

A DNS RR for specifying the location of services (DNS SRV)

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This document describes a DNS RR which specifies the location of the server(s) for a specific protocol and domain (like a more general form of MX).

Overview and rationale

Currently, one must either know the exact address of a server to contact it, or broadcast a question. This has led to, for example, ftp.whatever.com aliases [[RFC 2219](#)], the SMTP-specific MX RR, and using MAC-level broadcasts to locate servers.

The SRV RR allows administrators to use several servers for a single domain, to move services from host to host with little fuss, and to designate some hosts as primary servers for a service and others as backups.

Clients ask for a specific service/protocol for a specific domain (the word domain is used here in the strict [RFC 1034](#) sense), and get back the names of any available servers.

Note that where this document refers to "address records", it means A RR's, AAAA RR's, or their most modern equivalent.

Introductory example

If a SRV-cognizant web-browser wants to retrieve

`http://www.example.com/`

it does a lookup of

`_http._tcp.www.example.com`

and retrieves the document from one of the servers in the reply. The example zone file near the end of this memo contains answering RRs for this query.

Definitions

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT" and "MAY" used in this document are to be interpreted as specified in [BCP 14](#). Other terms used in this document are defined in the DNS specification, [RFC 1034](#).

The format of the SRV RR

Here is the format of the SRV RR, whose DNS type code is 33:

`_Service._Proto.Name TTL Class SRV Priority Weight Port Target`

(There is an example near the end of this document.)

Service

The symbolic name of the desired service, as defined in Assigned Numbers [STD 2] or locally. An underscore (underscore) is prepended to the service identifier to avoid collisions with DNS labels that occur in nature.

Some widely used services, notably POP, don't have a single universal name. If Assigned Numbers names the service indicated, that name is the only name which is legal for SRV lookups. Only locally defined services may be named locally. The Service is case insensitive.

Proto

The symbolic name of the desired protocol, with an underscore (underscore) prepended to prevent collisions with DNS labels that occur

in nature. `_TCP` and `_UDP` are at present the most useful values for this field, though any name defined by Assigned Numbers or locally may be used (as for Service). The Proto is case insensitive.

Name

The domain this RR refers to. The SRV RR is unique in that the name one searches for is not this name; the example near the end shows this clearly.

TTL

Standard DNS meaning [[RFC 1035](#)].

Class

Standard DNS meaning [[RFC 1035](#)]. SRV records occur in the IN Class.

Priority

As for MX, the priority of this target host. A client MUST attempt to contact the target host with the lowest-numbered priority it can reach; target hosts with the same priority SHOULD be tried in an order defined by the weight field. The range is 0-65535. This is a 16 bit binary integer in network byte order.

Weight

A load balancing mechanism. When selecting a target host among the those that have the same priority, the chance of trying this one first SHOULD be proportional to its weight, as specified below. Larger weights lead to a higher probability of being selected. The range of this number is 0-65535. This is a 16 bit binary integer in network byte order. Domain administrators are urged to use Weight 0 when there isn't any load balancing to do, to make the RR easier to read for humans (less noisy). In the presence records containing weights greater than 0, records with weight 0 have a very small chance of being selected.

To choose the target, the client SHOULD implement the effect of this algorithm. This permits administrators to plan weights to achieve the load distribution desired. Each time a target is needed, the client should order the remaining (not previously used) SRV RRs at the current priority in any random fashion, except placing all those with weight 0 at the beginning of the list. Compute the sum of the weights of those RRs, and with each RR associate the running sum in the selected order. Then choose a random number (not necessarily of integral value) between 0 and the sum computed (inclusive), and select the RR whose running sum value is the first in the selected order which

is greater than or equal to the random number selected.

Port

The port on this target host of this service. The range is 0-65535. This is a 16 bit binary integer in network byte order. This is often as specified in Assigned Numbers but need not be.

Target

As for MX, the domain name of the target host. There MUST be one or more address records for this name, the name MUST NOT be an alias (in the sense of [RFC 1034](#) or [RFC 2181](#)). Implementors are urged, but not required, to return the address record(s) in the Additional Data section. Unless and until permitted by future standards action, name compression is not to be used for this field.

A Target of "." means that the service is decidedly not available at this domain.

Applicability Statement

In general, it is expected that SRV records will be used by clients for applications where the relevant protocol specification indicates that clients should use the SRV record. The examples in this document use familiar protocols as an aid in understanding. It is not intended that those protocols will necessarily use SRV records.

Domain administrator advice

Expecting everyone to update their client applications when the first internet site adds a SRV RR for some server is futile (even if desirable). Therefore SRV would have to coexist with address record lookups for existing protocols, and DNS administrators should try to provide address records to support old clients:

- Where the services for a single domain are spread over several hosts, it seems advisable to have a list of address records at the same DNS node as the SRV RR, listing reasonable (if perhaps suboptimal) fallback hosts for Telnet, NNTP and other protocols likely to be used with this name. Note that some programs only try the first address they get back from e.g. `gethostbyname()`, and we don't know how widespread this behavior is.
- Where one service is provided by several hosts, one can either provide address records for all the hosts (in which case the round-robin mechanism, where available, will share the load equally) or just for one (presumably the fastest).

- If a host is intended to provide a service only when the main server(s) is/are down, it probably shouldn't be listed in address records.
- Hosts that are referenced by backup address records must use the port number specified in Assigned Numbers for the service.
- Designers of future protocols for which "secondary servers" is not useful (or meaningful) may choose to not use SRV's support for secondary servers. Clients for such protocols may use or ignore SRV RRs with Priority higher than the RR with the lowest Priority for a domain.

Currently there's a practical limit of 512 bytes for DNS replies. Until all resolvers can handle larger responses, domain administrators are strongly advised to keep their SRV replies below 512 bytes.

All round numbers, wrote Dr. Johnson, are false, and these numbers are very round: A reply packet has a 30-byte overhead plus the name of the service ("_telnet._tcp.example.com" for instance); each SRV RR adds 20 bytes plus the name of the target host; each NS RR in the NS section is 15 bytes plus the name of the name server host; and finally each A RR in the additional data section is 20 bytes or so, and there are A's for each SRV and NS RR mentioned in the answer. This size estimate is extremely crude, but shouldn't underestimate the actual answer size by much. If an answer may be close to the limit, using a DNS query tool (e.g. "dig") to look at the actual answer is a good idea.

The "Weight" field

Weight, the load balancing field, is not quite satisfactory, but the actual load on typical servers changes much too quickly to be kept around in DNS caches. It seems to the authors that offering administrators a way to say "this machine is three times as fast as that one" is the best that can practically be done.

The only way the authors can see of getting a "better" load figure is asking a separate server when the client selects a server and contacts it. For short-lived services like SMTP an extra step in the connection establishment seems too expensive, and for long-lived services like telnet, the load figure may well be thrown off a minute after the connection is established when someone else starts or finishes a heavy job.

The Port number

Currently, the translation from service name to port number happens at the client, often using a file such as /etc/services.

Moving this information to the DNS makes it less necessary to update these files on every single computer of the net every time a new service is added, and makes it possible to move standard services out of the "root-only" port range on unix.

Usage rules

A SRV-cognizant client SHOULD use this procedure to locate a list of servers and connect to the preferred one:

Do a lookup for QNAME=_service._protocol.target, QCLASS=IN, QTYPE=SRV.

If the reply is NOERROR, ANCOUNT>0 and there is at least one SRV RR which specifies the requested Service and Protocol in the reply:

If there is precisely one SRV RR, and its Target is "." (the root domain), abort.

Else, for all such RR's, build a list of (Priority, Weight, Target) tuples

Sort the list by priority (lowest number first)

Create a new empty list

For each distinct priority level

While there are still elements left at this priority level

Select an element randomly, with probability Weight, as specified above, and move it to the tail of the new list

For each element in the new list

query the DNS for address records for the Target or use any such records found in the Additional Data section of the earlier SRV response.

for each address record found, try to connect to the (protocol, address, service).

else if the service desired is SMTP (and SMTP has been defined elsewhere to expect SRV lookups)

skip to [RFC 974](#) (MX).

else

Do a lookup for QNAME=target, QCLASS=IN, QTYPE=A

for each address record found, try to connect to the
(protocol, address, service)

Notes:

- Port numbers SHOULD NOT be used in place of the symbolic service or protocol names (for the same reason why variant names cannot be allowed: Applications would have to do two or more lookups).
- If a truncated response comes back from an SRV query, the rules described in [[RFC2181](#)] shall apply.
- A client MAY use means other than Weight to choose among target hosts with equal Priority.
- A client MUST parse all of the RR's in the reply.
- If the Additional Data section doesn't contain address records for all the SRV RR's and the client may want to connect to the target host(s) involved, the client MUST look up the address record(s). (This happens quite often when the address record has shorter TTL than the SRV or NS RR's.)
- Future protocols could be designed to use SRV RR lookups as the means by which clients locate their servers.

Fictional example

This is (part of) the zone file for example.com, a still-unused domain:

```
$ORIGIN example.com.  
@ SOA server.example.com. root.example.com. (  
    1995032001 3600 3600 604800 86400 )  
NS server.example.com.  
NS ns1.ip-provider.net.  
NS ns2.ip-provider.net.
```



```

_ftp._tcp      SRV 0 0 21 server.example.com.
_finger._tcp   SRV 0 0 79 server.example.com.
; telnet - use old-slow-box or new-fast-box if either is
; available, make three quarters of the logins go to
; new-fast-box.
_telnet._tcp    SRV 0 1 23 old-slow-box.example.com.
                SRV 0 3 23 new-fast-box.example.com.
; if neither old-slow-box or new-fast-box is up, switch to
; using the sysadmin's box and the server
                SRV 1 0 23 sysadmins-box.example.com.
                SRV 1 0 23 server.example.com.
; HTTP - server is the main server, new-fast-box is the backup
; (On new-fast-box, the HTTP daemon runs on port 8000)
_http._tcp      SRV 0 0 80 server.example.com.
                SRV 10 0 8000 new-fast-box.example.com.
; since we want to support both http://example.com/ and
; http://www.example.com/ we need the next two RRs as well
_http._tcp.www  SRV 0 0 80 server.example.com.
                SRV 10 0 8000 new-fast-box.example.com.
; SMTP - mail goes to the server, and to the IP provider if
; the net is down
_smtp._tcp      SRV 0 0 25 server.example.com.
                SRV 1 0 25 mailhost.ip-provider.net.
@               MX 0 server.example.com.
                MX 1 mailhost.ip-provider.net.
; NNTP - use the IP provider's NNTP server
_nntp._tcp      SRV 0 0 119 nntphost.ip-provider.net.
; IDB is an locally defined protocol
_idb._tcp SRV 0 0 2025 new-fast-box.example.com.
; addresses
server          A   172.30.79.10
old-slow-box    A   172.30.79.11
sysadmins-box  A   172.30.79.12
new-fast-box    A   172.30.79.13
; backup address records - new-fast-box and old-slow-box are
; included, naturally, and server is too, but might go
; if the load got too bad
@               A   172.30.79.10
                A   172.30.79.11
                A   172.30.79.13
; backup address record for www.example.com
www             A     172.30.79.10
; NO other services are supported
*._tcp         SRV 0 0 0 .
*._udp         SRV 0 0 0 .

```

In this example, a telnet connection to "example.com." needs an SRV lookup of "_telnet._tcp.example.com." and possibly A lookups of "new-

fast-box.example.com." and/or the other hosts named. The size of the SRV reply is approximately 365 bytes:

- 30 bytes general overhead
- 20 bytes for the query string, "_telnet._tcp.example.com."
- 130 bytes for 4 SRV RR's, 20 bytes each plus the lengths of "new-fast-box", "old-slow-box", "server" and "sysadmins-box" - "example.com" in the query section is quoted here and doesn't need to be counted again.
- 75 bytes for 3 NS RR's, 15 bytes each plus the lengths of "server", "ns1.ip-provider.net." and "ns2" - again, "ip-provider.net." is quoted and only needs to be counted once.
- 120 bytes for the 6 address records (assuming IPv4 only) mentioned by the SRV and NS RR's.

IANA Considerations

The IANA has assigned RR type value 33 to the SRV RR. No other IANA services are required by this document.

Changes from [RFC 2052](#)

This document obsoletes [RFC 2052](#). The major change from that previous, experimental, version of this specification is that now the protocol and service labels are prepended with an underscore, to lower the probability of an accidental clash with a similar name used for unrelated purposes. Aside from that, changes are only intended to increase the clarity and completeness of the document.

Security Considerations

The authors believes this RR to not cause any new security problems. Some problems become more visible, though.

- The ability to specify ports on a fine-grained basis obviously changes how a router can filter packets. It becomes impossible to block internal clients from accessing specific external services, slightly harder to block internal users from running unauthorized services, and more important for the router operations and DNS operations personnel to cooperate.
- There is no way a site can keep its hosts from being referenced as servers (as, indeed, some sites become unwilling secondary MXes today). This could lead to denial of service.
- With SRV, DNS spoofers can supply false port numbers, as well as

host names and addresses. Because this vulnerability exists already, with names and addresses, this is not a new vulnerability, merely a slightly extended one, with little practical effect.

References

STD 2: Reynolds, J., Postel, J., "Assigned Numbers", STD 2, [RFC 1700](#), October 1994 (as currently updated by the IANA).

[RFC 1034](#): Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.

[RFC 1035](#): Mockapetris, P., "Domain names - Implementation and Specification", STD 13, [RFC 1035](#), November 1987.

[RFC 974](#): Partridge, C., "Mail routing and the domain system", [RFC 974](#), January 1986.

[BCP 14](#): Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC 2181](#): Elz, R., Bush, R., "Clarifications to the DNS Specification", [RFC 2181](#), July 1997

[RFC 2219](#): Hamilton, M., Wright, R., "Use of DNS Aliases for Network Services", [BCP 17](#), [RFC 2219](#), October 1997

Acknowledgements

The algorithm used to select from the weighted SRV RRs of equal priority is adapted from one supplied by Dan Bernstein.

Authors' Addresses

Arnt Gulbrandsen
Troll Tech
Postboks 6133 Etterstad
N-0602 Oslo, Norway
+47 22646966
<agulbra@troll.no>

Paul Vixie
Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
+1 650 779 7001
<paul@vix.com>

