Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 January 2026

S. Kerr
IBM

20 July 2025

Authoritative DNS Server-Side Answer Generation
draft-kerr-auth-dns-server-ans-gen-00

Abstract

The traditional model for DNS is that authoritative servers would read DNS data from static zone files and use that to answer DNS queries. Modern DNS servers do not act in this way, and their answers are created dynamically - either periodically or at query time.

This document presents a taxonomy breaking down the most common and useful methods used to customize responses on the authortiative side, as well as a survey of implementations by current large authoritative operators.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 21 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Many DNS authoritative servers have features to provide a different answer for the same DNS query.

For example, one user querying the AAAA record for www.example.org might get 2001:db8:1111::42 and another user querying the same record might get 2001:db8:9999::11. The DNS recursive resolver for each will cache the record normally, and the user's software will connect to different servers at different locations. The goal is to give the user a "better" answer, by whatever metric the operator is using; this can be latency, cost, adherence to regulations, or anything else.

DNS was not designed with this kind of responses in mind, and there are no standard ways to represent this data. This makes inter-operable systems difficult or impossible to build.

2. Taxonomy

While there are a wide variety of ways that DNS authoritative servers modify or generate answers, most of those fall into a few categories. This section presents a taxonomy of such answers.

For each type of generated or modified answer we briefly describe:

* Configuration, which is roughly static

* Data, which is relatively dynamic

2.1. CNAME-at-Apex (ALIAS)

In the DNS protocol the CNAME record type is special, and used to indicate that a name can be resolved at some other name. This is not possible at the apex of a zone (the "top" of a zone), because a CNAME cannot appear at the same name as another type, and both the SOA and NS types must appear at the zone apex.

A common solution is to do the equivalent of CNAME referral processing but on the authoritative side. This is often called an ALIAS record, but other names exist. Basically the authoritative server will do a lookup at the ALIAS target, acting as a client, and return the record there. This always includes address types (A and AAAA), although the details of exactly how this works will vary.

Configuration: The record is usually configured with the DNS name of the target, although other configuration might exist, for example specified behavior on failure or TTL behavior.

Data: The "data" for this type of response is the result of a DNS query.

2.2. GeoIP

The idea with GeoIP is to provide different answers depending on the IP address of the client, with the hope that the IP address provides a good hint at the physical location of the end user or system. This is not always true, but is good enough for many purposes.

Additionally, ECS (EDNS Client Subnet) (https://www.rfc-editor.org/rfc/rfc7871) is often used to provide more information about the IP address of the end user or system, which is useful for resolvers that are using IP addresses in different physical locations from the users.

Typical reasons to provide different answers include:

* Better performance for users by sending them to servers closer to where they are located.

* Attempting to meet legal requirements, such as blocking access to certain regions (for example due to gambling restrictions) or ensuring that traffic stays within specific borders (for example to avoid tariffs).

Sometimes instead of using the IP address directly, these may be mapped to an ASN (BGP Autonomous System Number). In a similar way that IP is used as a proxy for geographical location, ASN can be used as a proxy for the originating ISP (Internet Service Provider).

Configuration: Configuration will vary, but usually involves specifying a country or region associated with answers. It may also involve distance, for example choosing the answer closest to data centers of a service.

Data: The mapping of IP to geographic location (or other type of location) needs to be updated as IP addresses are transferred between organizations or used by an organization in different sites.

2.3. Service Availability

Within a single site some sort of load balancer can perform health checks on back-end servers and route traffic depending on their availability. For doing this across the Internet this is often done using the DNS; this technique is sometimes called Global Server Load Balancing (GLSB), DNS Server Load Balancing (DSLB) or just DNS Load Balancing (DLB).

The details of what the authoritative server will return vary. For example, the simplest case would be a service that returns one IP address when a health check is successful, and a different IP address otherwise. A more complicated case could be with pools of IP addresses, and the answer generated based on regional load metrics.

Configuration: In many cases the configuration can be quite simple, such as when a standby server is defined. If load or other factors are used to define availability, then the configuration can be arbitrarily complicated.

Data: The data about availability needs to be updated frequently, and may be something that organizations keep private. Cloud providers often have a rich set of availability information provided automatically by hosted services they operate.

2.4. Traffic Steering

The goal of traffic steering is to provide low latency, optimize for cost, or other network-related characteristics of a service.

Traffic steering is similar to service availability, but is used to alter responses based on the conditions of the network rather than conditions of the services themselves. Real-User Measurements (RUM) fall under this category; RUM is the ideal state, but the costs of obtaining, maintaining, and distributing RUM may be more than the benefits provided.

In contrast with availability checks, traffic steering is not about the status of systems under the control of the service operator, but about the Internet or possibly large systems not under their direct control, like Content-Delivery Networks (CDN) or transit network providers. While traffic steering requires more measurement points, and the networks being measured are not under direct control of the system operators, the modifications of DNS responses are similar.

Configuration: The source of information about the network is configured, as well as they type of transformation that is done on answers. The pool of answers might be configured, or it may be generated data as well.

Data: The measurements of network characteristics is typically proprietary information. For some services simple checks from a few points will be enough, although even in those cases organizations will often pay for distributed measurement networks. In other cases measurements will be provided by vendors, often the DNS provider themselves.

2.5. Everything Else

Any server-side processing imaginable is done somewhere. This includes anything from delivering weather reports to TXT-based traceroute information. For example:

* Allow different types of server-side generation or modification to be used together.

* Weight responses, to split load unevenly.

* Map contents of one zone or record to another at runtime.

* Generate IP6.ARPA and IN-ADDR.ARPA responses based on AAAA or A records.

Configuration: Anything from no configuration at all to Lua scripts as DNS records.

Data: Everything is data if you try hard enough.

3. Other Considerations

3.1. TTL Trade-Offs

The DNS includes a Time-To-Live (TTL) value which specifies the maximum time that a system is allowed to use a value. Systems do not have to use the value for that long - often a maximum of 1 day or even shorter is used. There is no minimum TTL in DNS, although many systems reject TTL that are too short.

A long TTL reduces load throughout all components in the DNS, and also reduces the impact of outages or other problems looking up DNS records. However, systems that modify or generate answers may need to respond to rapidly-changing conditions. CDN often use TTL of 1 minute or less.

4. Survey of Some DNS Vendors

In order to try to understand what the proprietary capabilities are in this space, a survey of some current authoritative DNS vendors was undertaken. This was not chosen based on science or data, although the largest players in this space are represented.

4.1. Enterprise Managed DNS Vendors

4.1.1. IBM NS1 Connect

IBM NS1 Connect has a couple of custom record types, ALIAS and REDIRECT.

They have a special zone type called a _linked zone_, which maps the entire contents of one zone to another zone name. It does this without DNS lookups, so is limited to IBM NS1 Connect customers.

They have a special record type called a _linked record_, which is similar to a linked zone, but for a single record.

They have "filter chains" which provide special processing based on:

* Geographic (country, region, latitude/longitude)
* Network (ASN or IP prefix)
* Health check-based (load, up/down)
* Traffic-steering (shuffle in various flavors, priority, pick N)
* "Pulsar" (RUM-based)

Any number of filter chains can be used on a record to answer queries. Results are always on the same DNS type and name, and result codes cannot be modified.

4.1.2. UltraDNS

UltraDNS has a couple of custom record types, Apex Alias and Web Forwarding.

UltraDNS has something called a Pool, which combines specific records which work with specific checks. These are:

* Resource Distribution (RD): a group of A/AAAA records
* SiteBacker (SB): A or CNAME that monitors backend service with a redirect to another service on outage
* Traffic Controller (TC): SB as a Global Server Load Balancing solution
* Simple Load Balancing (SLB): A/AAAA records, an HTTP monitor, and a backup
* Simple Failover (SF): simple monitoring with failover

* Directional (DIR): using GeoIP to determine response

4.2. Consumer Managed DNS Vendors

4.2.1. DNS Made Easy

DNS Made Easy has a couple of custom record types, ANAME and HTTP redirect types.

They have a monitoring/failover solution.

Their upscale product, Constellix, also supports weighted round-robin load balancing and GeoDNS (including custom IP rules).

4.2.2. dnsimple

dnsimple has a few proprietary types, ALIAS, POOL, and URL.

The POOL type picks a single CNAME from a set of CNAME.

4.3. Cloud Vendors

4.3.1. AWS Route 53

Route 53 has an alias record types.

You can specify an address record just by the service, rather than an IP address.

Route 53 has "routing" in the DNS, based on:

* Simple; NS are only supported via this policy
* Geoproximity; either AWS region/local zone group (with bias setting), or latitude/longitude
* Latency; includes optional health check
* IP address (of the end user); CIDR /24 (IPv4) or /32 (IPv6); "CIDR collections"
* Geolocation; IP to location mapping; if no default -> "no answer" on miss (NXDOMAIN? NODATA?); EDNS0 supported
* Failover; based on AWS health checkers
* Multivalue; health check for backend services

* Weighted routing; distribute load proportionally

4.3.2. Azure DNS

Azure has an alias record, which updates automatically for Azure data. The alias record can point to several types of automatically configured data, including applications and Traffic Manager, but also CDN endpoints. It does not seem to be able to alias to non-Azure DNS names.

Traffic Manager does support tracking hosts outside of Azure, either by IP address or FQDN. It include "endpoint monitoring", which is looking at HTTP/HTTPS/TCP endpoints. It also includes RUM.

4.3.3. Google Cloud DNS

Google Cloud DNS has a couple of custom types, ALIAS and IPSECVPNKEY.

There does not seem to be a way to provide custom responses.

4.4. CDN/Hosting Vendors

4.4.1. Akamai Edge DNS

Akamai has _alias zones_, which are a bit like ALIAS records but for the whole zone.

Akamai has _zone apex mapping_ via the AKAMAICDN record type, which is a bit like ALIAS records.

4.4.2. Cloudflare

Cloudflare supports _CNAME flattening_, which is similar to ALIAS records. The server will resolve the CNAME target and return it directly.

4.5. Registrars

4.5.1. GoDaddy Premium DNS (Akamai)

GoDaddy does not seem to have any special server-side record processing.

Appendix A. Acknowledgments

Jan Vcelak reviewed the text, correcting errors and providing additional information about advanced features of some vendors.

Author’s Address

Shane Kerr
IBM
Johan Huizingalaan 765
1066 VH Amsterdam
Netherlands
Email: shane.kerr@ibm.com