

Network Working Group  
Request for Comments: 2052  
Updates: 1035, 1183  
Category: Experimental

A. Gulbrandsen  
Troll Technologies  
P. Vixie  
Vixie Enterprises  
October 1996

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

#### Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

#### 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, 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.

#### Introductory example

When a SRV-cognizant web-browser wants to retrieve

http://www.asdf.com/

it does a lookup of

http.tcp.www.asdf.com

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

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 or locally.

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

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.

#### Class

Standard DNS meaning.

#### 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 pseudorandom order. The range is 0-65535.

**Weight**

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. The range of this number is 1-65535. 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).

**Port**

The port on this target host of this service. The range is 0-65535. 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 A records for this name. Implementors are urged, but not required, to return the A record(s) in the Additional Data section. Name compression is to be used for this field.

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

**Domain administrator advice**

Asking everyone to update their telnet (for example) clients when the first internet site adds a SRV RR for Telnet/TCP is futile (even if desirable). Therefore SRV will have to coexist with A record lookups for a long time, and DNS administrators should try to provide A 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 A RRs 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 behaviour is.
- Where one service is provided by several hosts, one can either provide A 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 A records.

- Hosts that are referenced by backup A records must use the port number specified in Assigned Numbers for the service.

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.asdf.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 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, and move it to the tail of the new list

For each element in the new list

    query the DNS for A RR's for the Target or use any RR's found in the Additional Data section of the earlier SRV query.

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

else if the service desired is SMTP

    skip to RFC 974 (MX).

else

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

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

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.