SCTE CABLE-TEC EXPO 23
CELEBRATING 40 YEARS

2023 Fall Technical Forum
SCTE • CABLELABS • NCTA

BGP Info Over DNS

A Technical Paper prepared for SCTE by

Tony Tauber
Engineering Fellow
Comcast

Charlie Helfinstine
Senior Principal Engineer
Comcast

Mark Feldman
Senior Principal Engineer
Comcast

Table of Contents

List of Figures

1. Introduction

Quickly knowing what is happening across one's network is a crucial, if challenging, task for the scale of any given modern network, particularly large service provider networks. Getting the BGP (Border Gateway Protocol) routing information about how a given node would reach a particular destination and comparing it against the same query gathered from various other vantage points can be transformational.

Logging into routers one at a time to execute the specific command for the given vendor and operating system is tedious, error-prone, and needs to be automated to reduce friction. Also, some groups can benefit from this information without needing general access to the devices. Generally, such access is, understandably, under tight control.

Our novel scheme exposes BGP routing information via DNS (Domain Name System) queries, which can be structured to query the information for a given router node or location. Network Operations staff can thus quickly gather details from around the network and synthesize it for the matter at hand. If a specific route or set of routes is of interest, one could retrieve that information regularly as a form of monitoring. The data sources inside a given operators network (BGP-speaking routers, Geolocation data stores, etc.) can push data to the DNS server(s) unidirectionally, meaning they are not subject to risk of compromise by outside users.

Furthermore, the delegated structure of the DNS namespace allows each network operator to tune the visibility of information according to their needs. Existing techniques for optimizing performance of DNS or other similar services (e.g., load-sharing, horizontal scaling) can be brought to bear.

Our paper and presentation will elaborate on this approach and explain the extensible nature of the technique. This technique of information sharing within and among Internet operators will make troubleshooting and problem resolution quicker and more accessible while maintaining infrastructure security and reliability.

2. The Problem – Sharing BGP (Border Gateway Protocol) information

To understand the need for gathering BGP information, we will give a brief overview of Internet routing.

2.1. Internet routing basics

The basic function of the Internet is to allow for the transfer of data. Consider a user retrieving information from a website. The user sends a request to the web server indicating what data they would like, and the server sends the data back in return. Each piece of this bi-directional flow of traffic between two endpoints has a "source" and a "destination". In the initial request from the user to the server, the user is the "source", and the server is the "destination". When the reply is returned, the server is the "source", and the user is the "destination".

The data for each the request and reply is encoded in one or more IP (Internet Protocol) packets (sometimes called "datagrams"). Between the user and the web server will be some number of intermediate "router" nodes which have some information about how to get to the destination.

Figure 1 shows an example national Internet backbone topology for purposes of illustration.

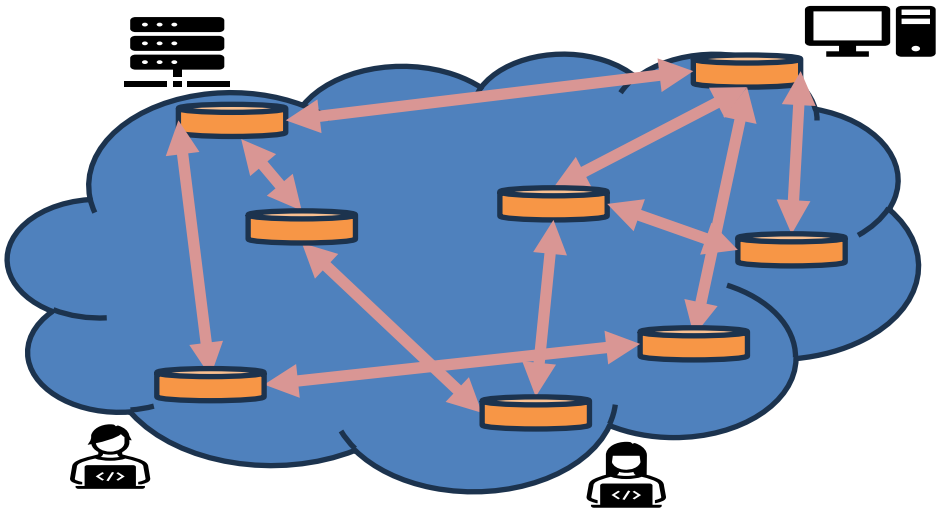

Figure 1 – Router network core with users and servers at the edges

When an IP packet arrives at a router for processing, the router examines the IP address in the "destination" field of the packet and searches its routing table for the "prefix" which is the best match to the destination address. This "answer" is known as the "Longest Prefix Match," sometimes abbreviated as "LPM" and the answer will be the "next-hop" to the destination. That "next-hop" is another router and this process continues until the last router before the destination (e.g., web server) is reached and the packet is finally delivered.

This routing and delivery process can be thought of much like a letter or package is delivered between cities or even countries using the postal network of postal stations and roads. Each "hop" along the way may only have general knowledge of how to reach a bit closer to the destination, getting more specific as things progress, e.g., Country to Country, State to State, City to City, City to Street, Street to Building.

2.2. Network Operations basics

In the course of network operations, it is often necessary to gather certain pieces of information about or from many different points and parts of a network.

The context could be within the infrastructure of a given operator, or could be from some other network(s) comprising the broader Internet. The reason often involves interrogating how a given router (and the user data traversing that router) would reach a given IP destination. Most often, this investigation happens in the case of troubleshooting a reported problem.

The traditional way of retrieving BGP routing information is via command-line interface to one or several network devices. Often it is necessary to gather the information quickly, and possibly submit it to further processing and cross-referencing.

Several problems arise from the need to use the command-line interface.

- The command structure and output format of these interfaces is non-standard and can vary by vendor, model, and software version. Figure 2 depicts an example of such CLI (Command-Line Interface) command output:

```
RP/0/RP0/CPU0:cs01.doraville.ga.ibone#show bgp 13.0.40.0/23
Mon Jul  3 03:27:04.080 UTC
BGP routing table entry for 13.0.40.0/23
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker          529060364   529060364
Last Modified: Jun 30 05:24:54.926 for 2d22h
Paths: (6 available, best #2)
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    96.110.43.82    96.110.42.210   68.86.94.162
  Path #1: Received by speaker 0
  Not advertised to any peer
  15133 1321 33631, (received & used)
    96.109.22.227 (metric 4296) from 68.86.1.0 (96.109.22.227)
      Origin IGP, metric 0, localpref 275, valid, internal
      Received Path ID 1, Local Path ID 0, version 0
      Community: 7922:403 7922:3020
      Originator: 96.109.22.227, Cluster list: 68.86.1.0, 96.109.22.50
```

Figure 2 – Example BGP information retrieved via CLI

- Typically, such interfaces are considered privileged access to the routers in question.
  Such access is protected by authentication and authorization schemes as well as other controls.
  The reasons for these measures are legitimate:
  - Non-public data may be available by such means.
  - The invocation of the command-line interface itself may have cause some extra processing load to the devices which runs a risk, however minor, of compromising the critical network functions of the network devices.
- The overhead of the authentication and authorization can inject considerable lag into the information gathering process.
- For situations inside a given organization where different levels of access exist, but many people may have use for network information, the ability to query and retrieve such information in a dynamic manner that does not compromise the integrity of the devices for information that is deemed non-sensitive.
- Furthermore, where information is acceptable to be publicly available, no standard scheme for structuring and providing such information exists.

3. Current Solutions for sharing BGP routing information

For retrieval of BGP data, two main solutions exist:

- Route Servers - These are typically router CLI (Command-Line Interface) means.
  As described earlier, they have no standard query language nor result format.
  The need to handle authentication and authorization by third parties without explicit permission makes for a sub-optimal experience. Example outputs from two separate route servers are illustrated in Figure 3.

```
RP/0/RSP0/CPU0:route-server.newyork.ny.ibone#show bgp 13.0.40.0/23
Sat Jul  1 04:55:11.784 utc
BGP routing table entry for 13.0.40.0/23
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker          633072720   633072720
Last Modified: May 14 12:03:50.037 for 2y06w
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  15133 1321 33631, (received & used)
    66.208.229.9 from 66.208.229.9 (96.109.22.227)
      Origin IGP, metric 0, localpref 275, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 0, version 633072720
      Community: 7922:403 7922:3020
      Originator: 96.109.22.227, Cluster list: 96.109.22.250, 96.109.22.50
RP/0/RSP0/CPU0:route-server.newyork.ny.ibone#
```

```
rviews@route-server.ip.att.net> show route 13.0.40.0/23

inet.0: 911801 destinations, 14587004 routes (911801 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

13.0.40.0/23       *[BGP/170] 3d 18:45:14, localpref 100, from 12.122.83.238
                      AS path: 7018 701 1321 33631 I, validation-state: unknown
                    > to 12.0.1.1 via em0.0
```

Figure 3 – Example BGP information details from Route Servers

- Looking Glasses - These are web servers which have some backend capability to query one or more network devices for the underlying information. (See Figure 4) Not only is the exact query language and output format not standard, but some back-end rate limits must be imposed to prevent adverse effects on the network devices themselves.

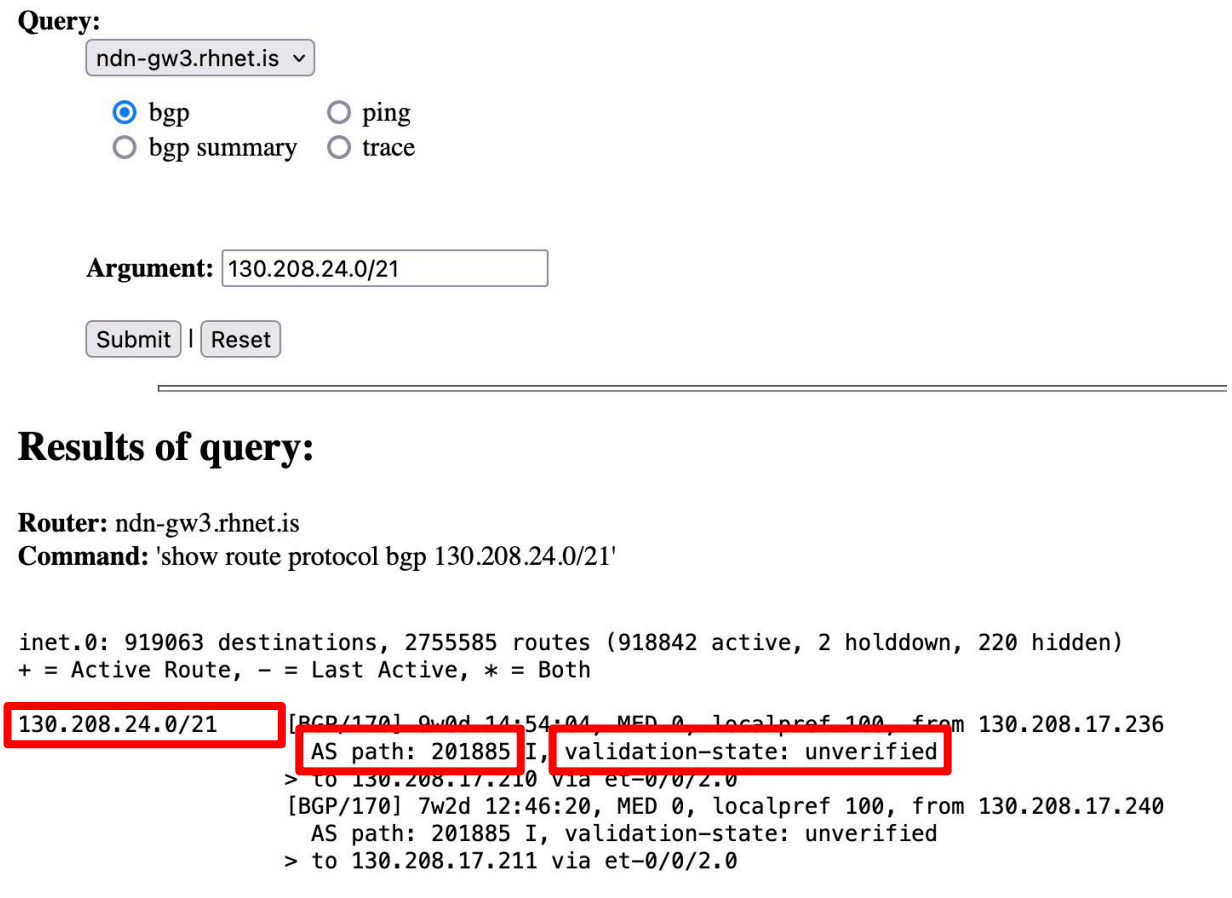

Figure 4 – Example Looking Glass Web Interface

4. DNS (Domain Name System) is standard and universally available

The DNS (Domain Name System) primarily provides a mapping between human-recognizable and "significant" names to IP addresses. The fundamental operation of a DNS "query" involves a resolver which recursively asks name servers, with presumed authority over the namespace, for an answer. These name servers respond to the best of their ability, which usually means undertaking the iterative function of asking another name server that has been delegated authority over a more specific part of the name space until one with a matching record is found. Figure 5 illustrates such a search.

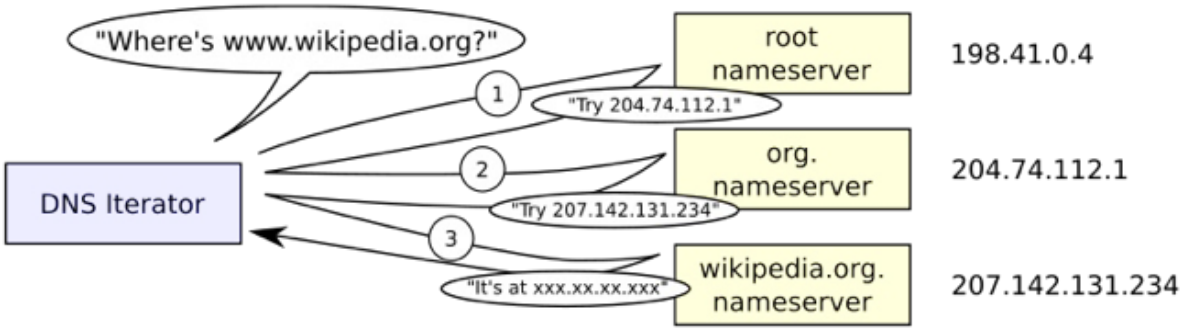

Figure 5 – DNS resolver iteratively queries more specific authorities to find an answer

These operations are done behind the scenes when using a web browser, for instance.

Using the "dig" command-line DNS utility allows for examining DNS details and can produce results such as those shown in Figure 6 below.

```
bash-3.2$ dig example.com

; <<>> DiG 9.10.6 <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5374
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            74249   IN      A       93.184.216.34

;; Query time: 84 msec
;; SERVER: 69.252.80.80#53(69.252.80.80)
;; WHEN: Thu Aug 17 10:03:08 EDT 2023
;; MSG SIZE  rcvd: 56
```

Figure 6 – Using "dig" utility to look up a DNS record

Note the command and arguments we supplied at the top.
In the middle is the answer returned by the DNS.

The DNS has many defined Resource Record (RR) types.

We will focus on one in particular.

RFC1035 section 3.3.14 defines the TXT (pronounced "text") records as:

TXT-DATA One or more <character-string>s.

TXT RRs are used to hold descriptive text.

The semantics of the text depends on the domain where it is found.

A very common current use relates to SPF (Sender Policy Framework) which is part of the global email system, but not germane to this discussion. We use it here to show an example of retrieving such a TXT record using the "dig" utility. Figure 7 shows the query at the top, the answer in the middle, and at the bottom we re-run the command adding the "+short" directive to give more terse output (just the "answer" without the verbose diagnostic information.)

```
bash-3.2$ dig example.com txt

; <<>> DiG 9.10.6 <<>> example.com txt
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3299
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      TXT

;; ANSWER SECTION:
example.com.            86400   IN      TXT     "wgyf8z8cgvm2qmxpnbnldrcltvk4xqfn
"
example.com.            86400   IN      TXT     "v=spf1 -all"

;; Query time: 65 msec
;; SERVER: 69.252.80.80#53(69.252.80.80)
;; WHEN: Wed Aug 23 16:46:56 EDT 2023
;; MSG SIZE  rcvd: 109

bash-3.2$ dig +short example.com txt
"v=spf1 -all"
"wgyf8z8cgvm2qmxpnbnldrcltvk4xqfn"
bash-3.2$
```

Figure 7 – Using "dig" utility to look up a DNS TXT record

5. Solution – BGP info served via DNS

Our proposed solution would be to use the DNS TXT ("text") record type to return BGP routing information when queried for a given IP address or range. Earlier we described the operational process of querying a router via CLI to retrieve information from or about the BGP table on that router. That process uses LPM to retrieve the relevant information. Our solution involves issuing a DNS query for an IP address or prefix to a server which performs LPM against a data store, which is similar to the BGP routing table and contains similar data which is then returned to the user.

The BGP table information can be exported from the router(s) to a server on a periodic or dynamic basis. An example of a periodic basis would be one where a daily process would:

1. Log into a router.
2. Use CLI commands to dump the routing table.
3. Parse the routing table contents into a format and data structure usable by the name server program.

An example of dynamic update might be:

1. A server runs a BGP software package.
    a. Examples: BIRD, Zebra, Quagga, openBGPd, FRR (Free Range Routing)
2. That server's BGP process is set to receive a BGP feed from router(s) in the network.
3. A process on the server converts the BGP information into a format and data structure usable by the name server program.

That server would then respond to queries which specify a given IP prefix (sometimes called a "route") and would then respond using the LPM mechanism to find the "best matching" prefix and then return the table information for that prefix in the form of one or more DNS TXT (text) record(s).

An example DNS query using the 'dig' DNS utility might be: ---
dig +short 1.1.4.1/24.get-ip-info.comcast.net
---

returns

'prefix: 1.0.4.0/22'
'as-path: 174 7545 2764 38803'
---
or alternately in JSON (JavaScript Object Notation) which is more suited to machine readability:
'{"prefix":"1.1.1.0/24", "as-path":"174 7545 2764 38803"}'

The solution benefits from the properties of the DNS:

5.1. Delegation

A key scaling feature of the DNS is achieved through delegation. No one set of servers could contain all the information about the Internet, so different pieces are delegated for administration at various levels. Returning to the analogy of postal delivery on the road system, a given country typically delegates some authority to states which, in turn, delegate more local authority to counties and/or cities and towns within their borders. Figure 8 illustrates part of the structure behind the delegated makeup of the DNS.

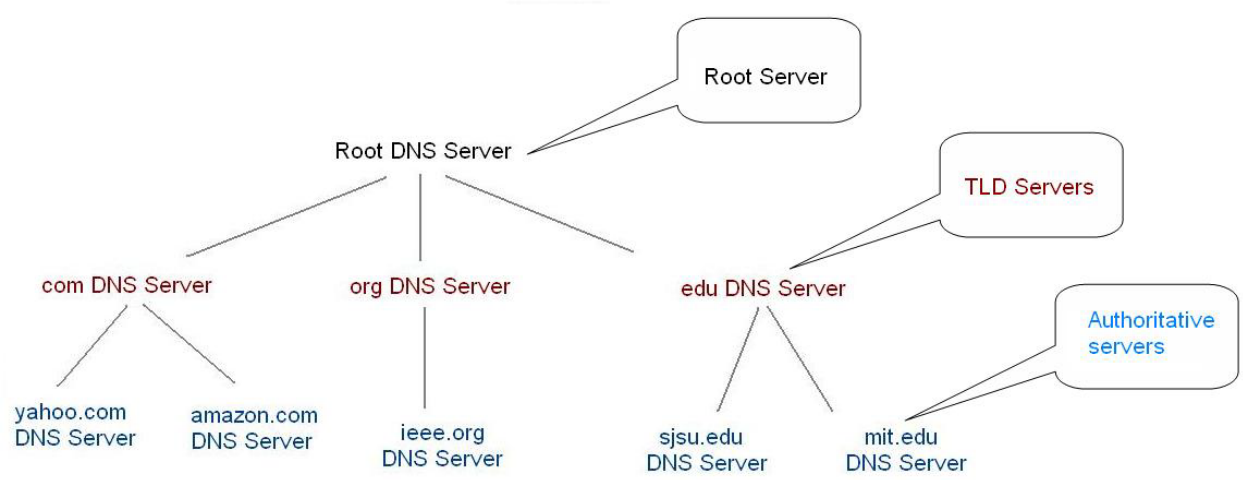

Figure 8 – The delegated structure of the DNS name space

5.2. Governance

As we introduced earlier, the most typical way the DNS is used is to provide mapping between a "name" and an IP address where one may retrieve information about that "name". In a related way, a "reverse" mapping between IP addresses and names is also possible within the DNS. As IP addresses are delegated from a "root" with increasing levels of specificity, those levels further delegate authority for their details.

RFC1035 section 3.5 describes the IN-ADDR.ARPA domain:

> The domain begins at IN-ADDR.ARPA and has a substructure which follows the Internet addressing structure.

and provides this example:

> A resolver which wanted to find the host name corresponding to Internet host address 10.0.0.6 would pursue a query of the form:
>
> QTYPE=PTR, QCLASS=IN, QNAME=6.0.0.10.IN-ADDR.ARPA, and would receive:
>
> 6.0.0.10.IN-ADDR.ARPA. PTR MULTICS.MIT.EDU.

RFC3596 describes the reverse mapping for IPv6 addresses under the IP6.arpa domain.

Figure 9 depicts the delegation structure for the in-addr.arpa domain which can be used to find "reverse" mapping of IP addresses to names.

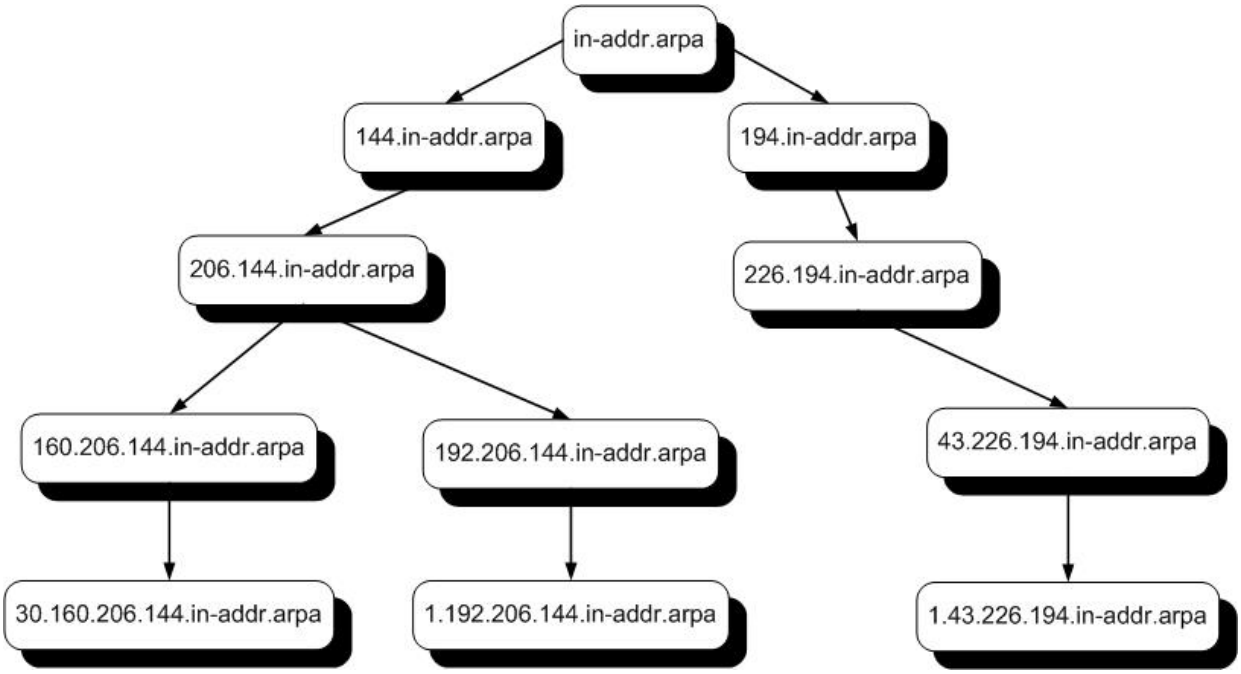

Figure 9 – The delegated structure of the DNS "reverse" name space for IPv4 addresses

The world of the Internet network operators is structured and reflected in BGP with each "network" as an AS (Autonomous System) having an ASN (Autonomous System Number) to uniquely identify it. Each AS, by definition:

> "…is a set of routers under a single technical administration ... [which] appears to other ASes to have a single coherent interior routing plan and presents a consistent picture of the destinations that are reachable through it."

[Source: RFC4271 (BGP - Border Gateway Protocol)]

In this sense, information for each ASN could be "located" in the DNS hierarchical namespace by reference to their ASN.

For example, the ICANN (Internet Corporation for Assigning Names and Numbers) and IANA (Internet Assigned Number Authority) could define a place in the DNS structure such as "network-info.arpa" where each delegation below (to the "left of") that domain could be delegated to the operator who has been assigned that ASN. Hence, Comcast may have a delegation for "7922.network-info.arpa" as well as each other ASNs which Comcast has been assigned. (The specific name "network-info.arpa" is used only for illustration.)

There is already an established delegation structure for AS numbers as well as DNS delegation so the "business rules" of such an invention would be straightforward to apply. Each AS operator would have a place to publish that could be found without any different "directory" than already exists today with ASNs being handled by the RIRs (Regional Internet Registries: ARIN, LACNIC, Afrinic, RIPE, APnic - and their delegates).

5.3. Structure of information

With this name space delegated among the various "AS (Autonomous System)" operators, the operators could further break it up as they pleased by location or network device. They could "publish" the structure of their part of the namespace using a different structured TXT record which would enumerate the delegation below the root of their portion of the namespace.

5.4. Caching

DNS includes the concept and ability of caching by intermediate nodes. The length of time such records are cached (including "zero") can be controlled by the authority for the namespace.

5.5. Performance

Many performance enhancements have been undertaken by the industry in the form of software and hardware optimizations.

6. More Details

BGP data is just one example of network information that could benefit from such treatment. There are other parts of network inventory, network configuration, or network state which could be published and retrieved by these means. The concept and approach are flexible and extensible.

One example might be a DNS responder that replies with the client's source address in TXT records:

```
dig +short TXT what-is-my-ip.comcast.net @labweek.dns.server.comcast.net
        3600 IN TXT "ip6:2001:558:1438:21:8c3:3f13:8cfb:9ef3"
```

Another example might reply with administrative information about the address:

- Security Zone
- Location
- Edge Router Name
- Owner

```
dig +short TXT 10.0.0.1/32.get-ip-info.comcast.net @labweek.dns.server.comcast.net
        3600 IN TXT "containerPrefix:10.0.0.1/16"
        3600 IN TXT "securityGroup:PURPLEZONE"
        3600 IN TXT "site:OmahaDC"
        3600 IN TXT "lat:41.2438° N"
        3600 IN TXT "long:99.4874° W"
        3600 IN TXT "owner:lls_ops@comcast.com"
```

Or embed the prefix *and* the router name to get the view from that specific router:

- AS-PATH
- BGP Community
- Date/Time stamp

```
dig +short TXT 10.0.0.1/32.cs01.pittsburgh.pa.ibone.get-asn-info.comcast.net
@labweek.dns.server.comcast.net
        3600 IN TXT "AS-path: 22909 64922"
        3600 IN TXT "community: 7922:416 7922:999 7922:2800 7922:2900 22909:7000"
```

```
3600 IN TXT "date:Thu Mar 16 13:23:59.985 UTC"
```

7. Conclusion

Network operations often require synthesizing many sources of data from many directions.

In the realm of BGP routing, interrogating the routing information held within a given router and across several vantage points is sometimes needed to perform diagnosis and analysis.

The traditional method for getting such information from routers is using an authenticated and authorized CLI . Sharing or retrieval of such information among and between different network operators is done using the CLI of "route servers" or a GUI (Graphical User Interface) provided by a "Looking Glass" web server which issues commands on the users' behalf to routers to retrieve such information.

Significant drawbacks of these approaches are they lack standardization and are cumbersome to use. They are not easily amenable to automation and pose some risks to the routing infrastructure where they get their data.

Our solution is to obtain BGP information proactively from one or more routers within an operator's network and make it available via DNS queries which perform Longest Prefix Matching to return the relevant information via TXT ("text") records.

This scheme benefits from several features of the DNS including hierarchy, delegation, caching, and performance enhancements. Moreover, it can be used to encode other types of network metadata as well such as geolocation, etc.

Abbreviations

AS	autonomous system
ASN	autonomous system number
BGP	border gateway protocol
CLI	command line interface
DNS	domain name system
GUI	graphical user interface
IANA	Internet Assigned Number Authority
ICANN	International Corporation for Assigned Names and Numbers
IP	internet protocol
LPM	longest prefix match

Bibliography & References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1101] Mockapetris, P., "DNS encoding of network names and other types", RFC 1101, DOI 10.17487/RFC1101, April 1989, <https://www.rfc-editor.org/info/rfc1101>.

[RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 2317, DOI 10.17487/RFC2317, March 1998, <https://www.rfc-editor.org/info/rfc2317>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <https://www.rfc-editor.org/info/rfc3596>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

Route Views Project <https://www.routeviews.org/routeviews/>

BGP Looking Glass Database <https://www.bgplookingglass.com/>

