Reliable PIM Registers
draft-anish-reliable-pim-register

Stig Venaas,

Toerkess Eckert,

Anish Peter,

Robert Kebler,

Vikram Nagarajan

IETF 101 - March-2018

Motivation
to be added to next rev of draft

- draft-acg-mboned-deprecate-interdomain-asm
  - Deprecate ASM/PIM-SM interdomain -> Interdomain MSDP too
  - MSDP intradomain for MSDP mesh group unaffected
- MSDP mesh-group compared to PIM RP mesh group (RFC4610)
  - MSDP only IPv4, RFC4610 IPv4/IPv6, but MSDP has better performance, operational features
  - Reliable transport (TCP): Works reliable especially under bursts (large #state)
    - Even without anycast-RP: Big video server (large #(S,G)) to RP: Datagram PIM Hello issues
      - Recommendation: make FHR be RP, use MSDP to overcome PIM Register issues
  - MIB, YANG model, cache (which RP sent which (S,G)), limits (#state), filter (AC) – better Mgmt
- Want to have TCP (== PORT) based RFC4610 variant
  - also improve (see example above) FHR-DR<->RP reliability/performance
  - Finally deprecate MSDP (without loosing reliability, performance, manageability)
  - Define MSDP anycast equivalent YANG model for reliable PIM register

PIM Registers – How it is today

- First-Hop-Router (FHR-DR) tunnels via PIM (unicast) “Register” message/encap sources (S,G) packets to.
- PIM registers serve two purposes
    - It helps FHR to inform that it is getting traffic for a given (source, group).
    - It helps in avoiding initial packet loss.
- Each individual S, G is “ack’ed” with Register Stop
    - Register-stops prevent FHR from sending data Registers
- Subsequent to this, NULL-Registers are used to maintain the aliveness of the source
- Many Multicast applications are tolerant to initial packet loss.
- Many intradomain Multicast applications are not ssm capable.
    - Forcing networks to run on asm mode.

Observations

- PIM Null-Register
  - Is soft-state based
  - Packet format does not allow state refresh for multiple flows in the same message
- PIM register-stop messages inherit all the problems in Null-Register messages
- In the FHR, if Register-Stop times-out, its expected to resort to Packet-Register's (RFC defaults to 60+5s).
  - This could happen even if one RS-message gets dropped.

Reliable Registers

- Reliable-Registers would support a reliable transport between FHR and RP
- Create a “targeted” adjacency between FHR and RP
  - These routers form adjacency.
  - Sends PIM Hellos with normal Hello Options to advertise capabilities
  - Can use Anycast-RP address to find closest RP
- Use TCP/SCTP
  - Some of the same encoding as RFC 6559 (PIM PORT)
  - Reliability and Flow control
  - New messages created to notify of new active source
- FHR sends message to RP to add/remove active sources

Targeted Hellos

- As per present spec, PIM hellos are link-level
- This draft extends that to supported pim neighbors over multiple hops reached via its known unicast address
- FHR router upon learning an RP (could be anycast-RP) address would transmit targeted hellos
- RP could respond to those targeted hellos
- From these hellos RP and FHR would learn the port capability and could start with reliable-registers
- RP when responding to targeted hello would use its unique address and would add its other address (including anycast addresses) in its secondary address TLV's.
- New TLV would be added for targeted neighbor properties/capabilities.
- Hellos will have TLV's as specified by PORT for reliable connection setup

Connection Setup

- Based on hello FHR and RP would learn its peers PORT capabilities.
- Once adjacency is formed, RP would connect to FHR to form the reliable connection.
- PORT Keep-alive could be used to maintain aliveness of session.

Hard-State Register Messages

- Stream-Register Message send by FHR
- Similar to a NULL-register
- FHR can withdraw the register when it finds doing so is appropriate (KAT trigger)
- To withdraw, set withdraw flag in the same register message

Anycast RP

- FHR would discover nearest RP by means of sending targeted hellos to anycast address.
- Reliable full mesh connection among the anycast RP-Set.
- Redistribution of source information
  - RP’s would transmit stream-register messages received from FHR to all the other any-cast peers.
  - When a new anycast-RP connection is setup, an RP would send to the peers all the stream-registers it had learned from FHR.

Management Considerations

- Only mandatory configuration needed is an enable/disable knob for reliable register/packet registers (No need configure peers)
- Incremental deployment is possible
- Feature support needed only on RP and FHR

Security Considerations

- Can help improve the pim register attack vulnerability
- TCP sync attack vulnerability is limited due to targeted hello session
- Targeted hellos are introduced, which may in future have an authentication extension for FHR

Next steps

- Please review, discuss on mailing list
- Call for working group adoption (IETF102 ?!)
- Open work
  - Policy for register encap of actual data packet (not null register)
    - What do we want ?
  - YANG model
    - Creates ask for manageability features of registers (cache, limit, filter)
- Possible extensions
  - Source control (RP based permit/deny of (S,G) via register msg
    - Simple oversight for original PIM register message mechanism (next rev)

Thank You

Opinions

&

Clarifications

Summary: FHR <-> RP

- FHR and RPs configured to support this feature (part of port ? TBD)
- FHR learns RP as usual (configuration or discovery via BSR)
- FHR that is DR exchanges new directed (unicast) PIM Hello (datagram) with RP (FH-DR start)
- After RP sees directed PIM Hello, opens TCP Reliable Register (PORT-Register) to FHR-DR
- Two routers who are both DR and RP: determine which is TCP initiator
  - same method as in PORT (RFC6559)
- Reset situation via Directed PIM Hello with updated GenID, rebuild TCP connection
  - Reconfiguration, redundancy failover (route processor), …
- Timeout (various error conditions) -> rebuild after directed PIM hello rediscovers neighbor mutually

Summary: RP <-> RP (Anycast RP)

- Mesh-group-logic: like RFC4610
  - Full mesh of onfigured RP-neighbors
    - Remember per (S,G) whether receeived from mesh-group tunnel peer or FHR-DR tunnel peer
    - Forward only FHR-DR learned (S,G) to mesh-group-peer
    - For diagnostics (not protocol) good to remember exact neighbor (S,G) was learned from)
- Anycast FHR-DR to RP relies on anycast to unicast resolution via directed PIM Hello
  - Learned/configured peer address can be anycast (from PR).
  - Directed PIM Hello signals “primary address” PIM option so other side can learn unicast IP address for TCP connection
- Backward compatibility – MSDP peers, legacy PIM Register peers
  - Defined. Not sure if MSDP should still be mentioned, legacy PIM peer support more important for migration.
    Easier to change RP set to be capable of new mechanisms than all FHR-DR at once)

Protocol: New Hello Optional TLV's (IPv4/IPv6)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type = H1 (for alloc)     |           Length = 4          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|R|                        Reserved                   |  Exp  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Protocol: Port Register Message TLV (IPv4/IPv6)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type = P1 (for alloc)     |        Message Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                   | Exp.  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|N|A|                      Reserved-1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            src addr-1                         z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                            grp addr-1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                            2, 3, . . .                        z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|N|A|                      Reserved-n                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            src addr-n                         z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                            grp addr-n                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Protocol: Port Register Stop Message TLV (IPv4/IPv6)

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type = P2(for alloc)      |       Message Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                  | Exp.   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved-1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            src addr-1                         z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                            grp addr-1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                           2, 3, . . .                         z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved-n                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            src addr-n                         z
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
z                            grp addr-n                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```