Albuquerque Gigapop

- Network Aggregation Point in Central New Mexico
- Keeping New Mexico network traffic in New Mexico
Title: ABQG BGP Community standard

Summary:

ABQG (Albuquerque Gigapop), the New Mexico on-ramp service provider to Internet 2 and National Lambda Rail (NLR), also provides Local peering service for New Mexico local entities. The session will focus on BGP (Border Gateway Protocol) community strings that ABQG uses to serve its own peering partners and how peering partners can take advantage of this attribute for route manipulation.

Co-Presenter:

Ed May - Network Engineer UNM/ABQG emay@unm.edu
Paul Chang - Network Engineer UNM/ABQG pchang@unm.edu
Session Pre-requisites

- Interested in peering with ABQG
- Somewhat familiar with BGP (Border Gateway Routing Protocol)
**Prefixes in ABQG routing table**

<table>
<thead>
<tr>
<th>BGP peer</th>
<th>Prefix learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLR</td>
<td>9668</td>
</tr>
<tr>
<td>I2 R&amp;E</td>
<td>11264</td>
</tr>
<tr>
<td>I2 CPS</td>
<td>97780</td>
</tr>
<tr>
<td>TWTC</td>
<td>1</td>
</tr>
<tr>
<td>UNM</td>
<td>16</td>
</tr>
<tr>
<td>Checs</td>
<td>120</td>
</tr>
<tr>
<td>LANL</td>
<td>5</td>
</tr>
<tr>
<td>Esnet</td>
<td>146</td>
</tr>
<tr>
<td>CABQG</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>119,004</strong></td>
</tr>
</tbody>
</table>
119,000 +
Too many prefixes !!!!
How to scale this to be manageable?
Typical given requests:

- Newly peer ed client A wants to get UNM routes; I2 R&E routes but no LANL Routes; no I2 CPS routes.
Using BGP attributes: AS origin...AS Path, Next HOP, MED, Local preference....

**BGP Community is the Answer!**
### ABQG BGP Community assignment

**ABQG AS: Connector AS ==== 40498:XX**

<table>
<thead>
<tr>
<th>Items</th>
<th>Community Value</th>
<th>Description</th>
<th>Status</th>
<th>Service type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40498:4323</td>
<td>TWTC</td>
<td>In production</td>
<td>Commodity Service</td>
</tr>
<tr>
<td>2</td>
<td>40498:19401</td>
<td>NLR 10G</td>
<td>In production</td>
<td>NLR Service</td>
</tr>
<tr>
<td>3</td>
<td>40498:19402</td>
<td>NLR 1G</td>
<td>In production</td>
<td>NLR Service</td>
</tr>
<tr>
<td>4</td>
<td>40498:11537</td>
<td>I2 R&amp;E</td>
<td>In production</td>
<td>I2 Service</td>
</tr>
<tr>
<td>5</td>
<td>40498:11538</td>
<td>I2 CPS</td>
<td>In production</td>
<td>I2 Service</td>
</tr>
<tr>
<td>6</td>
<td>40498:3388</td>
<td>UNM primary</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>7</td>
<td>40498:3389</td>
<td>UNM secondary</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>8</td>
<td>40498:68</td>
<td>LANL</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>9</td>
<td>40498:3912</td>
<td>Checsnet</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>10</td>
<td>40498:293</td>
<td>Esnet</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>11</td>
<td>40498:26439</td>
<td>CABQ</td>
<td>In production</td>
<td>Peering partner</td>
</tr>
<tr>
<td>12</td>
<td>40498:17153</td>
<td>NMT</td>
<td>Coming soon</td>
<td>Peering partner</td>
</tr>
<tr>
<td>13</td>
<td>40498:64516</td>
<td>NMCAC</td>
<td>In production</td>
<td>NMCAC dedicated</td>
</tr>
</tbody>
</table>

**Note**

1. By defaults peering partner prefixes are announced to all ABQG peering partners

2. Community string listed above are additive.

**10/17/2008**
How does it work? From ABQG side

- ABQG tags all incoming prefixes learned with ABQG BGP community standard additively.
- ABQG then announce prefix to its clients based on what type of combination services client selects.
- Now the control plant is set up for this particular clients.
- When packet hit ABQG router, data plant will route traffic based on the control plant info.
How is it done? From ABQG side

BGP Mantra:

- Incoming policy affects outgoing traffic
- Outgoing policy affects incoming traffic.

- Example of the set up….
How to configure it?

A. Leg work first – some global configuration

```
ip as-path access-list 1 permit ^$  
ip as-path access-list 68 permit ^3388_  
ip bgp-community new-format  
ip community-list 11 permit 40498:4323  
ip community-list 12 permit 40498:19401  
ip community-list 13 permit 40498:19402  
ip community-list 14 permit 40498:11537  
ip community-list 15 permit 40498:11538  
ip community-list 16 permit 40498:3388  
ip community-list 17 permit 40498:3389  
ip community-list 18 permit 40498:68  
ip community-list 19 permit 40498:3912  
ip community-list 21 permit 40498:293  
ip community-list 22 permit 40498:26439  
ip community-list 23 permit 40498:17153
```

B. Route-map set up secondly so it can later applied to BGP peer:

```
route-map ITSin permit 10  
match as-path 68  
set local-preference 100  
set community 40498:3388 addictive  
route-map ITSin permit 20
```

```
route-map ITSOut permit 10  
match as-path 1  
!  
route-map ITSOut permit 20  
match community 11 12 13 14 15 16 17 18 19 21 22 23
```

C. Finally apply to routing policy for this peer:

```
router bgp 40498  
address-family ipv4  
neighbor 129.24.198.98 send-community both  
neighbor 129.24.198.98 route-map ITSin in  
neighbor 129.24.198.98 route-map ITSout out
```

D. Check the bgp routes to see whether community is properly applied.

E. Check whether the routes announced to your BGP peer is based on the community string list you applied.
Inspecting prefix sent by AS 668 - DREN

```
11537 668, (received-only)
  208.77.76.130 from 208.77.76.130 (64.57.28.246)
    Origin IGP, metric 0, localpref 100, valid, external
    Community: 668:100 11537:3000
505-6513-01#show ip bgp 6.4.0.0
BGP routing table entry for 6.4.0.0/16, version 37768977
Paths: (5 available, best #4, table Default-IP-Routing-Table)
   Advertised to update-groups:
       3       6       11      14      16
  19401 668
   129.24.198.110 from 129.24.198.110 (216.24.191.228)
     Origin IGP, metric 1351, localpref 400, valid, external
     Community: 668:100 19401:924 19401:7000 40498:19402
  19401 668, (received-only)
   129.24.198.110 from 129.24.198.110 (216.24.191.228)
     Origin IGP, metric 1351, localpref 100, valid, external
     Community: 668:100 19401:924 19401:7000
  19401 668, (received & used)
   129.24.198.106 (metric 20) from 208.77.76.144 (208.77.76.144)
     Origin IGP, metric 1350, localpref 600, valid, internal
     Community: 40498:19401
  11537 668
   208.77.76.130 from 208.77.76.130 (64.57.28.246)
     Origin IGP, metric 0, localpref 700, valid, external, best
     Community: 668:100 11537:3000 40498:11537
  11537 668, (received-only)
   208.77.76.130 from 208.77.76.130 (64.57.28.246)
     Origin IGP, metric 0, localpref 100, valid, external
     Community: 668:100 11537:3000
```
How does it work? From ABQG client side. (Name few possibilities)

- Do nothing
- Filter prefix(es) based on community value
- Filter prefix(es) based on combination of community value and AS origin
- Etc…..
Case Study I

Redundancy backup commodity service breaks between two ABQG partners.

Solution: Use ABQG community string attribute to manipulate inbound route policy to fix it.

(See details...)
What do we achieve here?

1. Make routing policy task scalable
2. Provide easy maintenance and set up for peering arrangement

What are other alternatives and limitations?

- Using MP-BGP VRFs
- MPLS-VPN
ABQG Stats

Enterprise Command Center
University of New Mexico Information Technology Services

ABQG Utilization Map

10/17/2008
References:

1. ABQG web site: [http://abqg.unm.edu/](http://abqg.unm.edu/)
2. UNM Enterprise Command Center (ECC): [http://ecc.unm.edu](http://ecc.unm.edu)
3. ABQG BGP community:
5. I2 Community: [http://noc.net.internet2.edu/i2network/bgp-communities.html](http://noc.net.internet2.edu/i2network/bgp-communities.html)
7. BGP best path selection: [http://www.ripe.net/projects/ris/docs/bgpcheat.html](http://www.ripe.net/projects/ris/docs/bgpcheat.html)
8. This presentation:
Q&A