I2RS RIB Route Example

Sue Hares
Hackathon
IETF 96
Goals

Network of routers simulated using mininet/mininetx

Local config
static route
IP table

Router

Router

confd

Zebra

Quagga

IP Table

RPC

<route-add>

NETCONF

with i2RS
RIB + FB-RIB

i2RS Client

config Client

yangcli-pro

CLI/GUI

CLI/GUI
Extended RIB
+ FB-RIB

NETCONF

Router with i2RS agent

NETCONF

with i2RS

Router

Router

ospf/bgp

Local config
static route
IP table
I2RS RIB Example

- First Case
  - 128.2/16 with nexthop 1 – added by netconf config
  - 128.2/16 with nexthop 2 – added by I2RS RIB client 1
  - DDOS attack causes you to overwrite NETCONF config with I2RS RIB route via client 2
Ephemeral Additions

config true;

config false;

Derived state
Normal +
Ephemeral state

I2RS Agent

I2RS Agent

intended config

Ephemeral Intended

applied config
(normal + ephemeral)


candidate

running

startup
Thermostat Model RIB Equivalent

Running:
Route 128.1/16
Nexthop id 1 (192.1.1.1)

Scheduler
Client

intended config

applied config

config true;

config false;

Route 128.1/16
nexthop id 1 (192.1.1.1)
Route-installed-state Installed
Route + Ephemeral Route

I2RS Client 1
Config checker

I2RS Client 2
IPS application

NETCONF
config true;

config false;

Applied config

Route 128.2/16 nexthop id 1

intended config

Route 128.2/16 nexthop id 2

Derived State
Route 128.2/16 nexthop 2
Route-installed-state Installed
intf 1
RESTCONF Example

**RESTCONF Running Datastore Edit**

PUT /restconf/data/i2rs-rib/instance=1/rib=IPv4/route=128.2
/nexthop { “next-hop”:1}

**RESTCONF Ephemeral Datastore Edit of config=true**

PUT /restconf/data/i2rs-rib/instance=1/rib=IPv4/route=128.2
/nexthop?context=ephemeral
{ “next-hop”:2 }
Route

Index for route direct reference without prefix match; Main key.
Type: ipv4, ipv6, mpls, mac, interface

Type: v4 prefix match

Index for nexthop direct index without match
IPv4 prefix

module i2rs-rib { ....
  container routing-instance { ....
    list rib-list { ....
      list route-list {
        key “route-index”;
        leaf route-index {
          type uint64;
          mandatory true;
        }
        leaf route-type {
          type route-type-def;
          mandatory true;
        }
        Container match {
          choice rib-route-type { ....
            leaf destination-ip-v4-prefix {
              type inet:ipv4-prefix;
              mandatory true;
            }
          }
        }
        leaf nexthop-id {
          type uint32;
          mandatory true;
        }
        leaf next-hopo-ipv4-address {
          type inet:ipv4-prefix;
          mandatory true;
        }
      }
    }
  }
  container route-statistics {
    leaf route-installed state {
      type route-installed-state def;
      config false;
    }
  }
}

Add Ephemeral true;

Defined as:
Installed, uninstall

Type: ipv4, ipv6, mpls, mac, interface
RESTCONF Running Datastore Edit
PUT /restconf/data/i2rs-rib/instance=1/rib=IPv4/route=128.2
   /next-hop { “next-hop”:1 }

RESTCONF Ephemeral Datastore Edit of config=true
PUT /restconf/data/i2rs-rib/instance=1/rib=IPv4/route=128.2.1
   /next-hop?content=ephemeral { “next-hop”:2 }

RESTCONF Ephemeral Datastore Edit of config=false
GET /restconf/data/i2rs-rib/instance=1/rib=IPv4/route=128.2.1
   /next-hop=2/route-installed-state/datastore=ephemeral
   { “route-installed-state”: Installed }
RIB-DM Design issues

- Next-hop protection
- Overlapping routes