Internet Engineering Task Force   Alain Durand
INTERNET-DRAFT   SUN Microsystem
Oct 2, 2001
Expires Apr. 3, 2002

IPv6 DNS lookup proxy

draft-durand-dns-proxy-00.txt

Status of this memo

Abstract

This document describe a DNS lookup proxy to enable IPv6 only resolver to query data on IPv4 only server.

1. Introduction

As analyzed in [DNSOPreq], the operation of DNS in a mixed environment IPv4 and IPv6 require Some bridging to happen to enable an IPv6 only system to query data on an IPv4 only server and vice-versa. However, such bridges do not need to be symmetrical, that is, it is OK, for the sake of efficiency, to design two different systems, one for each case. This document presents a scalable solution to enable IPv6 only systems to query IPv4 only servers. The case of an IPv4 only system querying an IPv6 only server is not discussed here.

2. Recursive vs non recursive fallback system

One of the approach suggested to solve the bridging problem was to use some kind of dual stack, general forwarder that will resolve the queries on behalf of the IPv6 only resolver. An IPv6 only resolver could either delegate all its queries to this forwarder or only use it in last resort mode, when IPv4 transport is needed to reach the desired DNS server.

In the first mode of operation, the general forwarder may face massive scaling issues. In the second mode of operation, the forwarder will still have to operate in recursive mode because the information gathered previously by the IPv6 only resolver in its attempt to resolve the name will be lost. It is feared that such general forwarder will also face serious scaling issues once the IPv6 traffic will increase.

The lookup-proxy design is based upon the following observation: When a resolver is following a chain of referrals and cannot complete because it is referred to an address it lacks transport for, then it knows both the query and where to send it. It is just lacking transport. The solution presented here aims at bridging seamlessly the two transports by providing a new protocol that can send the tuple:

```
        {query, server}
```

to a proxy, have the proxy send the query on (directly) to the server, collect the response and return it to the resolver. The proxy will be non-recursive, and thereby much more scalable. Furthermore, the proxy does not (or should not) know much about DNS, it should only know enough to repack the query and response in IPv4 and IPv6 packets respectively.

3. Lookup proxy architecture

3.1 An IPv6 anycast prefix

As an IPv6 address is much larger than an IPv4 address, it is possible to embed an IPv4 address within an IPv6 address. Proposal like [6to4] or [isatap] use this property to embed IPv4 tunnel endpoint within IPv6 addresses.

This document suggest to use a well know, globally routable prefix P as an anycast DNS lookup proxy prefix. The prefix length of P MUST be shorter than 96 and SHOULD be small enough not to be filtered in common BGP announcement.

A set of DNS lookup proxies MUST advertise this anycast prefix and MUST intercept any IPv6 packet whose destination address is of the form P::a.b.c.d (a.b.c.d represent the 32 bits of an IPv4 address) and UPD destination port is 53.

3.2 DNS lookup proxy behavior

A DNS lookup proxy SHOULD check the payload to make sure it really is a valid DNS query and then MUST forward it in a new IPv4 packet.

The source address of this new packet is one of the proxy IPv4 addresses. The destination address is taken from the 32 lowest bits of the destination address of the incoming IPv6 packet. The transport protocol MUST be set to UDP and destination port to 53. The payload of the new IPv4 packet MUST be directly copied from the one in the IPv6 packet.

3.3 Fragmentation and MTU

Simple UDP DNS queries and answers are expected to fit within 512 bytes, fragmentation and MTU are not an issue for them. However, queries using EDNS 0 or falling back to TCP may have a larger payload. For DNS connections using TCP, MTU is not an issue, as TCP will adapt the correct MTU in each connection on both side of the proxy. Using EDNS 0, the client may specify a large packet size than 512. As an IPv6 header is longer than an IPv4 header (with no options), this mechanism will not results in fragmented UDP packets.

However, if the DNS communication results in exchanging more than one packet, there is a theoretical chance that different packets will go through different proxies, defeating the mechanism. It is expected that the routing system will be stable enough to prevent this case to happen in reality.

3.4 Mapping

A DNS lookup proxy MUST maintain some kind of mapping between the incoming IPv6 query and the outgoing IPv4 packet so that when the answer will come back from the IPv4 DNS server, it will know where to sent it to in IPv6 land.

A DNS lookup proxy MUST implement some time-outs on those mappings to do garbage collection.

3.5 Caching

A DNS lookup proxy MAY implement positive and/or negative caching technique to improve efficiency.

In the case of positive caching, the proxy MUST honor the TTL provided in the DNS answer; the proxy MAY use a smaller TTL than it received, but MUST NOT cache the answer beyond the period specified by the TTL.

3.5 rate limitation

A DNS lookup proxy may also impose some rate limitation measure on packet they sent to the same address, either IPv4 or IPv6, to lower the impact of potential DOS attack inherent with any public proxy.

4. Converting IPv4 referrals into IPv6 referrals

When an IPv6 only resolver is following a chain of referrals and cannot complete because it is referred only to IPv4 addresses, it SHOULD automatically derived an IPv6 addresses by padding the IPv4 addresses to the prefix P and send the DNS queries to those addresses.

5. Scaling issues

Using an anycast prefix P will allow to use as many proxy as necessary, thus this mechanism has very good scaling properties.

6. Anycast issues

IPv6 architecture requires the anycast addresses MUST NOT be used as source addresses. Thus, when returning the DNS answer, the proxy MUST replace the anycast address by one of its unicast address with the appropriate scope. Also, the IPv6 DNS resolver MUST not check the source address of packets returning from the proxy.

7. Security consideration

Any public proxy is inherently a source of DOS attack. Rate limiting packet emission as suggested in 3.5 is expected to lower the risks.

8. Author address

Alain Durand

SUN Microsystems, Inc
901 San Antonio Road
MPK17-202
Palo Alto, CA 94303-4900
USA
Mail: Alain.Durand@sun.com

9. References

[DNSOPreq] draft-ietf-ngtrans-dns-ops-req-02.txt

10. Acknowledgment

The author whishes to acknowledge the input of Johan Ihren and
Akira Kato.