A YANG Data Model for Routing Policy Management
draft-ietf-rtgwg-policy-model-02

Yingzhen Qu (yingzhen.qu@huawei.com)
Jeff Tantsura (jefftant.ietf@gmail.com)
Acee Lindem (acee@cisco.com)
Xufeng Liu (xufeng_liu@jabil.com)
Anees Shaikh (aashaikh@google.com)

Status of this draft

- -01 version expired in October, 2016
- Jeff Tantsura discussed the issues with OpenConfig, agreement was IETF routing design team would take over the draft and progress it
- Anees remains as co-author
- Other original authors became contributors

Major Changes

- NMDA compliant
- Imports IETF modules
- Reduced grouping nesting
- Changed to IETF YANG style, e.g., enumerations use lowercase letters

Tree Structure

```
+--rw routing-policy
  +--rw policy-definitions
    +--rw policy-definition* [name]
      +--rw name        string
      +--rw statements
        +--rw statement* [name]
          +--rw name        string
          +--rw conditions
          |    ...
          +--rw actions
              ...
```

- Policies are expressed as a sequence of top-level policy definitions each of which consists of a sequence of policy statements.
- Policy statements in turn consist of simple condition-action tuples.
- Conditions may include multiple match or comparison operations, and similarly, actions may effect multiple changes to route attributes, or indicate a final disposition of accepting or rejecting the route.

Defined Sets for Policy Matching

```
+--rw routing-policy
    +--rw defined-sets
    |  +--rw prefix-sets
    |  |  +--rw prefix-set* [name]
    |  |     +--rw name      string
    |  |     +--rw mode?     enumeration
    |  |     +--rw prefixes
    |  |        +--rw prefix* [ip-prefix masklength-range]
    |  |           +--rw ip-prefix         inet:ip-prefix
    |  |           +--rw masklength-range   string
    |  +--rw neighbor-sets
    |  |  +--rw neighbor-set* [name]
    |  |     +--rw name      string
    |  |     +--rw address*  inet:ip-address
    |  +--rw tag-sets
    |     +--rw tag-set* [name]
    |        +--rw name       string
    |        +--rw tag-value*  tag-type
```

- prefix sets - Define a set of IP prefixes, each with an associated mask range or exact length.

- neighbor sets - Define a set of neighboring nodes by their IP addresses. These sets are used for selecting routes based on the neighbors advertising the routes.

- tag set - Define a set of generic tag values that can be used in matches for selecting routes.

Policy Conditions

```
+--rw routing-policy
   +--rw policy-definitions
      +--rw policy-definition* [name]
         +--rw name        string
         +--rw statements
            +--rw statement* [name]
               +--rw conditions
               | +--rw call-policy?
               | +--rw install-protocol-eq?
               | +--rw match-interface
               | | +--rw interface?
               | +--rw match-prefix-set
               | | +--rw prefix-set?
               | | +--rw match-set-options?
               | +--rw match-neighbor-set
               | | +--rw neighbor-set?
               | | +--rw match-set-options?
               | |     match-set-options-restricted-type
               | +--rw match-tag-set
               |    +--rw tag-set?
               |    +--rw match-set-options?
                        match-set-options-restricted-type
```

Policy statements consist of a set of conditions and actions (either of which may be empty). Conditions are used to match route attributes against a defined set (e.g., a prefix set), or to compare attributes against a specific value.

Match conditions may be further modified using the match-set-options configuration which allows operators to change the behavior of a match. Three options are supported:

- ALL - Match is true only if the given value matches all members of the set.
- ANY - Match is true if the given value matches any member of the set.
- INVERT - Match is true if the given value does not match any member of the set.

Policy Actions

```
+--rw routing-policy
    +--rw policy-definitions
        +--rw policy-definition* [name]
            +--rw statements
                +--rw statement* [name]
                    +--rw actions
                        +--rw policy-result?   policy-result-type
```

When policy conditions are satisfied, policy actions are used to set various attributes of the route being processed, or to indicate the final disposition of the route, i.e., accept or reject.

Next Steps

- Collect/address comments
- Update examples

Question?

Thank You