Overview of MVPN Developments

- Purpose:
  - Familiarize multicast ops group with recent MVPN activity
  - Get feedback on some controversial options
    - How well will they support enterprise multicast applications?
    - Do they ignore lessons learned from multicast experience?
  - Always remember: MVPN service provides *enterprise* multicast, not *Internet* multicast

Functionality of Existing Deployments

- Each VPN gets one default multicast *P-tunnel* through *P-network*
  - *P* for service *P*rovider, *PE* for *P*rovider *E*dge
- PEs attached to sites of same MVPN auto-discover each other through BGP
- Individual multicast *C-flows* can be dynamically assigned to P-tunnels
  - *C* for *C*ustomer of Provider, *CE* for *C*ustomer *E*dge

Technology of Existing Deployments

- P-tunnels are PIM-built source and/or shared trees connecting the PEs of a VPN
- PEs discover each other, and the P-tunnel identifier, via BGP
- C-packets encapsulated in GRE to be sent on P-tunnel
- For given VPN, PEs are PIM peers
  - "Overlay signaling" of C-multicast data is PE-PE PIM over the default P-tunnel

New MVPN Work

- Expand range of supported P-tunnel technologies
  - MPLS as well as PIM/GRE
    - Replace PIM on P-routers
- Allow all kinds of aggregation strategies
  - Enhance BGP auto-discovery to support general "bind C-flow to P-tunnel" capability
- Provide *option* to use BGP for "overlay signaling" instead of PE-PE PIM
  - Much of the controversy stems from this option

Generalized P-tunnels

- MPLS P-tunnels allowed
  - Not just PIM/GRE
  - LDP P2MP, LDP MP2MP, RSVP-TE
  - N.B.: For P-tunnels, PIM replaced by MPLS, *not* by BGP
- LDP tunnels are receiver-driven, old familiar paradigm, but with simplifications
- RSVP-TE tunnels are a bit strange in this context (more later)

Aggregation

- Allows general set of tools for binding C-flows to P-tunnels, including aggregation:
  - Non-default P-tunnels not restricted to one C-flow
  - With major MPLS enhancement *(upstream-assigned labels)*, can aggregate multiple VPNs
  - No real knowledge about how best to use this, if at all.
  - Controversy over just how useful this is (feedback?)
  - In abstract, seems like scaling improvement, not clear how useful in practice

What's Strange about RSVP-TE?

- All signaling is head-end initiated
- To assign C-flow to non-default P-tunnel, explicit tracking is *required*
  - Not required for other P-tunnels
- Leaf can't even remove itself without signaling to head end
- P routers must keep track of downstream nodes on RSVP-TE tree
- ATM-like scaling properties seem problematic
- No real alternative when TE is really needed, e.g., for guaranteed bandwidth

Option to use BGP Instead of PIM for Overlay Signaling

- Use BGP, not PIM, to send Join/Prunes from PE to PE
  - New address family to represent and distribute PIM states
- In theory, improves scale in some dimensions:
  - Assuming Route Reflector, eliminates some amount of PE-PE adjacency state
  - Eliminates periodic transmissions:
    - Hellos
    - J/P states that don't change (of course, this is helpful if things are static, less so if things are always changing)
  - (None of these are proven bottlenecks though)

Neat Features of the BGP Option

- Join(S,G)s from different PEs to same upstream PE are comparable BGP routes
  - RR gets a route from each PE receiver
  - RR passes along only one
  - By default, no explicit tracking
- Automated filtering so that only selected upstream PE gets Join
- Backbone not treated as LAN
- Provides unified L3VPN control plane

Bogus Claims about the BGP Option

- Eliminates need for SP to manage PIM
  - NOT! PEs still run PIM with CE.
  - Question: Is PE-CE PIM a bottleneck? If so, BGP signaling addresses the wrong issue.
- PE-PE PIM must run on emulated LAN, which requires full mesh per PE per VPN of PIM trees
  - NOT! See “partitioned MDT”, “PORT”
- Only way to get rid of Hello overhead
  - NOT! Even deployed MVPN uses BGP, not PIM Hellos, for auto-discovery

Some Not So Good Features of the BGP Option

- Latency increased, less predictable
  - Two TCP hops
    - each one with processing, flow control, congestion control, possibly long queues of unicast routing data
- BGP thrashing can now be caused directly by enduser *(not IT dept.)* IGMP activity:
  - BGP updates directly related to Join/Prunes
    - DoS attack risked
  - Some BGP dampening possible, at expense of increased latency or more unwanted traffic or more unwanted state
  - Is dedicated RR required?

Sparse Mode is Handled "Differently"

- Difficult to replicate SM exactly in BGP
  - Very hard to replicate Prune(S,G,R) states in BGP
  - Don't want data driven events
- Approach:
  - Once some PE joins a source tree:
    - Use BGP to generate "source active" advertisements
    - Force everyone to the source tree
  - Replace data-driven events by timers
  - Should work, but additional signaling mechanism
- Questionable whether new stuff in support of SM is worthwhile

Ugly Combination: RSVP-TE, BGP, and Aggregation

- BGP signals all PEs in VPN:
  - "I want to move C-(S,G) to this new tunnel"
  - Leaf PEs signal back in BGP, "count me in"
- RSVP-TE signaling done from head-end to add leaves to RSVP-TE P2MP LSP
- If leaf no longer has receivers:
  - leaf uses BGP to tell head end,
  - head end uses RSVP-TE to prune leaf
- To change aggregation, need BGP to advertise P-tunnel identifier, RSVP-TE to create new P-tunnel

Feedback Needed from MBONED

- Will all multicast applications in the enterprise continue to work as expected if BGP-based C-multicast routing is deployed?
- Where if anywhere is multicast deployment experience being disregarded?
- Are the real bottlenecks properly addressed?