A Software Approach to Distributing Requests for DNS Service using GNU Zebra, ISC BIND 9 and FreeBSD *

Joe Abley
Internet Systems Consortium, Inc.
950 Charter Street, Redwood City, CA 94063, USA
jabley@isc.org

Abstract

This paper describes an approach for deploying authoritative name servers using a cluster of hosts, across which the load of client requests is distributed. DNS services deployed in this fashion enjoy high availability and are also able to scale to increasing request loads in a straightforward manner.

The approach described here does not employ any custom load-balancing appliances (e.g. devices commonly marketed as as "layer-four switches," "content switches" or "load-balancers"); instead the individual members of the cluster announce a service address to one or more gateway routers by participating in routing protocols to provide an intra-cluster anycast architecture.

The F Root Name Server is deployed using clusters built in this fashion, using FreeBSD [1], GNU Zebra [2] and ISC BIND 9 [3].

1 Design Goals

This paper describes a alternative to the deployment of individual authoritative DNS [4] servers which seeks to achieve the following:

1. Reliability: The DNS service should be highly available, and should hence survive both unplanned single-point events such as hardware failures, and also planned maintenance which might cause individual components to become unavailable.

2. Integrity: Software problems on individual hosts in the cluster should result in the corresponding hosts being automatically removed from service, so their impact on the service being provided is minimised.

3. Scalability: The DNS service should be capable of scaling to handle very high request loads without requiring very high performance on individual hosts. In combination with the reliability requirement above, upgrades to allow increasing request loads to be handled should be possible without making the service unavailable.

The architecture described in this paper could also be applied to other services. DNS, however, is ideally suited to this approach by virtue of its stateless (or stateful but short-lived) transactions; other protocols with different characteristics may not fare as well. See Section 6.4 for more discussion.

The use of this technique for load-balancing authoritative DNS service has been widely tested in the field by ISC in the deployment of the F root name server.

2 General Approach

One or more routers are connected to a common, multi-access network such as a single VLAN on an Ethernet switch. Two or more hosts are also connected to this common subnet (see Figure 1).

Individual interfaces on hosts and routers are configured with globally-unique addresses, such that each interface on each component can be addressed unambiguously by other devices on the Internet. These unicast addresses are used for management and other non-service traffic.

A service is associated with an IPv4 or IPv6 service address. This address is different to any of the addresses configured on any of the router or host interfaces described above.

Each host is configured with a dedicated software loopback interface, on which just the service address is configured.

The routers and hosts are all configured to participate in an OSPF [6] backbone area, through which hosts signal reachability of the service loopback address using OSPF Link State Advertisements (LSAs).

Requests from clients are sent to the service address. Each request will be delivered to a single host for processing, and hence the DNS software on each host should be configured to listen for requests on that service address. Responses from

*This paper contains material originally published in ISC-TN-2004-1 [5]. That document contains configuration examples which have been omitted from this paper in the interests of brevity.

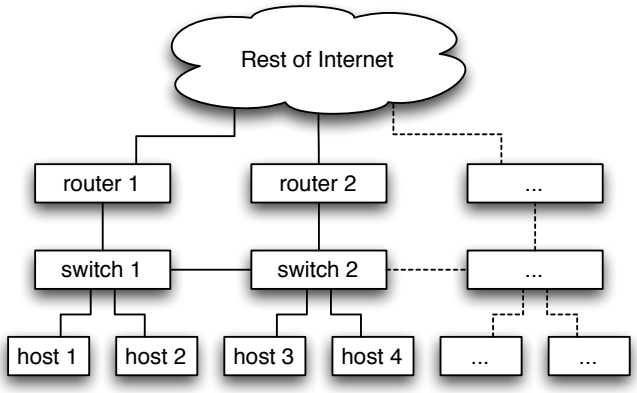

Figure 1: Components of the Anycast Cluster

the host being sent back to a client are generated with the source address set to the service address.

Non-service traffic sent from outside the cluster to individual hosts (e.g. management traffic) is directed to the globally-unique, unicast address of each host. Corresponding response traffic originated from the host is sourced from that unicast address, and not the service address.

Each router provides routes to the rest of the Internet, probably including a gateway of last resort.

Each router now enjoys multiple paths to the service address, learnt through OSPF. The routers are configured to route packets destined for the service address using some appropriate heuristic to obtain the desired load-balancing (see Section 3.2).

3 Routing Details

3.1 General

In the context of OSPF, each host is a router advertising availability of a link connected to a distant common subnet which covers the service address. The SPF algorithm when run on the two gateway routers (labelled "router 1," "router 2," etc. in Figure 1) yields multiple, equal-cost candidate routes for the service address; these are described as Equal Cost Multi-Path (ECMP) routes in the OSPF specification.

The host route for the service address is anycast within the routing system of the cluster. The service address may appear to be a unicast system as viewed from the Internet as a whole, or the cluster may be one of many providing the same service as part of a distributed, Internet-wide anycast deployment [7].

3.2 Gateway Routers

It is a requirement that the gateway routers used in the cluster are able to make use of all available ECMP routes. Although supported by OSPF, availability of ECMP support may be limited by the routing architecture of the system on which OSPF is implemented: for example, some operating systems cannot accommodate more than one route to a single destination.

3.2.1 Stateless Transactions

For DNS requests carried over UDP [8] with no fragmentation, an entire transaction consists of a single-packet request followed by a single-packet response. The protocol is stateless and it is acceptable for subsequent packets sent to the service address by the same client to be delivered to different hosts. Any route selection algorithm on the gateway routers will provide an distribution of request traffic which allows DNS transactions to proceed.

3.2.2 Stateful Transactions

Flow Hashing DNS transactions which require multiple packets to be exchanged between client and server cannot accommodate successive packets being delivered to different hosts. This is the case, for example, for DNS transactions which are performed over TCP [9], since state is maintained on an individual host between packets for a single transaction.

For DNS transactions carried over TCP it is necessary to associate a single route from the set of candidate ECMP routes with all packets associated with a single transaction (a "flow"). Cisco routers using Cisco Express Forwarding (CEF) [10] are able to associate a hash of (source, destination) internet- and transport-layer addresses with a single route, which satisfies this requirement.

CEF's route selection algorithm is stateless and deterministic for a stable set of ECMP routes. In general, however, a change in the number or ordering of those routes may cause the route selected for a particular (source, destination) hash to change. This fragility should be considered when gauging whether this load distribution approach is appropriate to particular protocols. See Section 6 for further discussion.

Juniper routers can be configured with `load-balance per-packet` which, on routers with the Internet Processor II ASIC, provides similar forwarding behaviour to that described for CEF.

Load distribution between hosts using flow-hashing forwarding algorithms will tend to be uneven in terms of traffic presented. The degenerate, illustrative case of this is a single host sending a stream of requests to the service address from a consistent source port: requests will always be answered by a single host, since the (source, destination) hash will always select a single route. See also Section 6.1.

Upstream Router Considerations Different gateway routers will, in general, map the same (source, destination) hashes to different candidate routes, since the ordering of ECMP routes in each router will be different. In order to ensure that packets from a single flow are routed to a single host for processing it is necessary that all the packets enter the cluster via the same gateway router.

This imposes the requirement on upstream routers that the route to the service address be stable for a single flow, which can be accommodated, for example, by having those routers use flow hashing forwarding algorithms with ECMP routes, or routing protocols which explicitly deny ECMP such as standard BGP [11] without multipath extensions.

3.3 Hosts

Hosts do not share the forwarding requirements described in Section 3.2; datagrams sourced from the hosts are unambiguously addressed to the unicast addresses of clients, and any equal-cost route diversity in the path back to those clients will inevitably converge on a single device.

The availability of the DNS service on a particular host is signalled to the gateway routers by issuing an LSA through which the service address is reachable. Correspondingly, the non-availability of the DNS service is signalled by issuing an LSA which withdraws the route, as if the link between the "host" router and the service address had been severed.

The straightforward requirements for the host's routing ability are easily satisfied by unix-based OSPF implementations such as the one included in GNU Zebra.

4 Host Operating System Considerations

The operating systems used on the hosts support cloneable Loopback interfaces, and a dedicated software loopback address is created and configured with the service address. The availability or non-availability of the DNS service is then signalled to the OSPF process on the host by simply raising or lowering the interface (e.g. `ifconfig lo1 up`, `ifconfig lo1 down`).

Nodes of the F root name server are hosted on FreeBSD 4 (with support for cloneable loopback interfaces added) and FreeBSD 5 (which supports cloneable loopback interfaces as released).

5 Name Server Considerations

5.1 General Configuration

The DNS software running on the host is configured to:

1. Bind to the service address, configured on a loopback address on the local host, and listen for client requests;
2. Bind to the unicast address of the host's interface on the cluster subnet at all other times (e.g. to perform zone transfers).

5.2 Zone Transfers

Special consideration may be required to accommodate zone transfers from master services which provide access control based on source address.

In the case where a master server insists that a slave server's service address be used to source a zone transfer request, it will frequently be the case that traffic from the master server will be delivered to a different host from the host which originated the request. Zone transfers in this case will time out.

In an N-host cluster (assuming random distribution of flow hashes on the gateway routers) one zone transfer request in N can be expected to succeed from any host. To help ensure that the transferred data is available quickly to all hosts in the cluster, each host can be configured to attempt zone transfers from the master server and also from each other.

Some slave name servers may be configured to perform zone transfers from a mixture of master servers which accept zone transfer requests sourced only from the service address, and others which accept zone transfer requests sourced from individual hosts' unicast addresses. To facilitate this ISC BIND 8 and ISC BIND 9 will attempt zone transfers from their configured `transfer-source` address first, and will retry using an unbound socket if the first attempt fails. Using an unbound socket has the effect in this case of sourcing the zone transfer request from the local unicast address.

Where zone transfers are authenticated using methods which do not rely on source address checking (e.g. using TSIG [12]), or where zone transfers are not authenticated, zone transfer requests may be sourced from hosts' unicast addresses and the concerns described here are avoided.

5.3 Self-Consistency

ISC BIND 9 is designed in such a way that anomalous run-time conditions which might lead to defective behaviour cause the named process to terminate. This facilitates a straightforward automatic control mechanism to allow the advertisement of the service to be tightly coupled to the availability of the DNS software, using a wrapper script.

5.4 Troubleshooting

For troubleshooting purposes, it is sometimes useful to be able to identify the individual host in a cluster which is servicing a particular client. ISC BIND 9 will answer a query for a TXT record in the CHAOS class for `HOSTNAME.BIND` with RDATA coresponding to the local hostname (see Figure 2).

```
$ dig +short @F.ROOT-SERVERS.NET \
>   HOSTNAME.BIND CH TXT
"sfo2a.f.root-servers.org"
$
```

Figure 2: Troubleshooting with `HOSTNAME.BIND`

Support for the `HOSTNAME.BIND` query is included in ISC BIND 8 (all versions) and ISC BIND 9 (from version 9.3).

6 Limitations

6.1 Load Balancing

The load distribution scheme is limited by the ECMP route selection algorithm used in the gateway routers. More sophisticated load-balancing algorithms are supported by dedicated load-balancing appliances (e.g. "least loaded server," "least recently used server").

6.2 Service Monitoring

The integrity of the service is sustained by rigourous checks on configuration files prior to the service starting, coupled with the self-consistency checks in ISC BIND 9 which cause the process to exit if they fail. External consistency checks are possible based on transactions against individual hosts' unicast address, and against the service address in general. Testing transactions directed at the service address on specific hosts is not straightforward due to the anycast routing of the service address.

6.3 Host Identification

Although the `HOSTNAME.BIND` lookup illustrated in Figure 2 provides some degree of diagnostic support to troubleshooting, it is not possible in general to determine the precise host which served a response to a particular DNS query without resorting to packet capture or expensive query logging on every host which is able to provide service to a client. This is an issue common to all load balancing techniques, and not just the one described in this paper.

At the time of writing work is underway to extend EDNS to provide a method to identify individual hosts from transaction response data.

6.4 Applicability to Other Protocols

DNS request and response traffic has the characteristic that transactions tend to be short-lived, and are executed rapidly. Other protocols whose transactions are longer lived may suffer from changing flow hash results as the ECMP route set changes; this might happen, for example, following LSAs sent from a host whose DNS service is taken down for maintenance, due to a failure in an individual host or because the cluster is being enlarged.

The deployment of the F root name server using the technique described in this paper provides data to suggest that this technique is effective for DNS service; those wishing to deploy protocols whose transactions are substantially different in nature to DNS are advised to test thoroughly.

7 Acknowledgements

Much of the material in this technical note is based on work done by Paul Vixie, Stephen Stuart and Peter Losher in providing service for the F root name server.

References

[1] The FreeBSD Project, `http://www.freebsd.org/`.

[2] GNU Zebra, `http://www.zebra.org/`.

[3] ISC BIND 9, `http://www.isc.org/sw/BIND/`.

[4] P. Mockapetris, *Domain Names – Implementation and Specification*, STD 13, RFC 1035 (1987).

[5] J. Abley, *A Software Approach to Distributing Requests for DNS Service using GNU Zebra, ISC BIND 9 and FreeBSD*, ISC Technical Note Series, ISC-TN-2004-1, `http://www.isc.org/pubs/tn/isc-tn-2004-1.html` (2004).

[6] J. Moy, *OSPF Version 2*, STD 54, RFC 2328 (1998).

[7] J. Abley, *Hierarchical Anycast for Global Service Distribution*, ISC Technical Note Series, ISC-TN-2003-1, `http://www.isc.org/pubs/tn/isc-tn-2003-1.html` (2003).

[8] J. Postel, *User Datagram Protocol*, STD 6, RFC 768 (1980).

[9] J. Postel, *Transmission Control Protocol*, STD 7, RFC 793 (1981).

[10] Cisco Systems, "Cisco Express Forwarding (CEF)", `http://www.cisco.com/warp/public/cc/pd/iosw/iore/tech/cef_wp.html` (2002).

[11] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4)*, RFC 1771 (1995).

[12] P. Vixie, O. Gudmundsson, D. Eastlake and B. Wellington, *Secret Key Transaction Authentication for DNS (TSIG)*, RFC 2845 (2000).