

# MSDP YANG

## draft-ietf-pim-msdp-yang-00

PIM WG

IETF98# Chicago

Xufeng Liu  
Zheng(Sandy) Zhang  
Anish Peter  
Mahesh Sivakumar  
Feng Guo  
Pete McAllister

# MSDP YANG

- The WG 00 version is a status update of draft-zhang-pim-msdp-yang-02 that is presented in Seoul meeting.
- According to RFC3618 [Multicast Source Discovery Protocol (MSDP)]
- Scope covers MSDP protocol and commonly used configuration
- Defines a YANG data model for MSDP configuration and operation
  - High-level structure
  - Configuration attributes
  - Operational state attributes
  - Statistics attributes
  - RPC attributes

# MSDP YANG

- This YANG module is based on the consensus in the YANG multicast design team.
- Archive: <http://www.ietf.org/mail-archive/web/yang-multicast/current/maillist.html>
- Wiki: <http://trac.tools.ietf.org/wg/pim/trac/wiki/yang>
- This draft is versioned on github: <https://github.com/mcallisterjp/pim-yang/>

# MSDP YANG

Next to do:

- Adjust the reference of key-chain because of some changes in the models we depend on (draft-ietf-rtgwg-yang-key-chain).
- Revise some spelling mistakes.
- Need YANG doctors review.

Thanks!

# MSDP configuration

augment /rt:routing/rt:control-plane-protocols:

+--rw msdp!

+--rw global

.....

+--rw peers

.....

Two level:

Global: General configuration for the entire protocol.

Peer: Special configuration for every MSDP peer.

```
+--rw msdp!  
  +--rw global  
    | +--rw connect-source? if:interface-ref  
    | +--rw default-peer! {global-default-peer}?  
    | | +--rw peer-addr    -> ../.././peers/peer/address  
    | | +--rw prefix-policy? string {global-default-peer-policy}?  
    | +--rw originating-rp  
    | | +--rw interface? if:interface-ref  
    | +--rw sa-filter  
    | | +--rw in? string  
    | | +--rw out? String  
    | +--rw sa-limit? uint32 {global-sa-limit}?  
    | +--rw ttl-threshold? uint8
```

- Connect-source: The interface is to be the source for the TCP connection.

- Default-peer: The default peer accepts all MSDP SA messages. The reverse path forwarding (RPF) check on SA messages can fail, and no SA messages are accepted. We can use a default peer and bypass RPF checks.

- Originating-rp: This parameter can be used to define a unique IP address for the RP of each MSDP peer.

- Sa-filter: Specifies an access control list (ACL) to filter source active (SA) messages.

- Sa-limit: A limit on the number of SA entries accepted.

- Ttl-threshold: Maximum number of hops data packets can traverse before being dropped.

# MSDP configuration

- Authentication: Commonly used authentication attributes.
- Enable: Same usage with BGP protocol.
- Mesh-group: The mesh group that the peer belongs.
- Peer-as: Peer's autonomous system number (ASN).
- Timer: The timer definition according to RFC3618.

```
+--rw peers
  +--rw peer* [address]
    +--rw address      inet:ipv4-address
    +--rw authentication
      | +--rw (authentication-type)?
      |   +--:(key-chain) {peer-key-chain}?
      |     | +--rw key-chain?  key-chain:key-chain-ref
      |     | +--:(password) {peer-key-chain}?
      |     |   +--rw key?      string
      +--rw enable?        boolean {peer-admin-enable}?
      +--rw connect-source? if:interface-ref
      +--rw description?   string {peer-description}?
      +--rw mesh-group?    string
      +--rw peer-as?       string {peer-as}?
      +--rw sa-filter
        | +--rw in?  string
        | +--rw out? String
      +--rw sa-limit?  uint32 {peer-sa-limit}?
      +--rw timer
        | +--rw connect-retry-interval?  uint16 {peer-timer-connect-retry}?
        | +--rw holdtime-interval?       uint16 {peer-timer-holdtime}?
        | +--rw keepalive-interval?      uint16 {peer-timer-keepalive}?
      +--rw ttl-threshold?  uint8
```

# MSDP state

augment /rt:routing-state/rt:control-plane-protocols:

+--ro msdp!

+--ro global

.....

+--ro peers

.....

+--ro sa-cache

.....

Three levels:

Global: The same as configuration.

Peer: Include the peer configuration and statistics.

Sa-cache: SA cache state attributes.

```
+--ro sa-cache
  +--ro entry* [group source-addr]
    +--ro group      inet:ipv4-address
    +--ro source-addr  union
    +--ro origin-rp* [rp-address]
      | +--ro rp-address  inet:ip-address
      | +--ro is-local-rp?  boolean
      | +--ro sa-adv-expire? uint32
    +--ro up-time?      uint32
    +--ro expire?       uint32
    +--ro holddown-interval? uint32
    +--ro peer-learned-from? inet:ipv4-address
    +--ro rpf-peer?     inet:ipv4-address
```

Group: The group address of this sa cache.

Source-addr: The source addr of this sa cache.

Origin-rp: The rp information.

Up-time: The up time of this sa cache.

Expire: The expire time of this sa cache.

Holddown-interval: Holddown timer value for SA forwarding.

Peer-learned-from: The address of peer that we learned this SA from.

Rpf-peer: RPF peer.



# MSDP state

## Peer state:

- Session-state: Per peer state attributes for MSDP.
- Elapsed-time: Elapsed time for being in a state.
- Connect-retry-expire: Connect retry expire time of peer connection.
- Hold-expire: Hold expire time of peer connection.
- Is-default-peer: If this peer is default peer.
- Keepalive-expire: Keepalive expire time of this peer.
- Reset-count: The reset count of this peer.
- Statistics: Include the statistics information of received and sent. And other relative information.

```
+--ro peers
  .....
  | +--ro session-state?    enumeration
  | +--ro elapsed-time?    uint32
  | +--ro connect-retry-expire? uint32
  | +--ro hold-expire?    uint32
  | +--ro is-default-peer?  boolean
  | +--ro keepalive-expire? uint32
  | +--ro reset-count?    uint32
  | +--ro statistics
  |   +--ro discontinuity-time? yang:date-and-time
  |   +--ro error
  |     | +--ro rpf-failure? uint32
  |     +--ro queue
  |       | +--ro size-in?  uint32
  |       | +--ro size-out? uint32
  |       +--ro received
  |         | +--ro keepalive? yang:counter64
  |         | +--ro notification? yang:counter64
  |         | +--ro sa-message? yang:counter64
  |         | +--ro sa-response? yang:counter64
  |         | +--ro sa-request? yang:counter64
  |         | +--ro total?    yang:counter64
  |         +--ro sent
  |           +--ro keepalive? yang:counter64
  |           +--ro notification? yang:counter64
  |           +--ro sa-message? yang:counter64
  |           +--ro sa-response? yang:counter64
  |           +--ro sa-request? yang:counter64
  |           +--ro total?    yang:counter64
```

# MSDP rpc

rpcs:

```
+---x msdp-clear-peer
| +---w input
|   +---w peer-address? inet:ipv4-address
+---x msdp-clear-sa-cache {rpc-clear-sa-cache}?
+---w input
+---w entry!
| +---w group      inet:ipv4-address
| +---w source-addr? union
+---w peer-address? inet:ipv4-address
+---w peer-as?     string
```

RPC includes the operation of clearing peer and sa-cache.