Customized BGP Route Selection Using BGP/MPLS VPNs

Laurent Vanbever
Université catholique de Louvain, BE
Laurent.Vanbever@uclouvain.be

Pierre Francois (UCLouvain, BE), Olivier Bonaventure (UCLouvain, BE) and Jennifer Rexford (Princeton, USA)

Cisco Systems, Routing Symposium
Monday, Oct. 5 2009
Customized BGP Route Selection Using BGP/MPLS VPNs

Introduction and motivation

Implementing CRS

Practical considerations and solutions

Conclusion
Customized BGP Route Selection Using BGP/MPLS VPNs

Introduction and motivation

Implementing CRS

Practical considerations and solutions

Conclusion
BGP Route Selection: *One-route-fits-all* model

- A BGP router selects **one** best route for each destination
- Globally, AS E knows 4 paths towards D
- Locally, some routers only know one path (*e.g.*, C1...C3)
Many ISPs have a rich path diversity
- It is common to have 5-10 paths *per prefix*
- Different paths have different properties
- It could be in terms of security, policies, etc.
Clients may want different paths to the same prefix
- If C1 is a competitor of C, he’d prefer to reach D via A or B
- C1 may even want to pay an extra fee for that

BGP Route Selection: *One-route-fits-all* model
With vanilla BGP, you can’t match customers’ preferences to available paths.

Customers of a given PE receive the same path.
Under CRS, one router can offer different interdomain routes to different neighbors.

- C1 reaches D via B, C2 reaches D via C.
Customized BGP Route Selection
Using BGP/MPLS VPNs

Introduction and motivation

Implementing CRS

Potential issues and solutions

Conclusion
Two notions: *class* and *service*

- A *class* is a set of routes sharing a property
  - *e.g.*, all the routes learned via provider *X*
  - One route can belong to more than one class

- A *service* is the union of one or more classes
  - Some classes can be preferred over others
  - *e.g.*, service *Y* is the union of *class 1* and *class 2*
  where preference is given to *class 1*
What do we need to implement CRS with BGP MPLS VPNs?

- Mechanisms to *disseminate* and *differentiate* paths
  - Multiprotocol BGP is used as dissemination protocol
  - Route Targets (RT) are used to identify classes
  - Route Distinguishers (RD) are used to ensure diversity

- *Customized* route selection mechanisms at ASBR
  - Use Virtual Routing and Forwarding (VRF) instances to build services

- Traffic forwarding on the chosen paths
  - MPLS tunneling
How do we implement CRS with BGP MPLS VPNs?

- C1 wants to reach D via B, C2 via C
- Define two services on R1: prefer B (resp. C) routes
- Define three classes: learned via A, B or C
How do we implement CRS with BGP MPLS VPNs?
How do we implement CRS with BGP MPLS VPNs?

- Consider peers as VPNs and put them in VRFs
How do we implement CRS with BGP MPLS VPNs?

- Consider peers as VPNs and put them in VRFs
- Use RT to identify *classes*

<table>
<thead>
<tr>
<th>Route Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>101: learned via A</td>
</tr>
<tr>
<td>102: learned via B</td>
</tr>
<tr>
<td>103: learned via C</td>
</tr>
</tbody>
</table>
How do we implement CRS with BGP MPLS VPNs?

- Consider peers as VPNs and put them in VRFs
- Use RT to identify *classes*
- Use different RD to differentiate routes
How do we implement CRS with BGP MPLS VPNs?

- Define *services* by using VRFs’ import filters

<table>
<thead>
<tr>
<th>Route Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>101: learned via A</td>
</tr>
<tr>
<td>102: learned via B</td>
</tr>
<tr>
<td>103: learned via C</td>
</tr>
</tbody>
</table>

**prefer B routes**
- import RT: 101,102,103;
- from 102: set pref:=200;

**prefer C routes**
- import RT: 101,102,103;
- from 103: set pref:=200;
How do we implement CRS with BGP MPLS VPNs?

- MPLS is used for forwarding
  - Two levels label stack
  - R3 only knows label to reach the PEs

<table>
<thead>
<tr>
<th>Route Targets</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>101:</td>
<td>learned via A</td>
</tr>
<tr>
<td>102:</td>
<td>learned via B</td>
</tr>
<tr>
<td>103:</td>
<td>learned via C</td>
</tr>
</tbody>
</table>
CRS applied to *classical* policies

- Define three classes
  - Providers (RT 100)
  - Peers (RT 101)
  - Customers (RT 102)

- Define two services
  - VRF Provider/Peer
    - `import RT 102;`
  - VRF Customers
    - `import RT 100,101,102;`

- Thanks to VRF isolation, policies violations vanish
Customized BGP Route Selection Using BGP/MPLS VPNs

Introduction and motivation

Implementing CRS

Practical considerations and solutions

Conclusions
Is CRS pushing a M120 to the limit?

Four services are defined on the Unit Under Test (UUT)

- Each service is fed with one class (one RT)
- In each class, ~300k routes (1 path per route)
- In the end, 1.200.000 routes in RIB & FIB
Is CRS pushing a M120 to the limit?

- UUT was a Juniper M120 [JunOS 9.3R2.8]
- Routing Engine (RE) has 4 GB DRAM
- Forwarding Engine Boards (FEB) have 512 MB DRAM

<table>
<thead>
<tr>
<th></th>
<th>RE</th>
<th>FEB</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>17%</td>
<td>9%</td>
</tr>
<tr>
<td>fully-loaded</td>
<td>38%</td>
<td>39%</td>
</tr>
<tr>
<td>(1,200,000 routes)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- FIB could handle more than 2,000,000 routes
- Enough to support a few services *without* modifications
More services? *scalability* and...*scalability*

- Routes *dissemination* overhead
  - All PEs receive all VPN routes

- Routes *storage* overhead
  - RIB
    - Modest performance demand
    - Add more DRAM to support CRS?
  - FIB
    - CRS’s biggest challenge
    - Sharing between the VRFs in the FIB?
How could we improve CRS FIB’s scaling: *Selective VRF Download*

- By default, *all* VRFs are installed on *all* line cards

<table>
<thead>
<tr>
<th>Slot</th>
<th>State</th>
<th>Temp (C)</th>
<th>CPU Utilization (%)</th>
<th>Memory DRAM (MB)</th>
<th>Heap Utilization (%)</th>
<th>Buffer Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Online</td>
<td>24</td>
<td>1</td>
<td>0</td>
<td>512</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>Online</td>
<td>28</td>
<td>1</td>
<td>0</td>
<td>512</td>
<td>39</td>
</tr>
</tbody>
</table>

- Customers ask for the same services?
  - Connect them on the same line card
  - Download VRFs only to line cards that need them
  - It could be a management nightmare...
How could we improve CRS FIB’s scaling: *Cross-VRF Lookup*

- Specific routing for a small set of prefixes?
- Create one small VRF *per service*
- Add default entry towards a default VRF
- The price to pay is 2 IP lookups

<table>
<thead>
<tr>
<th>VRF1</th>
<th>VRF2</th>
</tr>
</thead>
<tbody>
<tr>
<td>*-&gt;10/8 via R1</td>
<td>*-&gt;10/8 via R2</td>
</tr>
<tr>
<td>0/0 via default</td>
<td>0/0 via default</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>*-&gt;10/8 via R3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
How could we improve CRS FIB’s scaling: *Distributed VRF*

- Distribute VRFs among routers which can afford extra load
- PEs do not maintain complete VRFs anymore
- PEs default route traffic towards these routers
- Increase in latency and load
- Distributed version of *Cross-VRF Lookup*

![Diagram showing detour and direct paths between R, R1, and R2]
Customized BGP Route Selection Using BGP/MPLS VPNs

Introduction and motivation

Implementing CRS

Practical considerations and solutions

Conclusion
CRS is feasible

- **Implementable**
  - It can be realized on today’s routers
  - It uses well known BGP MPLS/VPNs techniques

- **Scalable (for a few services)**
  - “Modest” message and storage overhead
  - Lab experiments tend to confirm that

- **Guaranteed interdomain convergence**
  - Extra flexibility does not compromise global routing stability\(^1\)

---

\(^1\) Proof in SIGMETRICS'09 paper by Y. Wang, M. Schapira, and J. Rexford
Customized BGP Route Selection
Using BGP/MPLS VPNs

Questions?

Cisco Systems, Routing Symposium
Monday, Oct. 5 2009