BGP Auto Discovery

R. Raszuk, J. Mitchell, W. Kumari, K. Patel, J. Scudder

Agenda

A bit of history

An overview

Idea presentation

Discussion

History ....

- The effort has been started in 2004 where the proposal has been made to flood BGP configuration information in IGPs as an opaque data
- Draft presented in Vienna IETF during IDR as well as IGP WGs .. Well accepted !
- Subsequent conversations with customers and industry collegues provided a recommendation to migrate this idea to 100% BGP contained without using IGP
- Key contributors & co-authors of this idea (2005): Pedro Marques, Rex Fernando, Keyur Patel
- Other contributors and co-authors: Chandra Appana & Enke Chen

Agenda

A bit of history

An overview

Idea presentation

Discussion

Overview

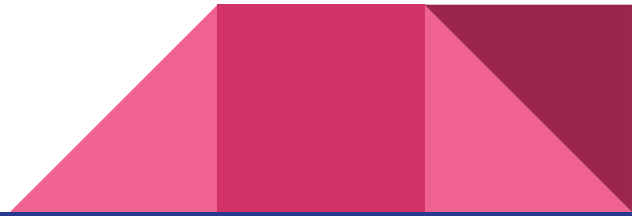
- In today's networks most of the applications and protocols have the capability to auto discover their peers, auto negotiate and auto establish peering relations.
- BGP until now is one of those protocols which still require manual neighbor configuration
- For Tier 1|2 EBGP peers this will likely still remain the case as in most cases peering relation require policy configuration
- For lower tiers where peering via Route Server is taking place auto discovery could happen ..
- The original focus was on auto IBGP peering or different AS peering where AS#s do not matter (CSC case). Now we extended it also for IXP use case and EBGP auto peering.

Use of RRs

- This proposal is not an attempt to propose the removal of ibgp route reflectors.
- They have their own applicability in many networks and will likely remain there for a long time.
- This proposal offers an alternative for those customers where manual peering provisioning for any SAFIs (existent or new) are bringing more opex cost then required.
- In particular this proposal is targetted to enterprise customers, customers using CSC technology or those service providers which do use full mesh today (Worldcom, NTT-Global Communications ...)

RRs today

- RRs were introduced and are used today for the following reasons:
  1. Configuration ease when adding or deleting IBGP peers
  2. Information reduction ... Sending best path only
  3. Reduction of number of TCP sessions to be handled by BGP speakers
  4. Efficient update generation (peer groups)
- This proposal addresses point #1
- Let’s now discuss points 2 - 4 .....



#2 Information reduction … best path propagation

- Already no reduction for vpnv4 routes with different RDs per VRF
- No reduction when IBGP multipath is required
- No reduction when ADD_PATHs or Group Best External ideas are implemented for oscillation reduction, for BGP fast convergence or for BGP FRR improvements

Conclusion:

There is number of reasons in today's deployments when propagation of more then best path is required. RR/PE code change (add-path / Nth best) or auto full mesh of RR clients solves all of the above cases.

#3 Reduction of number of TCP sessions

Conclusions:

Under most circumstances today number of TCP connections required for IBGP session to all ASBRs/PEs is sufficient and supported by current routers.

As most networks use MPLS tunneling in the core P routers do not require to be part of classic IBGP full mesh

New platforms, code improvements and RP blades will increase those numbers even further.

#4 Efficient update generation (update/peer groups)

- Updates are generated for each peer group then just replicated to members
- Per prefix filtering require peer group multiplication, but operators do not filter per prefix on ibgp side
- Per attribute filtering (example: outbound RT based filtering) in VPNv4 routes does require an extra cycles in update generation, but can be substituted with much cheaper inbound filtering

Conclusion:

In case of equal number of peer groups to RRs or ASBRs/PEs update generation costs are comparable due to cheap replication costs.

Agenda

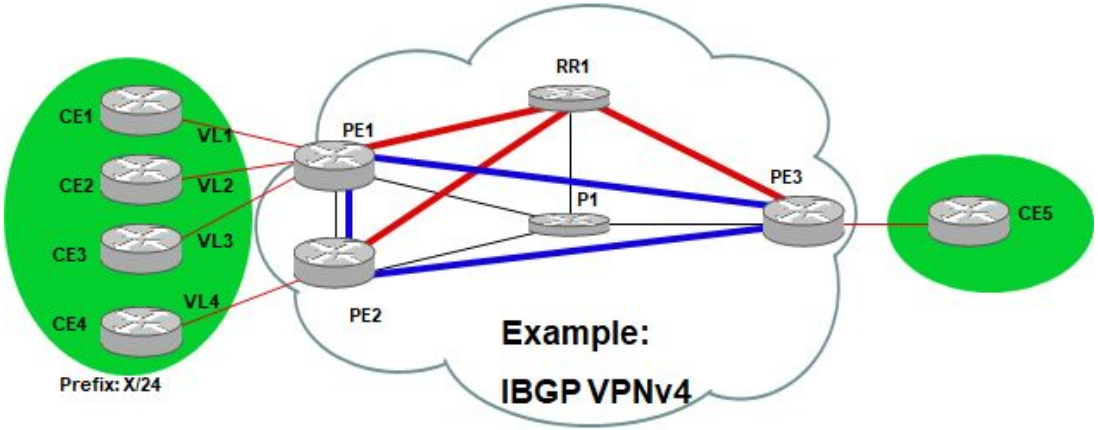
A bit of history

An overview

Idea presentation

Discussion

BGP Build In Auto Discovery



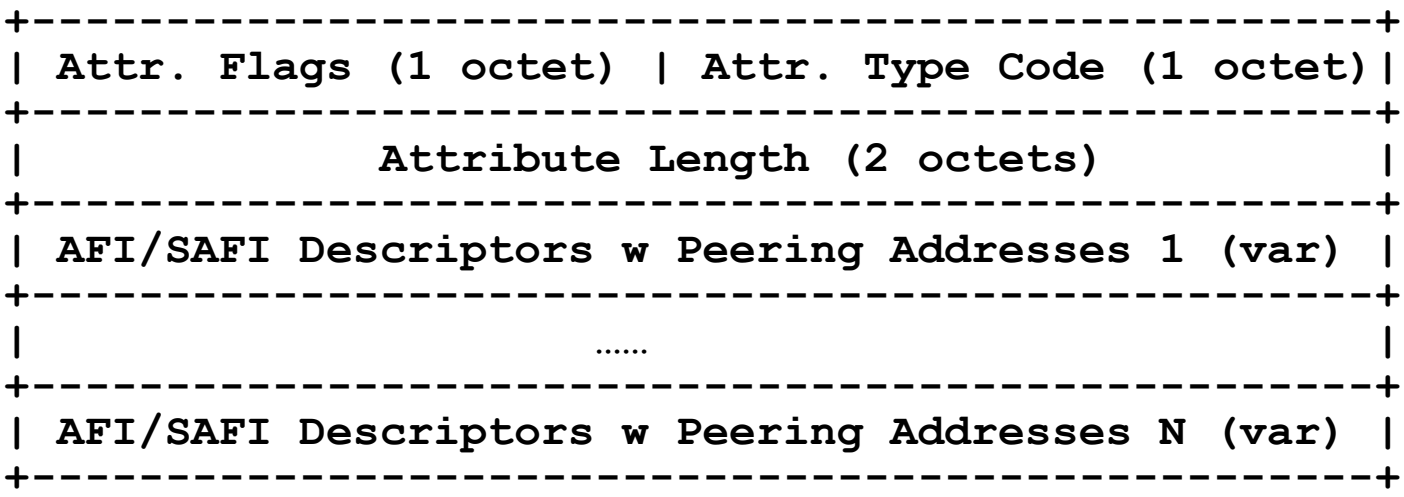
- Control sessions are configured from PEs to RR and established (red)
- RR propagates the config information per AFI/SAFI to all of his discovery clients
- Clients use this information to establish full mesh peering between BGP speakers for given SAFI (blue)

BGP Build In Auto Discovery - details

- BGP configuration information will be carried in new AFI/SAFI with the format of NLRI being <Group_ID:Router_ID> [2 octet:4 octet]
- Group_ID allows to auto configured scoped mesh groups creation (example: multiple confederations can be auto discovered and meshed with single control plane RR)
- Grouped meshes will be interconnected by the operator.
- Def Group_ID is all zeros (2 octets)

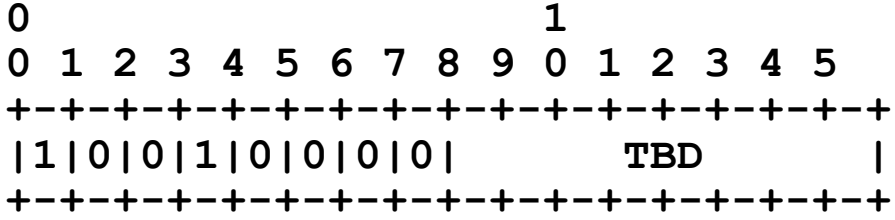
BGP Build In Auto Discovery - details

- A new BGP Peer Discovery Attribute is defined to carry information about all activated and flagged for auto discovery AFI/SAFIs



BGP Build In Auto Discovery - details

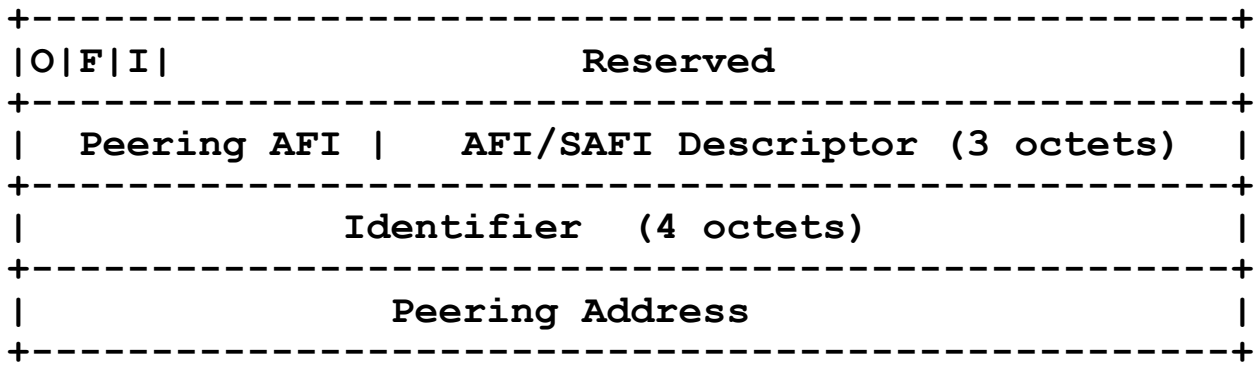
Flags & Type code fields:



Bit 0 - Optional attribute (value 1)
Bit 1 - Non transitive attribute (value 0)
Bit 2 - Partial bit (value 0 for optional non transitive attributes)
Bit 3 - Extended length of two octets (value 1)
Bit 4-7 - Unused (value all zeros)
Type code - Attribute type code (TBD)

BGP Build In Auto Discovery - details

AFI/SAFI Descriptor:



O bit - Route originator or EBGP speaker (Yes - 1, No - 0)
F bit - Force new peering. Default not set - 0, set - 1.
I bit - Informational only (Do not require to establish a bgp connection)

BGP Build In Auto Discovery - details

- On the change of Router_ID on BGP speaker sessions are restarted therefor new BGP NLRIs are generated and distributed
- No aggregation of BGP Peer Discovery on RRs is required
- The scope of discovery is per AFI, SAFI and Group_IDs in order to create disjoined meshes
- BGP speakers which do not originate any BGP routes should not attempt to establish BGP sessions to any other non originators.
- Presence of F (force) flag when used for previously advertised given AFI/SAFI/ID but now with different peering address should cause automatic session reset. Not set would require manual session clear.

BGP Build In Auto Discovery - details

- Possible „Informational” mode operation when BGP peers are discovered but the actual routes peering still happens with the manual action.
- Outbound policy for auto provisioned sessions (if used) would be still possible via the use of templates along with peer ranges configuration.
- New BGP Capability Code (TBD) is defined to exchange information about BGP configuration within the new AFI/SAFI
- It is highly recommended to have an additional cli allowing to limit the scope of auto configured BGP sessions based on the expected peering range.
- Use of MD5 on the auto configured BGP sessions is highly recommended.

Agenda

A bit of history

An overview

Idea presentation

Discussion

BGP or IGP for autodiscovery

Original proposal was about using IGP. After that the following reasons caused to transition the idea on to contained to BGP tracks:

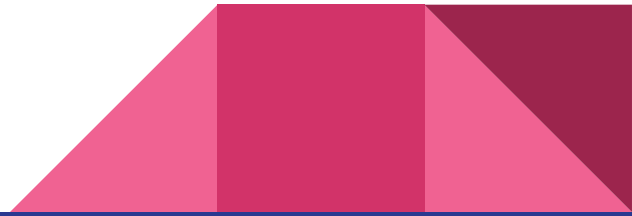
1. The solution should offer auto discovery for BGP peers without any relaying factor on IGP topology .. It should be equally easy to provision for single/multi-area IGPs as well as for peers where IGP is not running Carrier's of Carrier or IXes.
2. It should be possible to run where non link state IGP is used.
3. Requires reflooding entire LSA/LSP on ABRs when single info changes
4. Flooding information (up to 130 bytes per bgp speaker) via P routers seems an unnecessary architecture choice
5. Solution has to support sequence numbers
6. IGP implementation not to give BGP a burden to react to each flooding event even if delta is null should parse the message and only notify BGP about *NEW* events
7. We would need to develop and maintain all implementation flavourse of link state IGPs OSPFv2/OSPFv3/ISIS
8. Area linking would be a must as BGP topology can not be assumed to be congruent with BGP topology

How much ”auto” it is if needed to setup discovery session to RR ?

- This is _auto_ as you are only providing essentially just one IP address for any number of iBGP sessions for any SAFIs you like to be a part of.
- I see a good analogy to provide configuration of DNS servers, like NTP server, DHCP server, Syslog address or like IP address of the interface etc ....
- Still possible to flood the bootstraping route reflector ... aka as special case of service node. That would be a separate topic to discuss ... IMHO very much needed flooding functionality today.

Can we also extend it to discover peer's names ?

- I think this is an excellent idea !
- The only little drawback is that it would increase average PE autodiscovery chunk from 130 bytes to 164 bytes.
- Perhaps we could look at rearranging the attribute encoding to save some bytes



Thank you !