RIFT YANG

- This data model is defined according to [draft-ietf-rift-rift].

- This model includes protocol configuration, state information, some notifications and RPC(TBD).

- Some features are added to enhance protocol.
RIFT YANG update 01 version

- Change the “community” format to list
- Reference the common policy defined in [draft-ietf-rtgwg-policy-model]
RIFT configuration

augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
+--rw rift!
   +--rw node-info
      |  +--rw systemid     systemid
      |  +--rw address?     inet:ip-address
      |  +--rw level-flag?  enumeration
      |  +--rw level-value?  level-value
      |  +--rw name?        string
      |  +--rw pod?         uint16
      |  +--rw hal?         level-value
      |  +--rw ol?          boolean {overload}?
      |  +--rw ztp?         boolean {ztp-mode}?
      |  +--rw bfd?         boolean {bfd}?
      |  +--rw default-originate?  boolean {default-route-originate}?
      |  +--rw flood-reducing?  boolean {flood-reducing}?
      |  +--rw interfaces*   [local-id]
      |  |  +--rw local-id   uint32
      |  |  +--rw name?      if:interface-ref
      |  |  +--rw intf-type
      |  |  +--rw link-type? enumeration
      |  +--rw community
      |  |  +--rw community* [type value]
      |  |  +--rw type       enumeration
      |  |  +--rw value      uint64
      |  |  +--rw action?    enumeration
      |  +--rw policy-info [policy]?
      |  |  +--rw policy*    [name]
      |  |  |  +--rw name    string

- The configuration includes node info, interface info, policy.
- Base node configuration includes node systemid, address, node level, etc.
- Policies include community and policy-guide-information.
- Some features can be used to enhance protocol, such as BFD, flooding-reducing, overload bit and default-route-origination. etc.
RIFT State includes neighbor, database and kv-store information.
RIFT state

```
++-ro database
  | ++-ro ties* [tie-index]
  |   | ++-ro tie-index   uint32
  |   | ++-ro database-tie
  |     | ++-ro originator?  systemid
  |     | ++-ro direction
  |     |   | ++-ro direction-type?  enumeration
  |     | ++-ro type
  |     |   | ++-ro tie-type?  enumeration
  |     | ++-ro link-type
  |     |   | ++-ro link-type?  enumeration
  |     | ++-ro seq?         uint32
  |     | ++-ro lifetime?     uint16
  |     | ++-ro tie-node
  |     |   | ++-ro layer?   level-value
  |     |   | ++-ro nbr-info
  |     |   |   | ++-ro nbr-addr?  inet:ip-address
  |     |   |   | ++-ro distance?  uint32
  |     |   |   | ++-ro nbr-layer?  level-value
  |     |   |   | ++-ro cost?     uint32
  |     |   | ++-ro link-set* [local-id]
  |     |     | ++-ro local-id  uint32
  |     | ++-ro tie-prefix
  |     |   | ++-ro prefix?  inet:ip-prefix
  |     |   | ++-ro cost?     uint32
  |     | ++-ro (algorithm-type)?
  |     | +-:(spf)
```

```
++-ro kv-store
  | ++-ro kvs* [kvs-index]
  |   | ++-ro kvs-index  uint32
  |   | ++-ro kvs-tie
  |     | ++-ro originator?  systemid
  |     | ++-ro direction
  |     |   | ++-ro direction-type?  enumeration
  |     | ++-ro type
  |     |   | ++-ro tie-type?  enumeration
  |     | ++-ro link-type
  |     |   | ++-ro link-type?  enumeration
  |     | ++-ro seq?         uint32
  |     | ++-ro lifetime?     uint16
  |     | ++-ro tie-node
  |     |   | ++-ro layer?   level-value
  |     |   | ++-ro nbr-info
  |     |   |   | ++-ro nbr-addr?  inet:ip-address
  |     |   |   | ++-ro distance?  uint32
  |     |   |   | ++-ro nbr-layer?  level-value
  |     |   |   | ++-ro cost?     uint32
  |     |   | ++-ro link-set* [local-id]
  |     |     | ++-ro local-id  uint32
  |     | ++-ro tie-prefix
  |     |   | ++-ro prefix?  inet:ip-prefix
  |     |   | ++-ro cost?     uint32
```
RIFT notification

Unexpected TIE and neighbor's layer error should be notified.
• Any comment is welcomed ^^

Thanks!