Announcements

Lectures 1-14 are available.

Still looking for a TA.

Acknowledgement: some of the slides for this lecture have been “inspired” by Tim Griffin’s BGP Tutorial.
Learning External Prefixes

- So far, BGP has been presented as a pure EGP.
  - A protocol that runs between ASs.

- How do A, C and D learn about AS2’s routes?
  - Ditto for Y, Z, T about AS1’s routes?
- I.E., how are prefixes learned by an ASBR distributed inside the AS?
Learning External Prefixes, cont’d

- Inject into the IGP (using AS-External LSAs).
- Small networks can do this.
  - Default route + a few external routes.
- Does not work for large ISPs.
  - They carry a full routing table (100K-400K routes!).
- Would lose policy information.
  - No way to carry attributes.
- IGPs don’t scale well.
  - Computational complexity.
  - Memory requirements.
  - Additional traffic.
    - Fragmented LSAs.
- Clearly need a different way!
E-BGP and I-BGP

- The solution is called *Internal-BGP (I-BGP)*.
  - As opposed to *External-BGP (E-BGP)*.
- E-BGP is used between ASs.
- I-BGP is used **within** an AS.
  - Is used to distribute routes learned with E-BGP.
- E-BGP and I-BGP are the same protocol.
  - Same messages, attributes, state machine, etc.
- But: different rules about route redistribution:

<table>
<thead>
<tr>
<th>Learned from</th>
<th>Redistribute to I-BGP</th>
<th>Redistribute to E-BGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-BGP</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>E-BGP</td>
<td>yes</td>
<td>(yes)</td>
</tr>
</tbody>
</table>
I-BGP Route Redistribution

- How does D learn routes acquired by B?
  - Since A can’t redistribute routes learned over I-BGP?
- If D also had an external connection, how would it redistribute routes learned from other ASs?
I-BGP Route Redistribution, cont’d

• Remember: BGP is a \textit{routed} protocol.
• Routes between routers already exist.
  – Carried by the IGP.
• I-BGP sessions can be formed between non-adjacent routers.
• I-BGP sessions must form a full mesh:
IGP / I-BGP Interaction

• Full mesh.
• Independent of actual links between (internal) routers.
• TCP src/dst of I-BGP session must be a loopback address.
  – Routing to the router must be independent of interfaces going up/down.
  – (Loopback) address of IBGP routers advertised as a /32 within the IGP.
• Full mesh is necessary to prevent loops.
  – AS_PATH is used to detect loops in E-BGP.
  – ASN appended to AS_PATH only when route is advertised to E-BGP peer.
• I-BGP is **NOT** an IGP.
  – Nor can be used as one.
NEXT_HOP and I-BGP

NEXT_HOP is rewritten to the loopback address.
BGP Route Selection is about Policy

- AS1 exports C1’s prefix to AS2.
- AS1 accepts C2’s prefix from AS2.
- AS2 accepts C1’s prefix from AS1
- AS2 does not export any prefixes learned from AS3 to AS1.
- ...
How Are Routes Chosen?

- AS3 has peers, customers, and a provider.
- What routes does it accept?
- What routes does it advertise?
Customer-Provider & Peer-Peer Rltnshps

• Enforce transit relationships:
  – Filter outbound routes.

• Enforce order of route preference:
  – Customer ≻ Peer ≻ Provider.

  – More rules on route preference later.
Imported Routes

Routes arrive from various sources: provider (★), peer (≡), customer (($) ), and own IGP (⊙).
Exported Routes

- Filters (・・・・・) block peer and provider routes!

To provider

To customer

AS 3

To peer
Picking Routes for Redistribution

- How does AS3 know which routes are customer/peer/provider/IGP?
- If AS3 were a single router, it could peek into Adj-RIB-In-x.
- But routes are redistributed with I-BGP.
  - Router that talks to provider is not router that talks to customer.
  - Routers could be (and were) configured with all of an AS’s customer/peer/etc ASes to do output filtering.

Better answer:

- COMMUNITY attribute.
COMMUNITY

• Specified in RFC 1997.
• Encodes arbitrary properties.
  – E.g., all of customer’s routes get a specific COMMUNITY.
• Much of the policy is specified using communities.

• Optional, Non-transitive. Type=8
• List of community values (length is multiple of 4).
  – Each prefix can belong to multiple communities.
• Each community value is 4 bytes: (e.g., 7018:100)
  – 2 bytes ASN (by convention).
  – 2 bytes administratively defined (no predefined meaning).
COMMUNITY, cont’d

- $0x00000000$ through $0x0000FFFF$ are reserved.
- $0xFFFF0000$ through $0xFFFFFFFF$ are reserved.
- $0xFFFFFFFF01$: NO_EXPORT
- $0xFFFFFFFF02$: NO_ADVERTISE
- $0xFFFFFFFF03$: NO_EXPORT_SUBCONFED

- Community values have local (intra-AS) meaning.
- Community values can also have meaning between two neighboring ASes (following bilateral agreement).

- Terminology: Route Coloring.
COMMUNITY Example

• When AS3 imports routes, it colors them with the appropriate community string.
  – From customers (💰): 3:100.
  – From providers (⭐): 3:300.

• When AS3 exports routes, it picks them according to their community string.
  – To customers: 3:100, 3:200, 3:300
  – To peers: 3:100
  – To providers: 3:100
Martians (or bogons)

• Some prefixes should not be advertised.
  – Some should not even appear!
  – Default (0.0.0.0/0) routes are never advertised.
  – Site-local (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16).
  – Link-local (169.254.0.0/16).
  – Loopback (127.0.0.0/8).
  – IANA-reserved (128.0.0.0/16, 192.0.0.0/24, etc.).
  – Test networks (192.0.2.0/24, etc.).
  – Class D and E (224.0.0.0/3).
  – Unallocated space.
    • Careful with that!

• Routes to martians are filtered on input.
  – Not that they should ever have been advertised!
Black Holes Are Out of Sight

• If another AS advertises one of our prefixes, bad things happen:
Black Holes Are Out of Sight

• Our prefix becomes unreachable from the part of the net believing C4’s announcement.
Preventing Bad Routing

• Preventing black holes:
  – Only accept customer routes advertising customer’s prefixes.
  – AS6 should only accept C4’s real prefixes, not anything C4 advertises.

• Filter out Martians:
  – Private address space is sometimes used for intra-AS management.
    • Should not accept routes for it!
  – Be a good citizen, do not leak martians!
Imported Routes, revisited

When importing, filter martians (辐射) and potentially bad customer routes (骷髅). Also, drop looping AS_PATH.

From provider

From peer

From customer
In/Out Route Processing

AS 1

Adj-RIB-In-1

Input Policy

Loc-RIB

Output Policy

Adj-RIB-In-2

AS 2

Adj-RIB-Out-1

Adj-RIB-Out-2

Adj-RIB-In-n

AS n

Adj-RIB-Out-n

FIB
Input Policy

• Apply input filtering.
  – Routes that are dropped here are not used internally.
  – Nor are they advertised.
  – They are dead!
• Tweak attributes:
  – Set LOCAL_PREF, add COMMUNITY
• Select best route.
  – Based on Path Attributes.
• Create Route table.
• Populate Forwarding table.
Best Route Selection

- If NEXT_HOP inaccessible, route is dropped.
- [cisco only] prefer path with highest weight.
- Select route with highest LOCAL_PREF.
- Prefer shortest AS_PATH.
- Prefer lowest origin (IGP < EGP < INCOMPLETE).
- If routes received from same AS (or bgp always-compare-med enabled), and MED enabled, prefer lowest MED.
- Prefer E-BGP paths over I-BGP paths.
- Prefer shortest IGP path to NEXT_HOP.
- Use lowest router ID as tie-breaker.
  - Some implementations use first installed route instead.
Why prefer E-BGP over I-BGP?

- B learns route to AS2 over E-BGP from K.
- B learns route to AS2 over I-BGP from C
  - (who learned it from L).
- Same local pref, as_path length, origin, etc.
- Obviously should use K!
What is the Best Route?

Which of the four possible routes will 9.5.1.2 take to get to AS4?
What is the Best Route?

• LOCAL_PREF to the rescue!
Alternatively…

- Now shortest AS_PATH takes effect!
Backup Links (outbound traffic)

- Set higher local pref on primary link on all routes from AS1.
- Forces all traffic to take primary unless it is down.

![Diagram showing backup links with LOCAL_PREF values]

- AS 1
  - LOCAL_PREF=100
- AS 2
  - LOCAL_PREF=50
Multihomed Backups (outbound traffic)

- Same idea.
Back to AS_PATH

- Traffic often follows reverse of AS_PATH:
• But it might not!
• AS2 filters prefixes longer than /24.
• Packet to 12.2.61.19 actually makes it to AS5.
Shortest AS_PATH?

- 1 2 3 4 or 1 5 4?
Backup Links (inbound traffic)

- Hack: AS_PATH padding.
Backup Links (inbound traffic)

- AS_PATH padding does not shut off all traffic.
- AS 9 has higher LOCAL_PREF for customer routes.
- Some traffic from AS9 still flows through the backup link.

![Diagram of backup links between AS 1, AS 2, and AS 9 with LOCAL_PREF values]
Backup links (inbound traffic)

- COMMUNITY to the rescue!
- AS9 has LOCAL_PREF = 100 for customer and 90 for peer.
- AS9 has the following import policy:
  - If 9:90 in community, set local_pref to 90.
  - If 9:80 in community, set local_pref to 80.
  - If 9:70 in community, set local_pref to 70.
- AS2 advertises its routes (over the backup link to AS9) with community 9:70.
- Now peer has higher local pref and traffic flows as intended!
Policy Interaction

• Example: backup route with community hack.
• AS4 advertises prefix a over its (only) link.

![Diagram showing the policy interaction between AS1, AS2, AS3, and AS4. AS4 advertises prefix a with LOCAL_PREF=100, while AS1 advertises with LOCAL_PREF=90. This creates a policy interaction.]
Policy Interaction cont’d

- Backup link gets installed, AS1 advertises community 4:70.
- AS4 still prefers route via AS3 (highest local_pref).

![Diagram of AS Network]

AS 1

AS 2

AS 3

AS 4

- LOCAL_PREF=80
- LOCAL_PREF=100
- LOCAL_PREF=90
- LOCAL_PREF=100
- LOCAL_PREF=123
Backhoe Severs Primary Link

- AS2 withdraws route to a.
- Backup link takes over.
Primary link restored

- AS4 is still advertising route to AS1.
- Route from AS2 has lower local pref, gets ignored!
- Route pinning.