BGP Path visibility issues
ToC

- iBGP
  - draft-ietf-idr-add-paths
    Why doing Add-paths
  - draft-ietf-idr-add-paths-guidelines
    (draft-uttaro-idr-add-paths-guidelines)
    Why only a small subset of proposals will be supported

- eBGP
  - Threats to BGP policies
Motivation for Add-paths

- Initial “motivation” was MED oscillation avoidance
- Emergence of new IDR requirements a few years ago
  - Fast recovery upon peering link / ASBR failure
  - Load balancing among multiple primary BGP NHs
  - Hitless planned maintenance
- “Optimal” hot-potato routing
- (Churn reduction / convergence concealment)
iBGP Path hiding

- Lack of path diversity in iBGP deployments
  - Policies
  - Route Reflection
Policies let paths be hidden

p : localpref = 100

p : localpref = 90
Route Reflection hides paths

\[ p : \text{localpref} = 100 \]

\[ p : \text{localpref} = 100 \]
Can’t we just turn adv-best-external on?

Doesn’t solve the route reflection problem
BGP Add paths

- Advertise multiple BGP paths
- towards the same NLRI
- over a single iBGP session
- draft-ietf-idr-add-paths
BGP Add paths

- PE can
  - Install PIC state
  - Load balance
  - Hitless reroute to alternate
Optimal Hot Potato

- RRs may perform different IGP tie-breaking
- Clients don’t get the path that they would pick
- Add-paths enabled RRs let the IGP tie-break to clients
  - Depending on which paths it advertises
Churn reduction

- Churn reduction for primary paths...
- ...with internal churn increase for non-primary ones
Churn Reduction

p : localpref = 100

Withdraw, Update
Withdraw, Update, Update
Update (or silence)
Churn Reduction

Update... or silence...
draft-ietf-idr-add-paths

- Adds an identifier to paths
- Identifier only has session meaning
draft-ietf-idr-add-paths-guidelines

• draft-ietf-idr-add-paths doesn’t tell which paths to select
• Multiple motivations lead to different “selection modes”
  • Evaluate them (what they give, at which cost)
    • analytical
    • “numbers”
Modes

- All paths
- N paths
- AS-Wide best paths (and variants)
- Best Loc Pref / Second best Loc Pref paths
- Decisive step -1 paths
- Neighbor-AS group best paths
Known paths almost like if iBGP full-mesh and adv-ext-best on
Add-All

- Easiest Decision Process algorithm
- Nice mode to turn on towards a BGP monitor
- Memory/internal update churn monster
  - Depending on how many paths for each p
N paths \((N\) is configured\)

Usually consider NH disjoint paths
Add-N-Paths

- Most practical use cases
  - Set N to 2 for basic PIC support
  - Set N to desired number of NHs for LB
- Memory hit kept under control through configuration of N
- Doesn’t solve MED oscillations
- Developers tend to implement it as N*DP
AS-Wide Best paths

Not hiding paths that another node would have preferred

(AS Path) (MED)
AS-Wide Best paths

- “The router doesn’t make local decisions”
- DP complexity < not running add-paths
- Provides routing optimality and max LB potential
- Provides MED oscillation avoidance

- !!! Doesn’t feed PIC !!!
AS-Wide Best paths

AS Path length: 3

LP 100

LP 90

AS Path length: 3

One winner problem still applies
AS-Wide Best paths

LP 100
AS Path length : 3

LP 100
AS Path length : 2

LP 90
AS Path length : 3

One winner problem still applies
Best LP/Second Best LP

- If \(\#(\text{paths with highest LP}) > 1\)
  - advertise paths with highest LP
- else
  - advertise the path with highest LP
  - advertise the paths with second highest LP
Best LP/Second Best LP

LP 100
AS Path length : 3

LP 100
AS Path length : 3

LP 90
AS Path length : 3

(LP)
Best LP/Second Best LP

LP 100
AS Path length : 3

LP 90
AS Path length : 3

LP 80
AS Path length : 3

There are always multiple “winners”
Best LP / Second Best LP

- Adj-Rib-In optimized for this mode contains two or three sets of paths per NLRI
  - Best bin
  - Second best bin if required
  - Others

- Decision Process:
  Select what’s in first and second bin
Decisive step - 1

- Apply normal BGP selection process, but
  - If IGP tie-break rule is reached, advertise what remains
  - If best path is found at a preceding rule i, advertise what remained when applying rule i-1
- Tries to obtain diversity while advertising as few paths as possible
Decisive step - 1

LP 100
AS Path length : 3

LP 90
AS Path length : 3

LP 90
AS Path length : 3

One LP winner
--> All paths!
Decisive step - 1

LP 100
AS Path length : 3

LP 100
AS Path length : 2

LP 90
AS Path length : 3

One AS Path winner:
Best Loc Pref paths
Neighbor-AS group best

- Avoids MED oscillations
  - draft-walton-bgp-route-oscillation-stop
- Advertise the best path from each neighboring AS
  - No ASBR picks as best a non-lowest MED path
Neighbor-AS group best

- Provides paths from different neighboring ASes, but
- their existence is not guaranteed
- nothing to deal with post-convergence paths
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-I</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td></td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>can be optimized</td>
<td></td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>“spaghetti”</td>
<td></td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>Path optimality</td>
<td>Backup availability / optimality</td>
<td>Control plane load and stress</td>
<td>DP Complexity</td>
<td>MED osc. avoidance</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>----------------------------------</td>
<td>-------------------------------</td>
<td>---------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>can be optimized</td>
<td></td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td><strong>N</strong></td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td><strong>AS-Wide</strong></td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td><strong>LP1/LP2</strong></td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td><strong>Decisive-1</strong></td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td><strong>Group best</strong></td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
# Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-I</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
## Summary

<table>
<thead>
<tr>
<th></th>
<th>Path optimality</th>
<th>Backup availability / optimality</th>
<th>Control plane load and stress</th>
<th>DP Complexity</th>
<th>MED osc. avoidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>OK</td>
<td>OK</td>
<td>Max</td>
<td>EASIEST</td>
<td>OK</td>
</tr>
<tr>
<td>N</td>
<td>?</td>
<td>OK / ?</td>
<td>Bounded</td>
<td>Depends on N can be optimized</td>
<td>?</td>
</tr>
<tr>
<td>AS-Wide</td>
<td>OK</td>
<td>KO / ~OK</td>
<td>~MAX</td>
<td>EASY</td>
<td>OK</td>
</tr>
<tr>
<td>LP1/LP2</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>EASIER</td>
<td>OK</td>
</tr>
<tr>
<td>Decisive-1</td>
<td>OK</td>
<td>OK</td>
<td>~MAX</td>
<td>Easy but “spaghetti”</td>
<td>OK</td>
</tr>
<tr>
<td>Group best</td>
<td>KO ...</td>
<td>KO</td>
<td>~MAX</td>
<td>?</td>
<td>OK</td>
</tr>
</tbody>
</table>
Current Recommendations

- MUST: Add-N
  - Default MUST be 2
  - N MUST be configurable
  - Option to not limit N (Add-All)

- OPTIONAL: AS-Wide best variants

- OPTIONAL-: All others
Diverse paths

• Competing solution

• Use // sessions instead of modified session
Deployment

- Session wide upgrade required (vs. per router)
- Add-path easily converted to diverse-paths
- As for all solutions
  - Forget about deployments w/o Ingress-Egress encap
  - Transient forwarding loops if naïve PIC implementation
Tool

- virginie.vandenschrieck@uclouvain.be
- input: BGP config, IGP config, as many `show ip bgp all` as possible, priority on RRs, please adv-ext-best
- output: for each mode
  - number of paths in Rib-in
  - optimality of paths
  - iBGP churn upon nexthop failure / single update
  - generated eBGP churn upon nexthop failure
Next Steps

- Add-path for eBGP
  - Route Server implementation
  - draft-jasinska-ix-bgp-route-server
    + Add-All
    + Filtering
    + Pick one for clients not supporting add-paths
eBGP path visibility issues
“How to sell BGP transit at no or little cost”
Observation 1

• Data plane is often disregarded when thinking about BGP

• “A BGP router will pick a path towards a given destination by applying the following rules”

  Weight
  Local-pref
  As Path Length
  IGP/Med
  ...

Observation 1

• Data plane is often disregarded when thinking about BGP

• “A BGP-router’s route processor will pick a path towards a given destination prefix by applying the following rules”
  
  Weight
  Local-pref
  As Path Length
  IGP/Med
  ...


In this talk
Think FIB

- Traffic follows **data-plane** state

- A **FIB** will pick a path towards a given **destination address** by applying the following rules

  **Longest prefix match to get the prefix**
  Best path towards that prefix was picked based on Weight
  Local-pref
  As Path Length
  IGP/Med
  ...

Observation II
Sprint’s publicly available recognized BGP community values (some of)

• Have to be a customer of Sprint

• 65000:XXX : Do not advertise to ASXXX can be AOL, NTT, BT-Infonet, Level3, GBLX, HKIX-RSI, Verizon, AT&T, ...

• (NO-EXPORT recognized)
Legend

A BGP Prefix advertisement for p/P

An advertisement of a prefix more specific than p/P, say p/P+2
What can you do with these communities?

- Play with

- Assume Y is also one of your providers

- Turn “don’t advertise to X” values into a only “advertise to Y”
  Just put them all but Y

- or explicit “only advertise to Y” community
Initial routing status
control-plane (only) driven forwarding

ISP A

ISP B

Stub AS Me
Initial routing status
control-plane (only) driven forwarding

ISP A

ISP B

Stub AS Me
Initial routing status
control-plane (only) driven forwarding
Let’s start playing: Inbound TE, increase RIB/FIB of everyone
Let’s start playing: Inbound TE, increase RIB/FIB of everyone
Let’s start playing: Inbound TE, increase RIB/FIB of everyone.
Let’s start playing: Inbound TE, increase RIB/FIB of everyone.
Let's start playing: Inbound TE, increase RIB/FIB of everyone
Let’s start playing: Inbound TE, increase RIB/FIB of everyone.
Let's start playing: Inbound TE, increase RIB/FIB of everyone
Let’s start playing: Inbound TE, increase RIB/FIB of everyone
Let’s start playing: Inbound TE, increase RIB/FIB of everyone

ISP A no longer provides transit for

ISP A only provides customer connectivity for its peer route
Let's start playing: Inbound TE, increase RIB/FIB of everyone.

The rest of the Internet goes through ISP B for
Let's start playing: Scope advertisement of the more specific
Let's start playing: Scope advertisement of the more specific
Let’s start playing: Scope advertisement of the more specific

Only to ISP A!
Let's start playing: Scope advertisement of the more specific

Only to ISP A!
Let’s start playing: Scope advertisement of the more specific

Only to ISP A!
Let’s start playing: Scope advertisement of the more specific

Only to ISP A!
Let’s start playing : Scope advertisement of the more specific

ISP A does not propagate BGP paths for to its providers and peers

It still does for

is likely to be installed in the FIB !
New paths in the network
New paths in the network
New paths in the network

ISP A

ISP B

Stub AS Me

Only to ISP A!
Consider this

• As a threat
  • Detect and shout
  • Defend
  • Inform about the threat to operators
• As a “flexibility opportunity” (do it openly)
  • Capture incentives
New paths in the network

ISP A

ISP B

Stub AS Me

Only to ISP A!
(Tell him to NO-EXPORT)
Incentives

• ISP A cannot do much on its own but

  • break the traffic / be bypassed / not follow customers’ desires (bring packets where he wants) or

  • try to get some “ROI” from this provider-peer path (consider it as a provider-customer)
Incentives

• ISP B
  • Gets some more traffic to be delivered to his customers
  • “Controls” the offering of the service
Incentives

- Customer
  - Does inbound TE
  - up to /32
  - rate does not matter much
  - more predictable performance impact
Expected money flow

• Stub to ISP B to ISP A

• ISP B goes into a customer-provider relationship with ISP A for some selected prefixes
Recommendation

- Detect and shout
- paolo@pmacct.net
- www.pmacct.net
Thanks