Considerations about Multicast BGP/MPLS VPNs Standardization

Thomas Morin – France Telecom
Ben Niven-Jenkins, Raymond Zang – British Telecom
Yuji Kamite – NTT Communications
Nicolai Leymann – Deutsche Telekom

Considerations about Multicast BGP/MPLS VPN Standardization

- Context
  - draft-ietf-l3vpn-2547bis-mcast
  - not (yet) a specification for a standard:
    - multiple options proposed for many of the building blocks
    - this does not provide interoperability
- Defining a core set of mandatory procedures is needed
  - leverage defined requirements
    - draft-ietf-l3vpn-ppvpn-mcast-reqts
- For each building block...
  - examine the different options
  - try to find which one best fits the requirements
- Authors of this draft make suggestions on the good candidates for being part of a set of mandatory procedures

- Two proposed mechanisms for auto-discovery:
  (1) BGP Auto-discovery
  (2) Discovery with PIM Hellos over shared tree
- Notes
  - (1) is consistent with unicast VPN operation
  - (2) is limited to shared trees (ASM multicast or MP2MP LDP) while (1) does not have such a limitation
  - (1) provides more control
    - → can be used to detect misconfiguration of shared trees ids/addresses
    - → can provide peer authentication (TCP MD5)
- Suggestions:
  - make BGP auto-discovery mandatory (1)
  - if needed, optionally provide (2) for compatibility purpose

- Proposed mechanisms for S-PMSI signalling:
  (1) UDP-based protocol and associated procedure
  (2) Procedure based on BGP extensions
- Notes
  - (2) can be used in an inter-AS option B context, in consistency with this model (no exchanges between PEs of different AS)
  - (2) can efficiently provide peer authentication
  - (1) is only for mVPNs having MI-PMSIs (more state)
  - (1) definitely is Yet Another Protocol™
    - *"[...] as far as possible, the design of a solution SHOULD carefully consider the number of protocols within the core network: if any additional protocols are introduced compared with the unicast VPN service, the balance between their advantage and operational burden SHOULD be examined thoroughly."* (5.210 of draft-ietf-l3vpn-ppvpn-mcast-reqts)
- Suggestions:
  - make BGP-based S-PMSI signalling mandatory (2)
  - implementations can provide (1) for compatibility purpose, but security implications of (1) should be closely studied, especially in an inter-AS context

S-PMSI switch-over procedure

- Two ways to switch traffic from an I-PMSI to an S-PMSI:
  (1) the source-side PE signals the S-PMSI, then sends on both trees for a while, each receiver-side PE chooses when to start accepting traffic on the new tree
  (2) the source-side PE signals the S-PMSI, wait for some time, then stops sending on old tree and starts sending on new tree
- Notes
  - (1) results in twice the bandwidth being used for some period of time
  - (1) is likely to introduce packet loss or duplicates
    (2) minimizes this risk
  - requirements state that *"[...] a multicast VPN solution SHOULD as much as possible ensure that client multicast traffic packets are neither lost nor duplicated, even when changes occur in the way a client multicast data stream is carried over the provider network"* (section 5.1.3 of requirement)
    - → provider's don't want that optimizing their backbone result in service degradation
- Suggestion:
  - make (2) the mandatory procedure

PE-PE multicast VPN routing [1/2]

- Proposed mechanisms are:
  (1) Full per-MVPN PIM peering across an MI-PMSI
  (2) Lightweight PIM peering across an MI-PMSI
  (3) Unicasting of PIM C-Join/Prune messages
  (4) Use of BGP for carrying C-Multicast routing
- Notes:
  - Scalability comments
    - contrary to “popular belief”, (1) and (2) require all PEs of an mVPN to process all messages
      - this processing requires parsing a PIM message, looking up the VRF MFIB, and possibly updating a timer
    - (3) put the burden of explicit tracking of receiver-side PE state, on the upstream PE
    - with (4) the equivalent of explicit-tracking is made by the RR (or spread in a hierarchy of them)
    - (4) advertise routes to all PEs, but:
      - these are easily discarded based on route-target (no VRF MRIB lookup)
      - if better is needed : use RT-Constraint to completely avoid this
    - (4) seems to provide all needed mechanisms to diminish/spread the load when scalability becomes a practical issue (e.g. see numbers in survey); many mechanisms are just inherited from BGP experience

PE-PE multicast VPN routing [2/2]

- ... notes (cont'd):
  - (1) and (2) require an MI-PMSI (more state in the core)
  - (4) enables an inter-AS mVPN deployment consistent with unicast VPN “Option B” (no exchanges between PEs in different ASs)
    - and can provide peer authentication
    - ”it is RECOMMENDED that a multicast VPN solution support means to ensure the integrity and authenticity of multicast-related exchanges across inter-AS or inter-provider borders”
  - (4) provides a good architectural and operational consistency
    - Extranet support is an example / Inter-AS is another
    - Consistency helps operational efficiency
  - Few return on experience on performance/impact of (4) as of today
  - No details on what (2) and (3) would precisely mean
- Suggestions
  - authors note that there are many strong points in favor of (4)
  - suggestion to keep (2) and (3) out of the spec at least until they are better defined/understood
  - implement both (4) and (1) / defer what to mandate ?

mVPN P-multicast trees encapsulation

- Multiple proposed encapsulation techniques
  - GRE/IP multicast w. PIM-SM ASM or SSM, or bidir-PIM
  - P2MP MPLS w. LDP
  - P2MP MPLS w. RSPV-TE
  - ...
- Notes
  - Different contexts, different needs
  - A new technique can be added or removed without any interoperability issue => not standardization issue at stake
- Suggestions
  - mVPN specifications should not unreasonably restrict the data plane technology that can be used
  - But no need to mandate an encapsulation technique
  - It is recommended that implementations support the multicast tree encapsulations techniques corresponding to widely used unicast VPN encapsulation techniques, namely: *mLDP, P2MP RSVP-TE and GRE/IP-multicast*

mVPN Inter-AS Deployments

- Two approaches are proposed:
  - Non-segmented inter-AS P-multicast trees
  - Segmented inter-AS P-multicast trees
- Notes:
  - A requirement:
    - *“a multicast VPN solution SHOULD provide inter-AS mechanisms requiring the least possible coordination between providers, and keep the need for detailed knowledge of providers' networks to a minimum”*
    - the segmented approach is helpful in this area
  - Choice of encapsulation technique
    - no coupling between different ASes
  - S-PMSI in Inter-AS
    - The segmented approach allows to keep the independence of the traffic-engineering decision made in different ASes
  - Different context, different needs
    - in an inter-AS / mono provider context, the non-segmented approach can be good enough
- Suggestion
  - Specifications should recommend implementing both

About deployments of early implementations

- There are deployments of early implementations
  - `draft-rosen-vpn-mcast`
  - `draft-raggarwa-l3vpn-2547-mvpn`
- Some of the suggestions are in line with these early implementations, and some differ.
- Authors' opinion
  - Run implementations of current specifications in parallel with early implementations (when an incremental modification is not possible)
  - Provide per VPN switching knob
    - → would happen during maintenance windows
  - A more complex update scheme ?
    - → maybe not worth the complexity...

Conclusion

- Main points
  - Security, especially in inter-AS
  - Consistency
    - with unicast VPN
    - of the overall mVPN architecture
  - Scalability
    - good to avoid the use MI-PMSI when not needed for the dataplane
    - what is the right tool to handle customer multicast routing load ?
  - Take deployments of early implementations into account
- Please react / discuss / comment
- Ask WG and specification authors to take these comments into consideration
- We plan to update/refine these suggestions
- Contributions welcome !

Thanks !