| | |
|---|---|
| Network Working Group | I.B.C. Baz Castillo |
| Internet-Draft | XtraTelecom S.A. |
| Intended status: Standards Track | April 14, 2011 |
| Expires: October 16, 2011 | |

DNS SRV Resource Records for the WebSocket Protocol
draft-ibc-websocket-dns-srv-02

## Abstract

This document specifies the usage of DNS SRV resource records by WebSocket clients when resolving a "ws:" or "wss:" Uniform Resource Identifier (URI). The DNS SRV mechanism confers load-balancing and failover capabilities for WebSocket service providers.

## Status of this Memo



## Copyright Notice



## Table of Contents

*1. Introduction

*2. Conventions

*3. Implementation

*4. Client Usage

*4.1. SRV Lookup

*4.2. Fallback Process

*4.3. Reusing TCP Connection

*4.4. Server Failure

*5. Examples

*5.1. Load Balancing and Failover

*5.2. Reusing TCP Connection

*6. Security Considerations

*7. IANA Considerations

*8. References

*8.1. Normative References

*8.2. Informative References

*Appendix A. Change Log (to be removed by RFC Editor prior to publication)

*Appendix A.1. Changes in -02

*Appendix A.2. Changes in -01

*Author's Address

## 1. Introduction

DNS SRV [RFC2782] is widely implemented in realtime communication protocols as SIP [RFC3261] and XMPP [RFC6120]. In both protocols the clients perform a DNS SRV query to get a list of connection addresses (pairs of IP address and port) for the given domain. The administrator of the domain can configure its DNS SRV records in a way that they provide automatic load-balancing along with redundancy/failover capability.
DNS SRV mechanism facilitates network applications scalability without requiring an intermediary node distributing the traffic in load-balancing or failover fashion. Instead, DNS SRV mechanism just requires a proper DNS setup.

By introducing DNS SRV records into WebSocket protocol [I-D.ietf-hybi-thewebsocketprotocol], WebSocket providers can, optionally, take same advantages and provide scalable services with a minimal infrastructure. This specification mandates the usage of DNS SRV resource records by WebSocket clients when resolving a "ws:" or "wss:" URI [RFC3986], but still leaves the decision of using SRV records up to the service administrator.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Implementation

This specification mandates the implementation of DNS SRV [RFC2782] in WebSocket [I-D.ietf-hybi-thewebsocketprotocol] clients (usually web browsers). Said that, WebSocket clients MUST implement this specification.
The client application (usually JavaScript code executed by the web browser) is not aware of the mechanism described in this document which is fully transparent for web developers and JavaScript developers. This is, the client application (usually JavaScript code) does not deal with DNS SRV resolution but just passes the given "ws:" or "wss:" URI to the WebSocket client which MUST perform steps in Section 4.
It is up to the system administrator whether to set, or not, DNS SRV resource records for the WebSocket protocol within the provided service. This specification allows the system administrator to use the DNS SRV [RFC2782] mechanism to improve the service reliability by providing load-balancing and failover capabilities, but does not mandate it (the system administrator could choose whichever scalability strategy).

## 4. Client Usage

DNS SRV lookup just applies when the host component of a WebSocket URI [RFC3986] is a domain and the URI does not contain an explicit port. If this is not the case, the WebSocket client MUST attempt the fallback process described in Section 4.2.
To clarify it, a WebSocket URI like "ws://example.org/myservice" requires the client to perform SRV resolution while "ws://example.org:80/myservice" does not (as the port is explicitly present in the URI).

### 4.1. SRV Lookup

Given a WebSocket URI ("ws:" or "wss:") in which the host component is a domain ("example.org") and the port is not present, the WebSocket client MUST perform the following steps:

1. If the scheme is "ws:", perform a DNS SRV query whose inputs are:

   *Service: "ws"

   *Proto: "tcp"

   *Name: The host component of the URI

   The resulting query looks like "_ws._tcp.example.org".

2. If the scheme is "wss:", perform a DNS SRV query whose inputs are:

   *Service: "wss"

   *Proto: "tcp"

   *Name: The host component of the URI

   The resulting query looks like "_wss._tcp.example.org".

3. If there is no SRV result, attempt the fallback process described in Section 4.2 and omit next steps.

4. If there is SRV result, it will contain one or more DNS SRV resource records (combinations of a domain target, port, priority attribute and weight attribute as described in [RFC2782]).

5. Choose one of the returned DNS SRV resource records (following the rules in [RFC2782]) and perform DNS A or AAAA lookups on the corresponding domain target. This will result in a list of one or more IPv4 or IPv6 addresses.

   *If the DNS A or AAAA lookup returns no result, it is considered an error and next DNS SRV resource record (according to rules in [RFC2782]) MUST be tried.

6. Use the first resolved IP address (with the corresponding port number in the DNS SRV resource record) as the connection address for the WebSocket service.

   *The client MAY now perform steps in Section 4.3 and reuse an existing TCP connection if available.

7. If the WebSocket establishment fails using that connection address because of a server failure (according to Section 4.4) but the A or AAAA lookups returned more than one IP address, then use the next resolved IP address for the connection address (keeping same port).

8. If the WebSocket establishment fails using all the resolved IP addresses for a given DNS SRV resource record, then repeat the process for the next DNS SRV resource record based on priority and weight attributes as defined in [RFC2782] until all the DNS SRV resource records have been tried.

9. If all the attempts fail, internally report the WebSocket establishment error.

When the client constructs the WebSocket handshake HTTP request, the URI MUST be set as described in Section 3.2 of [I-D.ietf-hybi-thewebsocketprotocol] regardless of the usage of SRV mechanism. This is, DNS SRV resolution for a "ws:" or "wss:" URI does not alter the usual construction of the WebSocket handshake request.

### 4.2. Fallback Process

The fallback process SHOULD be a normal A or AAAA address record resolution to determine the IPv4 or IPv6 address of the URI host component (or URI host value without DNS resolution if it contains an IP address).
The server connection port is obtained as stated in Section 3.1 of [I-D.ietf-hybi-thewebsocketprotocol].
If multiple IP addresses have been obtained from a DNS A or AAAA lookup, the client MUST choose the first one and try to establish a WebSocket communication with it. In case such attempt fails because of a server failure (as defined in Section 4.4) the client MUST repeat the process for each remaining IP address.

### 4.3. Reusing TCP Connection

A web browser is able to maintain persistent TCP connections with the HTTP [RFC2616] server and reuse them for sending new HTTP requests. Reusing an existing connection (when available) for WebSocket communication is a desirable behavior which just can take place when both the HTTP server and WebSocket server listen on the same IP address and port.

This section defines how to reuse an existing connection after resolving the location of the WebSocket server using the DNS SRV procedures:

1. The WebSocket client performs the steps in Section 4 and gets an ordered list of connection addresses (pairs of IP address and port) by following rules in [RFC2782].

2. For each connection address the client selects to communicate with, it first checks whether there already exists an established TCP connection against same IP address and port.

3. If so, the client MAY reuse the existing TCP connection for initiating the WebSocket handshake rather than opening a new one.

### 4.4. Server Failure

A WebSocket server failure occurs if the WebSocket establishment (TCP connection and WebSocket handshake procedure) fails because of a cause listed below:

*TCP connection is not possible due to timeout or server side rejection.

*The server does not return a valid HTTP response for the WebSocket handshake request within a specified ammount of time (TODO: specify such ammount).

*The server replies a 500 or 503 HTTP error response during the WebSocket handshake meaning that it suffers of internal problems (i.e. congestion) so it is not currently capable of handling the request.

-If HTTP response code other than 101 (success), 500 or 503 is returned by the server, it MUST NOT be considered a server failure.

-TODO: [I-D.ietf-hybi-thewebsocketprotocol] should describe how to handle different HTTP response codes (as 401 or 302).

## 5. Examples

By properly configuring domain SRV records, the WebSocket service administrator can take advantage of load-balancing and failover capabilities inherent in DNS SRV [RFC2782]. Sections below show some usage cases.

### 5.1. Load Balancing and Failover

```
$ORIGIN example.org.
@         SOA   dns.example.org. root.example.org.
(2011040501 3600 3600 604800 86400)
NS        dns.example.org.
_ws._tcp  SRV   0 3 80 ws1.example.org.
_ws._tcp  SRV   0 1 90 ws2.example.org.
_ws._tcp  SRV   1 0 80 ws3.example.org.

dns       A     1.1.1.100
ws1       A     1.1.1.1
ws2       A     1.1.1.2
ws2       A     1.1.1.3
```

Assuming there are three hosts providing the WebSocket service for the URI "ws://example.org/myservice", the following zone file for a fictional example.org domain provides load-balancing and failover for the WebSocket traffic:

- The first server with domain ws1.example.org listens on IP address 1.1.1.1 and port 80, and its associated DNS SRV record has priority 0 and weight 3.

- The second server with domain ws2.example.org listens on IP address 1.1.1.2 and port 90, and its associated DNS SRV record has priority 0 and weight 1.

- The third server with domain ws3.example.org listens on IP address 1.1.1.3 and port 80, and its associated DNS SRV record has priority 1 and weight 0.

By following the steps in Section 4, 75% of WebSocket clients would choose the first server and the other 25% would choose the second server to communicate with (as both have the higest SRV priority 0 in their respective DNS SRV resource records, and the first server has a SRV weight value which triples the value of the second server).
In case the WebSocket establishment fails because of a server failure (as defined in Section 4.4), WebSocket clients would try the other one.
If the WebSocket establishment fails with both the first and second servers, WebSocket clients would then try the third server (as the priority value in its respective DNS SRV resource record is lower).

### 5.2. Reusing TCP Connection

In this case a server resolving to www.example.org is used for both HTTP and WebSocket traffic, while a second server resolving to ws2.example.com is used for balancing the WebSocket traffic.

```
$ORIGIN example.org.
@         SOA   dns.example.org. root.example.org.
(2011040501 3600 3600 604800 86400)
NS        dns.example.org.
_ws._tcp  SRV   0 1 80 www.example.org.
_ws._tcp  SRV   0 1 80 ws2.example.org.

dns       A     1.1.1.100
www       A     1.1.1.1
ws2       A     1.1.1.2
```

The client (presumably a web browser) would open one or more TCP connections with www.example.org and port 80 for the usual HTTP communication. As the retrieved data contains a WebSocket URI "ws://example.org/myservice" the client would also initialize a WebSocket communication so would perform steps in Section 4.
Such DNS resolution would return two DNS SRV resource records (the first one with www.example.org as domain target and the second one with ws2.example.org as domain target), both of them with same priority and weight attributes.
As per target selection rules in [RFC2782] it is expected that half of the clients would choose www.example.org domain target and port 80 as the WebSocket communication address so they MAY reuse an existing TCP connection previously opened rather than creating a new one.

## 6. Security Considerations

Any Internet protocol offering DNS SRV resource records for locating servers is sensitive to security issues described in [I-D.barnes-hard-problem]. Usage of DNS security extensions (DNSSEC) as described in [RFC4033] is recommended to mitigate the problem.

## 7. IANA Considerations

This specification registers two new SRV Service Labels:

**ws:** MUST be used when constructing a DNS SRV query to locate the WebSocket service address (for regular WebSocket connections).

**wss:** MUST be used when constructing a DNS SRV query to locate the WebSocket service address (for WebSocket connections tunneled over TLS [RFC5246]).

## 8. References

### 8.1. Normative References


| | |
|---|---|
| **[RFC2119]** | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. |
| **[RFC2782]** | Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000. |
| **[RFC3986]** | Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005. |
| **[I-D.ietf-hybi-thewebsocketprotocol]** | Fette, I, "The WebSocket protocol", Internet-Draft draft-ietf-hybi-thewebsocketprotocol-06, February 2011. |


### 8.2. Informative References


| | |
|---|---|
| **[RFC3261]** | Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002. |
| **[RFC6120]** | Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011. |
| **[RFC5246]** | Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008. |
| **[I-D.barnes-hard-problem]** | Barnes, R and P Saint-Andre, "High Assurance Re-Direction (HARD) Problem Statement", Internet-Draft draft-barnes-hard-problem-00, July 2010. |
| **[RFC4033]** | Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005. |
| **[RFC2616]** | Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999. |


## Appendix A. Change Log (to be removed by RFC Editor prior to publication)

### Appendix A.1. Changes in -02

*Category changed to "std" (Standards-Track document).

*Editorial fixes.

*Section "Introduction" extended.

*Added a section "Implementation".

*Use "DNS SRV resource record" to refer a record in the DNS SRV lookup.

*Improvements in section "Fallback Process".

*Section "Websocket Establishment Fails" renamed to "Server Failure".

*Section "Examples" simplified.

## Appendix A.2. Changes in -01

*Editorial fixes.

*Avoid the word "target" when referring to connection addresses.

*Improvements in section "Examples".

## Author's Address

Inaki Baz Castillo Baz Castillo XtraTelecom S.A. Barakaldo, Basque Country Spain EMail: ibc@aliax.net