IVIPTR: Resource Record for DNS
Draft-Tariq-DNSOP-IVIPTR-00

Tariq Saraj
Muhammad Yousaf
Amir Qayyum
Introduction

• A new DNS Resource Record i.e. IVIPTR
• Resolves IPv4 address to IPv6 and vice versa
The Current DNS Standard does not support to resolve:

- IPv4 address to IPv6 address
- IPv6 address to IPv4 address

For example:
- When querying AAAA of a resource when IPv4 address is known
- The response code (RCODE) for such query is usually ‘Non-Existent Domain (3)’
Problem in Practice

www.google.com IP Address

RCode: Non-Exiting Domain
Resolving through current DNS Standard

• When:
  • IPv4 address is known and one wants to resolve it to IPv6.

Step-01

Stub Resolver

Domain name for IPv4?
Response: example.com
AAAA for example.com?
Response: AAAA record

Recursive Server

Domain name for IPv4?
Response: example.com
AAAA for example.com?
Response: AAAA record

Step-02

Authoritative Server

Response: AAAA record
Step-04

Step-03
Resolving through current DNS Standard

• **The bottleneck:**
  • Not all the domain name labels map to both IPv4 and IPv6 addresses
  • Mostly, these days domains have different PTR records for corresponding AAAA and A record

• Thus, current DNS standard cannot fully be utilized to resolve IPv6 address against IPv4 address and vice versa
IVIPTR: Proposed Resource Record

Query Response: **AAAA in A**
Use Case: Firewall Rules Auto Updation

- Firewall rules normally configured for IPv4 traffic monitoring
- IPv6 is enabled in the same network for some application testing or need IPv6 rules to be configured automatically for each corresponding IPv4 rule
- Firewall automatically resolve IPv6 address if available for each of the configured IPv4 address using the proposed Resource Record (IVIPTR)
- Traffic monitoring rules for IPv6 will automatically be deployed against each resolved IPv4 address
- Currently, without the proposed IVIPTR RR, one must configure these rules manually
IVIPTR: The Proposed RR

• The IVIPTR RR has the following format:

```
<OWNER> <TTL> <CLASS> IVIPTR <IVI target>
```
IVIPTR: Representation in Reverse Zone File

; reverse zone file for example.com A record
1.0.168.192.IN-ADDR.APRPA. IN PTR a.foo.example.com.
1.0.168.192.IN-ADDR.ARPA. IN IVIPTR a.x6.foo.example.com.

; reverse zone file for example.com AAAA record
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA. IN PTR a.x6.foo.example.com.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA. IN IVIPTR a.x.foo.example.com.
Query Processing

- The query processing involves both standard reverse and forward lookups:
  - when the recursive name server receives a response for the IVIPTR RR against reverse lookup
  - After caching the response it will form a new query for forward lookup in such a way that
- Case-01: If the original query NAME field has A.IN-ADDR.ARPA. and TYPE field is IVIPTR
  - The NAME field of the new query should be RDATA resource
  - The TYPE field should be ‘AAAA’
- Case-02: If the original query NAME field has AAAA.IP6.ARPA. and TYPE field is IVIPTR
  - The NAME field of the new query should be RDATA resource
  - The TYPE field should be ‘A’
- Finally, the response against forward lookup is placed in the answer section of the original query and replied back to stub resolver
Questions ?