I2RS Protocol

I2RS built for High performance

Sue Hares

Not the Pizza box CLI
I2RS Protocol

- Re-use Protocol
  - 5 Drafts
    draft-ietf-netconf-call-home
    draft-ietf-netconf-yang-library
    draft-ietf-netconf-yang-patch
    draft-ietf-netconf-yang-push
I2RS Protocol

- Data-Model Driven
  - Data drives function
  - Yang describes data model
  - Data model rather than CLI becomes common unit
What is Ephemeral Configuration?

The Journey to Ephemeral

- CLI Maestro
- NETCONF-Yang (RFC6244)
- Opstate
- Ephemeral
Traditionally CLI configurations have been considered:

- **Candidate** – Added config
- **Running** – What system is running
- **Start-up** – What system reboots to

**Operational data** – what system creates
RFC6244
Data Model view

client

Netconf
server

Config DataBase

Intended Config

Ephemeral config

Routing Software

Applied Config

Derived State

System Software

Applied Config

Derived State

Meta Data

config

State data

notification

operation

YANG as Source Code

User Docs

Server Code

WEB Apps

Client Code

Test Code

CLI Code
Definitions from ietf-netmod-opstate-req

- **config true;**
  - intended config
  - applied config
  - Derived state

- **config false;**

- **candidate** → **running** → **startup**
Ephemeral Additions

- candidate
- running
- startup

config true;

- intended config
- Ephemeral Intended
- applied config (normal + ephemeral)

config false;

- Derived state
  Normal + Ephemeral state

Read Only
I2RS Code

Store in I2RS agent, but not in configuration files
Examples

- Thermometer
- Route add
- Filter Add
Simple Thermostat Example

module thermostat {
    // Configuration
    leaf desired-temp {
        type int32;
        units "degrees Celsius";
        description "The desired temperature";
    }

    // operational state
    leaf actual-temp {
        type int32;
        config false;
        units "degrees Celsius";
        description "The measured temperature";
    }
}
Normal Thermostat Model

candidate

Desired-Temp = 68

running

Desired-Temp = 68

Start-up

Desired-Temp = 65

Intended Config

Desired-Temp = 68

Applied Config

Desired-Temp = 68

Read Only

Derived state

Actual-temp = 68

config true;

config false;
module thermostat {
  //config
  leaf desired-temp {
    type int32;
    ephemeral true;
    units "degrees Celsius";
    description "The desired temperature";
  }
  //operational State:
  leaf actual-temp {
    type int32;
    config false;
    units "degrees Celsius";
    description "The measured temperature";
  }
}
Ephemeral Thermostat Model

candidate

Desired-Temp = 68

running

Desired-Temp = 68

Start-up

Desired-Temp = 65

Intended Config

Desired-Temp = 68

Intended Ephemeral

Desired-Temp = 72

Applied Config

Desired-Temp = 72

Read Only

config true;

Desired-Temp = 72

config false;

Derived state

Actual-temp = 72

I2RS Client

Desired-Temp = 72
RESTCONF Example

RESTCONF Running Datastore Edit

PUT /restconf/data/thermostat:desired-temp

{ "desired-temp": 68 }

RESTCONF Ephemeral Datastore Edit of config=true

PUT /restconf/data/thermostat:desired-temp?context=ephemeral

{ "desired-temp": 72 }
PUT /restconf/data/thermostat:desired-temp
Host: example.com
Content-Type: application/yang.data+json
{ "desired-temp": 68 }

HTTP/1.1 204 No Content
Date: Mon, 23 Apr 2016 17:04:00 GMT
Server: example-server
Last-Modified: Mon, 23 Apr 2017 17:04:00 GMT
ETag: b27480aed4
RESTCONF Temperature Set

PUT /restconf/data/thermostat:desired-temp?context=ephemeral
Host: example.com
Content-Type: application/yang.data+json
{ “desired-temp”: 72 }

HTTP/1.1 204 No Content
Date: Mon, 22 Apr 2016 18:04:00 GMT
Server: example-server
Last-Modified: Mon, 23 Apr 2016 18:04:00 GMT
ETag: b27480aeda4
NETCONF

<rpc-message-id=101>
  <xmlns="urn:ietf:params:xml:ns:base:1.0">
    <edit-config>
      <target>
        <running/>
      </target>
      <config>
        <top xmlns="http://exaple.com/schema/1.0/thermostat/config">
          <desired-temp>68</desired-temp>
        </top>
      </config>
    </edit-config>
  </xmlns>
</rpc>
<edit-config>
  <target> </running/> </target>
  <config>
    <top xmlns="http://example.com/schema/1.0/thermostat/config">
      <desired-temp> 68 </desired-temp>
    </top>
  </config>
</edit-config>

<rpc-rply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-rply>
NETCONF

<rpc-message-id=101>
  <xmlns="urn:ietf:params:xml:ns:base:1.0">
    <edit-config>
      <target>
        <ephemeral-datastore/>
      </target>
      <config>
        <top xmlns="http://example.com/schema/1.0/thermostat/config">
          <desired-temp> 72 </desired-temp>
        </top>
      </config>
    </edit-config>
  </rpc>
RESTCONF Temperature Set

<edit-config>
  <target> <ephemeral-datastore/></target>
  <config>
    <top xmlns="http://exaple.com/schema/1.0/thermostat/config">
      <desired-temp> 72 </desired-temp>
    </top>
  </config>
</edit-config>

<rpc-rply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-rply>
I2RS RIB Route add

- Circumstance
  - 128.2/16 with nexthop 192.11.1.1 – added by netconf config
  - 128.2/16 with nexthop 192.11.1.2 – added by I2RS Client

- How I2RS adds route
  - rpc route-add
  - (RIB, route-list {prefix, attributes, vendor-attributes, nexthop})
Ephemeral Route

**candidate**

- Route = 128.2/16
  - nexthop 192.11.1.1

**running**

- Route = 128.2/16
  - nexthop 192.11.1.1

**Start-up**

- Route = 128.2/16
  - nexthop 192.11.1.1

**Intended Config**

- Route = 128.2/16
  - nexthop 192.11.1.1

**Intended Ephemeral**

- Route = 128.2/16
  - Nexthop 192.11.1.2

**Applied Config**

- Rt = 128.2/16
  - nexthop 192.11.1.2

**Read Only**

- 128.2/16
  - Nexthop 192.11.1.2

**Derived state**

- Route 128.2/16
  - nexthop 192.11.1.2
  - installed
module i2rs-rib {
  ...
  container routing-instance {
    ...
    list rib-list {
      ...
      list route-list {
        key "route-index";
        uses route;
      }
    }
  }
}

operational data

Extensions

grouping route {
  description
  "The common attribute used for all routes;"
  uses routeg-prefix;
  container nexthop {
    uses nexthop;
  }

  container route-statistics {
    leaf route-state {
      type route-state-def;
      config false; /* operational state */
    }
    leaf route-installed state {
      type route-installed-state def;
      config false;
    }
    leaf route-reason {
      type route-reason-def;
      config false;
    }
  }

  container router-attributes {
    uses router-attributes;
  }

  container route-vendor-attributes
    uses route-vendor attributes;
}
Ephemeral Filter

candidate

Filter 1
ip-dest 192.5.5.1
action = drop

running

Filter 1
ip-dest 192.5.5.1
action = drop

Start-up

Filter 1
ip-dest 192.5.5.1
action = drop

Intended Config

Filter 1
ip-dest 192.5.5.1
action = drop

Intended Ephemeral

Filter 1
ip-dest 192.5.5.1
forward to 192.11.1.12

Applied Config

config true;

Filter 1: ip-dest 192.5.5.1
forward to 192.11.1.12

Read Only

config false;

Derived state

Filter 1: installed, 11 matches
ip-dest: 192.5.5.1
Action:forward to 192.11.1.12

I2RS Client

Send to IPS device
Filter:
ip-dest = 192.5.5.1
Forward to
192.11.1.12
I2RS Code

Store in I2RS agent, but not in configuration files

I2RS Client
- CLI/GUI
- yangcli-pro

config Client
- CLI/GUI
- Extended RIB + FB-RIB

NETCONF

Network of routers simulated using mininet/mininet

Router with i2RS agent
- ospf
- BGP

confd
- I2RS Agent

Zebra

Quagga

RPC

L2 Code

IP Table

Router
Git Hub information for Code

- https://github.com/i2rs-wg
  - WG documents, code, yang modules
- https://github.com/i2rs-wg/hackathon-code
  - Interim work repository @ietfs
- https://github.com/i2rs-wg/code-implementations
  - Ready for you to try
  - ConfD requires Cisco Dev listen
Capability Specification for NETCONF/RESTCONF
NETCONF (1)

- **Capability:** ephemeral-datastore
- **Overview:**
  - Not in intended to survive a reboot, and Never locked
  - Normal Case: Priority of Ephemeral Pane higher than configuration Pane. Error if two clients write same variable (priority scopes error)
  - No Rollback on ephemeral
  - Ephemeral under non-ephemeral; No non-ephemeral under Ephemeral
  - NETCONF <hello> - but no non-ephemeral under ephemeral modules, sub-trees, node
- **Dependencies:**
  - Yang: ephemeral flag, ephemeral-validation
  - Yang modules – must support notification of write conflicts (Config/ephemeral and Priority)
NETCONF (2)

- **New operations:**
  - Link-ephemeral `<target-config>`
  - Bulk-write – [Not sure if need or if rpc better approach]

- **Modifications**
  - `<get-config>` `<get>` - target changes
  - `<edit-config>` - `<merge-priority>` `<replace-priority>`
    - `<default-operations>`: `<merge-priority>` or `<replace-priority>`
    - `<error-option>` - “all-or-nothing” == “rollback-on-error”
  - `<unlock>` `<lock>` - not supported
  - `<confirmed commit>` - not supported
  - `<close-session>` `<kill-session>` - target change
  - `<Writeable-running>` and `<candidate>` – support ephemeral (?)
  - Validate – supports ephemeral data store with three key words:
    - Syntax, reduced, full-check
Route Add rpc

- Screen shot of route add rpc added here
Filter-Add RPC here

- Screen shot of filter-add rpc added here
RESTCONF (1)

- **Capability**: ephemeral-context

- **Overview**: 
  - Same as netconf except RESTCONF Context

- **Dependencies**: 
  - Yang: ephemeral flag, ephemeral-validation
  - Yang modules – must support notification of write conflicts (Config/ephemeral and Priority)
  - I2RS Yang modules support: Yang patch and Yang module library
RESTCONF (2)

- **Data resources**
  - `+restconf/data` – ephemeral data tree with edit collision features of timestamp and Entity Tag
  - Assumption: Entity can be split to client-priority

- **Modifications**
  - Options: provide indication of ephemeral state in data modules, sub-modules [ietf-netconf-yang-library]
  - HEAD – returns ephemeral or config context
  - GET - determines if ephemeral or config
  - POST/PUT/PATCH - context=ephemeral:
    - uses ephemeral rules + validity + priority + no config below ephemeral
  - DELETE – ephemeral context
  - Query – Allows to filter by ephemeral
  - Error/Notifications – must interact with pub/sub push [ietf-netconf-yang-push]
  - Log and traceability -
Hands-on questions
I2RS Code

Router with i2RS agent

I2RS Client

NETCONF with i2RS RIB + FB-RIB

yangcli-pro

config Client

CLI/GUI

Extended RIB + FB-RIB

NETCONF

<route-add>

Network of routers simulated using mininet/mininetx

Router

Local config static route IP table

Router

Router

confd

Zebra

Quagga

IP Table

L2 Code

RPC

BGP

ospf

Router

IP Table