParamKeys are defined here. These keys are useful for HTTPS, and most are applicable to other protocols as well.

6.1. "alpn"

The "alpn" SvcParamKey defines the Application Layer Protocol (ALPN, as defined in {{!RFC7301}}) supported by a TLS-based alternative service. Its value SHOULD be an entry in the IANA registry "TLS Application-Layer Protocol Negotiation (ALPN) Protocol IDs".

The presentation format and wire format of SvcParamValue is its registered "Identification Sequence".

Clients MUST ignore SVCB RRs where the "alpn" SvcParamValue is unknown or unsupported.

6.2. "port"

The "port" SvcParamKey defines the TCP or UDP port that should be used to contact this alternative service.

The presentation format of the SvcParamValue is a numeric value between 0 and 65535 inclusive. The wire format of the SvcParamValue is the corresponding 2 octet numeric value in network byte order.

6.3. "esniconfig"

The SvcParamKey for ESNI is "esniconfig". Its value is defined in Section 8. It is applicable to most TLS-based protocols.

When publishing a record containing an "esniconfig" parameter, the publisher MUST ensure that all IP addresses of SvcDomainName correspond to servers that have access to the corresponding private key or are authoritative for the fallback domain. (See [ESNI] for more details about the fallback domain.) This yields an anonymity set of cardinality equal to the number of ESNI-enabled server domains supported by a given client-facing server. Thus, even with SNI encryption, an attacker who can enumerate the set of ESNI-enabled domains supported by a client-facing server can guess the correct SNI with probability at least 1/K, where K is the size of this ESNI-enabled server anonymity set. This probability may be increased via traffic analysis or other mechanisms.

6.4. "ipv4hint" and "ipv6hint"

The "ipv4hint" and "ipv6hint" keys represent IP address hints for the service. If A and AAAA records for SvcDomainName are locally available, the client SHOULD ignore these hints. Otherwise, clients SHOULD perform A and/or AAAA queries for SvcDomainName as in Section 3, and clients SHOULD use the IP address in those responses for future connections. Clients MAY opt to terminate any connections using the addresses in hints and instead switch to the addresses in response to the SvcDomainName. Failure to use A and/or AAAA response addresses may negatively impact load balancing or other geo-aware features and thereby degrade client performance.

The wire format for each parameter is a sequence of IP addresses in network byte order. Like an A or AAAA RRSet, the list of addresses represents an unordered collection, and clients SHOULD pick addresses to use in a random order.

These parameters MAY be repeated multiple times within a record. When receiving such a record, clients SHOULD combine the sets of addresses.

When selecting between IPv4 and IPv6 addresses to use, clients may use an approach such as [HappyEyeballsV2]. When only "ipv4hint" parameters are present, IPv6-only clients may synthesize IPv6 addresses as specified in [RFC7050] or ignore the "ipv4hint" key and wait for AAAA resolution (Section 3). Recursive resolvers MUST NOT perform DNS64 ([RFC6147]) on parameters within an SVCB record. For best performance, server operators SHOULD include "ipv6hint" parameters whenever they publish "ipv4hint" parameters.

The presentation format for each parameter is a comma-separated list of IP addresses in standard textual format [RFC5952].

These parameters are intended to minimize additional connection latency when a recursive resolver is not compliant with the requirements in Section 4, and SHOULD NOT be included if most clients are using compliant recursive resolvers.

7. Using SVCB with HTTPS and HTTP

Use of any protocol with SVCB requires a protocol-specific mapping specification. This section specifies the mapping for HTTPS and HTTP.

To enable special handling for the HTTPS and HTTP use-cases, the HTTPSSVC RR type is defined as an SVCB-compatible RR type, specific to the https and http schemes. Clients MUST NOT perform SVCB queries or accept SVCB responses for https or http schemes.

The HTTPSSVC wire format and presentation format are identical to SVCB, and both share the SvcParamKey registry. SVCB semantics apply equally to HTTPSSVC unless specified otherwise.

The presence of an HTTPSSVC record for an HTTP or HTTPS service also provides an indication that all resources are available over HTTPS, as discussed in Section 7.4. This allows HTTPSSVC RRs to apply to pre-existing HTTP scheme URLs, while ensuring that the client uses a secure and authenticated HTTPS connection.

The HTTPSSVC RR parallels the concepts introduced in the HTTP Alternative Services proposed standard [AltSvc]. Clients and servers that implement HTTPSSVC are NOT REQUIRED to implement Alt-Svc. However, many clients and servers will implement both, and a partial mapping exists between them (Appendix A).

7.1. Owner names for HTTPSSVC records

The HTTPSSVC RR extends the behavior for determining a QNAME specified above in Section 2.3. In particular, if the scheme is "https" with port 443, or the scheme is "http" and the port is 80, then the client's original QNAME is equal to the origin host name.

For origins other than https with port 443 and http with port 80, the port and scheme continue to be prefixed to the hostname as described in Section 2.3. Following of HTTPSSVC AliasForm and CNAME aliases is also unchanged from SVCB.

Note that none of these forms alter the HTTPS origin or authority. For example, clients MUST continue to validate TLS certificate hostnames based on the origin host.

7.2. Populating Alt-Used

When using an HTTPSSVC RR in ServiceForm, all clients SHOULD include the "Alt-Used" HTTP header (Section 5 of [RFC7838]). The header's value (in ABNF) SHOULD be

uri-host ":" port

where uri-host is the final value of HOST ({client-behavior}) minus the trailing ".", and port is the port number in use.

7.3. Differences from Alt-Svc

Publishing a ServiceForm HTTPSSVC record in DNS is intended to be similar to transmitting the corresponding Alt-Svc field value over HTTPS, and receiving an HTTPSSVC record is intended to be similar to receiving that field value over HTTPS. However, there are some differences in the intended client and server behavior.

7.3.1. Untrusted channel

SVCB does not require or provide any assurance of authenticity. (DNSSEC signing and verification, which would provide such assurance, are OPTIONAL.) The DNS resolution process is treated as an untrusted channel that learns only the QNAME, and is prevented from mounting any attack beyond denial of service.

Alt-Svc parameters that cannot be safely received in this model MUST NOT have a corresponding defined SvcParamKey. For example, there is no SvcParamKey corresponding to the Alt-Svc "persist" parameter, because this parameter is not safe to accept over an untrusted channel.

7.3.2. Caching and granularity

There is no SvcParamKey corresponding to the Alt-Svc "ma" (max age) parameter. Instead, server operators SHOULD encode the expiration time in the DNS TTL.

Some DNS caching systems incorrectly extend the lifetime of DNS records beyond the stated TTL. Server operators MUST NOT rely on HTTPSSVC records expiring on time, and MAY shorten the TTL to compensate.

Sending Alt-Svc over HTTP allows the server to tailor the Alt-Svc Field Value specifically to the client. When using an HTTPSSVC DNS record, groups of clients will necessarily receive the same Alt-Svc Field Value. Therefore, HTTPSSVC is not suitable for uses that require single-client granularity.

If the client has an Alt-Svc cache, and a usable Alt-Svc value is present in that cache, then the client MAY skip the HTTPSSVC query.

If the client has a cached Alt-Svc entry that is expiring, the client MAY perform an HTTPSSVC query to refresh the entry.

7.4. HTTP Strict Transport Security

By publishing an HTTPSSVC record, the server operator indicates that all useful HTTP resources on that origin are reachable over HTTPS, similar to HTTP Strict Transport Security [HSTS]. When an HTTPSSVC record is present for an origin, all "http" scheme requests for that origin SHOULD logically be redirected to "https".

Prior to making an "http" scheme request, the client SHOULD perform a lookup to determine if an HTTPSSVC record is available for that origin. To do so, the client SHOULD construct a corresponding "https" URL as follows:

1. Replace the "http" scheme with "https".

2. If the "http" URL explicitly specifies port 80, specify port 443.

3. Do not alter any other aspect of the URL.

This construction is equivalent to Section 8.3 of [HSTS], point 5.

If an HTTPSSVC record is present for this "https" URL, the client should treat this as the equivalent of receiving an HTTP "307 Temporary Redirect" redirect to the "https" URL. Because HTTPSSVC is received over an often insecure channel (DNS), clients MUST NOT place

any more trust in this signal than if they had received a 307 redirect over cleartext HTTP.

If the HTTPSSVC query results in a SERVFAIL error, and the connection between the client and the recursive resolver is cryptographically protected (e.g. using TLS [RFC7858] or HTTPS [RFC8484]), the client SHOULD abandon the connection attempt and display an error message. A SERVFAIL error can occur if the domain is DNSSEC-signed, the recursive resolver is DNSSEC-validating, and an active attacker between the recursive resolver and the authoritative DNS server is attempting to prevent the upgrade to HTTPS.

Similarly, if the client enforces DNSSEC validation on A/AAAA responses, it SHOULD abandon the connection attempt if the HTTPSSVC response fails to validate.

8. Alt-Svc and SVCB/HTTPSSVC parameter for ESNI keys

Both SVCB/HTTPSSVC and Alt-Svc "esniconfig" parameters are defined for conveying the ESNI configuration of an alternative service. The value of the parameter is an ESNIConfig structure [ESNI] or the empty string. ESNI-aware clients SHOULD prefer alt-values and SVCB/HTTPSSVC RRs with non-empty esniconfig.

Both the SVCB SvcParamValue presentation format as well as the Alt-Svc parameter value is the ESNIConfig structure [ESNI] encoded in [base64] or the empty string. The SVCB SvcParamValue wire format is the octet string containing the binary ESNIConfig structure.

This parameter MAY also be sent in Alt-Svc HTTP response headers and HTTP/2 ALTSVC frames. This parameter MUST NOT appear more than once in a single alt-value.

8.1. Handling a mixture of alternatives not supporting ESNI

The Alt-Svc specification states that "the client MAY fall back to using the origin" in case of connection failure (Section 2.4 of [AltSvc]). This behavior is not suitable for ESNI, because fallback would negate the privacy benefits of ESNI.

Accordingly, any connection attempt that uses ESNI MUST fall back only to another alt-value that also has the esniconfig parameter. If the parameter's value is the empty string, the client SHOULD connect as it would in the absence of any ESNIConfig information.

For example, suppose a server operator has two alternatives. Alternative A is reliably accessible but does not support ESNI. Alternative B supports ESNI but is not reliably accessible. The

server operator could include a full esniconfig value in Alternative B, and mark Alternative A with esniconfig="" to indicate that fallback from B to A is allowed.

Other clients and services implementing SVCB or HTTPSSVC with esniconfig are encouraged to take a similar approach.

9. Interaction with other standards

This standard is intended to reduce connection latency and improve user privacy. Server operators implementing this standard SHOULD also implement TLS 1.3 [RFC8446] and OCSP Stapling [RFC6066], both of which confer substantial performance and privacy benefits when used in combination with SVCB records.

To realize the greatest privacy benefits, this proposal is intended for use over a privacy-preserving DNS transport (like DNS over TLS [RFC7858] or DNS over HTTPS [RFC8484]). However, performance improvements, and some modest privacy improvements, are possible without the use of those standards.

Any specification for use of SVCB with a protocol MUST have an entry for its scheme under the SVCB RR type in the IANA DNS Underscore Global Scoped Entry Registry [Attrleaf]. The scheme SHOULD have an entry in the IANA URI Schemes Registry [RFC7595]. The scheme SHOULD have a defined specification for use with SVCB.

10. Security Considerations

SVCB/HTTPSSVC RRs are intended for distribution over untrusted channels, and clients are REQUIRED to verify that the alternative service is authoritative for the origin (Section 2.1 of [AltSvc]). Therefore, DNSSEC signing and validation are OPTIONAL for publishing and using SVCB and HTTPSSVC records.

Clients MUST ensure that their DNS cache is partitioned for each local network, or flushed on network changes, to prevent a local adversary in one network from implanting a forged DNS record that allows them to track users or hinder their connections after they leave that network.

11. IANA Considerations

11.1. New registry for Service Parameters

The "Service Binding (SVCB) Parameter Registry" defines the name space for parameters, including string representations and numeric

SvcParamKey values. This registry is shared with other SVCB-compatible RR types, such as HTTPSSVC.

ACTION: create and include a reference to this registry.

11.1.1. Procedure

A registration MUST include the following fields:

- Name: Service parameter key name
- SvcParamKey: Service parameter key numeric identifier (range 0-65535)
- Meaning: a short description
- Pointer to specification text

Values to be added to this name space require Expert Review (see [RFC5226], Section 4.1). Apart from the initial contents, the name MUST NOT start with "key".

11.1.2. Initial contents

The "Service Binding (SVCB) Parameter Registry" shall initially be populated with the registrations below:

SvcParamKey	NAME	Meaning	Reference
0	key0	Reserved	(This document)
1	alpn	ALPN for alternative service	(This document)
2	port	Port for alternative service	(This document)
3	esniconfig	Encrypted SNI configuration	(This document)
4	ipv4hint	IPv4 address hints	(This document)
5	key5	Reserved	(This document)
6	ipv6hint	IPv6 address hints	(This document)
65280-65534	keyNNNNN	Private Use	(This document)
65535	key65535	Reserved	(This document)

TODO: do we also want to reserve a range for greasing?

11.2. Registry updates

Per [RFC6895], please add the following entry to the data type range of the Resource Record (RR) TYPEs registry:

TYPE	Meaning	Reference
SVCB	Service Location and Parameter Binding	(This document)
HTTPSSVC	HTTPS Service Location and Parameter Binding	(This document)

Per [Attrleaf], please add the following entries to the DNS Underscore Global Scoped Entry Registry:

RR TYPE	_NODE NAME	Meaning	Reference
HTTPSSVC	_https	Alt-Svc for HTTPS	(This document)
HTTPSSVC	_http	Alt-Svc for HTTPS	(This document)

Per [AltSvc], please add the following entry to the HTTP Alt-Svc Parameter Registry:

Alt-Svc Parameter	Meaning	Reference
esniconfig	Encrypted SNI configuration	(This document)

12. Acknowledgments and Related Proposals

There have been a wide range of proposed solutions over the years to the "CNAME at the Zone Apex" challenge proposed. These include [I-D.draft-bellis-dnsop-http-record-00], [I-D.draft-ietf-dnsop-aname-03], and others.

Thank you to Ian Swett, Ralf Weber, Jon Reed, Martin Thompson, Lucas Pardue, Ilari Liusvaara, Tim Wicinski, Tommy Pauly, Chris Wood, and others for their feedback and suggestions on this draft.

13. References

13.1. Normative References

[AltSvc] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <https://www.rfc-editor.org/info/rfc7838>.

[AltSvcSNI] Bishop, M., "The "SNI" Alt-Svc Parameter", draft-bishop-httpbis-sni-altsvc-02 (work in progress), May 2018.

[Attrleaf] Crocker, D., "DNS Scoped Data Through "Underscore" Naming of Attribute Leaves", draft-ietf-dnsop-attrleaf-16 (work in progress), November 2018.

[base64] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[ESNI] Rescorla, E., Oku, K., Sullivan, N., and C. Wood, "Encrypted Server Name Indication for TLS 1.3", draft-ietf-tls-esni-04 (work in progress), July 2019.

[HappyEyeballsV2] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[HSTS] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <https://www.rfc-editor.org/info/rfc6797>.

[HTTP3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", draft-ietf-quic-http-20 (work in progress), April 2019.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, DOI 10.17487/RFC1928, March 1996, <https://www.rfc-editor.org/info/rfc1928>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, DOI 10.17487/RFC3225, December 2001, <https://www.rfc-editor.org/info/rfc3225>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <https://www.rfc-editor.org/info/rfc3597>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <https://www.rfc-editor.org/info/rfc6454>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <https://www.rfc-editor.org/info/rfc7595>.

[RFC7838] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <https://www.rfc-editor.org/info/rfc7838>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

13.2. Informative References

[DNSTerm] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[HTTP] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[I-D.draft-bellis-dnsop-http-record-00] Bellis, R., "A DNS Resource Record for HTTP", draft-bellis-dnsop-http-record-00 (work in progress), November 2018.

[I-D.draft-ietf-dnsop-aname-03] Finch, T., Hunt, E., Dijk, P., Eden, A., and W. Mekking, "Address-specific DNS aliases (ANAME)", draft-ietf-dnsop-aname-03 (work in progress), April 2019.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <https://www.rfc-editor.org/info/rfc6895>.

Appendix A. Mapping between HTTPSSVC and Alt-Svc

Conversion between HTTPSSVC's ServiceForm and Alt-Svc is possible. Note that conversion in either direction can be lossy, because some parameters are only defined for HTTPSSVC or Alt-Svc.

To construct an Alt-Svc Field Value (as defined in Section 4 of [AltSvc]) from an HTTPSSVC record:

- The SvcDomainName is mapped into the uri-host portion of alt-authority with the trailing "." removed. (If SvcDomainName is ".", the special handling described in Section 2.6.1 MUST be applied first.)

- The SvcParamValue of the "port" service parameter, or 443 if no such parameter is present, is written to the port portion of the alt-authority.

- The SvcParamValue of the "alpn" service parameter is mapped to the protocol-id. This MUST follow the normalization and encoding requirements for protocol-id specified in [AltSvc] Section 3. This parameter is MANDATORY.

- The DNS TTL is mapped to the "ma" (max age) Alt-Svc parameter.

- For SVCB parameters with defined mappings to HTTPS Alt-Svc, each should be included as an Alt-Svc parameter, typically as the SvcParamKey name "=" a defined encoding of the SvcParamValue.

Converting an Alt-Svc Field Value into an HTTPSSVC record follows the reverse of this procedure.

Conversion between HTTPSSVC and Alt-Svc Field Value MUST ignore any SvcParamKeys and Alt-Svc parameters that are unrecognized or do not have a defined mapping.

For example, if the operator of https://www.example.com intends to include an HTTP response header like

```
Alt-Svc: h3="svc.example.net:8003"; ma=3600; foo=123, \
         h2="svc.example.net:8002"; ma=3600; foo=123
```

they could also publish an HTTPSSVC DNS RRSet like

```
www.example.com. 3600 IN HTTPSSVC 2 svc.example.net. (
                                    alpn=h3 port=8003 foo=123 )
                         HTTPSSVC 3 svc.example.net. (
                                    alpn=h2 port=8002 foo=123 )
```

Where "foo" is a hypothetical future HTTPSSVC and Alt-Svc parameter.

This data type can also be represented as an Unknown RR as described in [RFC3597]:

```
www.example.com. 3600 IN TYPE??? \\# TBD:WRITEME
```

A.1. Multiple records and preference ordering

Server operators MAY publish multiple ServiceForm HTTPSSVC records as an RRSet. When converting a collection of alt-values into an HTTPSSVC RRSet, the server operator MUST set the overall TTL to a value no larger than the minimum of the "max age" values (following Section 5.2 of [RFC2181]).

Each RR corresponds to exactly one alt-value, as described in Section 3 of [AltSvc].

As discussed in Section 2.6.2, HTTPSSVC RRs with a smaller SvcFieldPriority value SHOULD be sorted ahead of and given preference over RRs with a larger SvcFieldPriority value.

When constructing equivalent Alt-Svc headers from an RRSet:

1. The RRs SHOULD be ordered by increasing SvcFieldPriority, with shuffling for equal SvcFieldPriority values. Clients MAY choose to further prioritize alt-values where address records are immediately available for the alt-value's SvcDomainName.

2. The client SHOULD concatenate the thus-transformed-and-ordered SvcFieldValues in the RRSet, separated by commas. (This is semantically equivalent to receiving multiple Alt-Svc HTTP response headers, according to Section 3.2.2 of [HTTP]).

A.2. Additional examples

The following:

```
www.example.com.  7200  IN CNAME    svc.example.net.
example.com.      7200  IN HTTPSSVC 0 svc.example.net.
svc.example.net.  7200  IN HTTPSSVC 2 svc3.example.net. (
    alpn=h3 port=8003 esniconfig="ABC..." )
svc.example.net.  7200  IN HTTPSSVC 3 . (
    alpn=h2 port=8002 esniconfig="123..." )
```

is equivalent to the Alt-Svc record:

```
Alt-Svc: h3="svc3.example.net:8003"; esniconfig="ABC..."; ma=7200, \
         h2="svc.example.net:8002"; esniconfig="123..."; ma=7200
```

for the origins of both "https://www.example.com" and "https://example.com".

Appendix B. Comparison with alternatives

The SVCB and HTTPSSVC record types closely resemble, and are inspired by, some existing record types and proposals. A complaint with all of the alternatives is that web clients have seemed unenthusiastic about implementing them. The hope here is that by providing an extensible solution that solves multiple problems we will overcome the inertia and have a path to achieve client implementation.

B.1. Differences from the SRV RR type

An SRV record [RFC2782] can perform a similar function to the SVCB record, informing a client to look in a different location for a service. However, there are several differences:

- SRV records are typically mandatory, whereas clients will always continue to function correctly without making use of Alt-Svc or SVCB.
- SRV records cannot instruct the client to switch or upgrade protocols, whereas Alt-Svc can signal such an upgrade (e.g. to HTTP/2).
- SRV records are not extensible, whereas SVCB and HTTPSSVC can be extended with new parameters.
- Using SRV records would not allow an HTTPS client to skip processing of the Alt-Svc information in a subsequent connection, so it does not confer a performance advantage.

B.2. Differences from the proposed HTTP record

Unlike [I-D.draft-bellis-dnsop-http-record-00], this approach is extensible to cover Alt-Svc and ESNI use-cases. Like that proposal, this addresses the zone apex CNAME challenge.

Like that proposal it remains necessary to continue to include address records at the zone apex for legacy clients.

B.3. Differences from the proposed ANAME record

Unlike [I-D.draft-ietf-dnsop-aname-03], this approach is extensible to cover Alt-Svc and ESNI use-cases. This approach also does not require any changes or special handling on either authoritative or master servers, beyond optionally returning in-bailiwick additional records.

Like that proposal, this addresses the zone apex CNAME challenge for clients that implement this.

However with this SVCB proposal it remains necessary to continue to include address records at the zone apex for legacy clients. If deployment of this standard is successful, the number of legacy clients will fall over time. As the number of legacy clients declines, the operational effort required to serve these users without the benefit of SVCB indirection should fall. Server operators can easily observe how much traffic reaches this legacy endpoint, and may remove the apex's address records if the observed legacy traffic has fallen to negligible levels.

B.4. Differences from the proposed ESNI record

Unlike [ESNI], this approach is extensible and covers the Alt-Svc case as well as addresses the zone apex CNAME challenge.

By using the Alt-Svc model we also provide a way to solve the ESNI multi-CDN challenges in a general case.

Unlike ESNI, SVCB allows specifying different ESNI configurations for different protocols and ports, rather than applying a single configuration to all ports on a domain.

B.5. SNI Alt-Svc parameter

Defining an Alt-Svc sni= parameter (such as from [AltSvcSNI]) would have provided some benefits to clients and servers not implementing ESNI, such as for specifying that "_wildcard.example.com" could be sent as an SNI value rather than the full name. There is nothing precluding SVCB from being used with an sni= parameter if one were to be defined, but it is not included here to reduce scope, complexity, and additional potential security and tracking risks.

Appendix C. Design Considerations and Open Issues

This draft is intended to be a work-in-progress for discussion. Many details are expected to change with subsequent refinement. Some known issues or topics for discussion are listed below.

C.1. Record Name

Naming is hard. "SVCB" and "HTTPSSVC" are proposed as placeholders that are easy to search for and replace when a final name is chosen. Other names for this record might include B, ALTSVC, HTTPS, HTTPSSRV, HTTPSSVC, SVCHTTPS, or something else.

C.2. Generality

The SVCB record was designed as a generalization of HTTPSSVC, based on feedback requesting a solution that applied to protocols pther than HTTP. Past efforts to over-generalize have not met with broad success, but we hope that HTTPSSVC and SVCB have struck an acceptable balance between generality and focus.

C.3. Wire Format

Advice from experts in DNS wire format best practices would be greatly appreciated to refine the proposed details, overall.

C.4. Where to include Priority

The SvcFieldPriority could alternately be included as a pri= Alt-Svc attribute. It wouldn't be applicable for Alt-Svc returned via HTTP, but it is also not necessarily needed by DNS servers. It is also not used for AliasForm RRs.

C.5. Whether to include Weight

Some other similar mechanisms such as SRV have a weight in-addition to priority. That is excluded here for simplicity. It could always be added as an optional SVCB parameter.

Appendix D. Change history

- draft-ietf-dnsop-svcb-httpssvc-01
  - Reduce the emphasis on conversion between HTTPSSVC and Alt-Svc
  - Make the "untrusted channel" concept more precise.
  - Make SvcFieldPriority = 0 the definition of AliasForm, instead of a requirement.
- draft-ietf-dnsop-svcb-httpssvc-00
  - Document an optimization for optimistic pre-connection. (Chris Wood)

    * Relax IP hint handling requirements. (Eric Rescorla)

- draft-nygren-dnsop-svcb-httpssvc-00

    * Generalize to an SVCB record, with special-case handling for Alt-Svc and HTTPS separated out to dedicated sections.

    * Split out a separate HTTPSSVC record for the HTTPS use-case.

    * Remove the explicit SvcRecordType=0/1 and instead make the AliasForm vs ServiceForm be implicit. This was based on feedback recommending against subtyping RR type.

    * Remove one optimization.

- draft-nygren-httpbis-httpssvc-03

    * Change redirect type for HSTS-style behavior from 302 to 307 to reduce ambiguities.

- draft-nygren-httpbis-httpssvc-02

    * Remove the redundant length fields from the wire format.

    * Define a SvcDomainName of "." for SvcRecordType=1 as being the HTTPSSVC RRNAME.

    * Replace "hq" with "h3".

- draft-nygren-httpbis-httpssvc-01

    * Fixes of record name. Replace references to "HTTPSVC" with "HTTPSSVC".

- draft-nygren-httpbis-httpssvc-00

    * Initial version

Authors' Addresses

Ben Schwartz
Google

Email: bemasc@google.com

Mike Bishop
Akamai Technologies

Email: mbishop@evequefou.be

Erik Nygren
Akamai Technologies

Email: erik+ietf@nygren.org