Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

J. White
D. Black
Dell EMC
J. Leung
Intel Corporation

March 7, 2017

Baseline Switch Model
draft-wbl-rtgwg-baseline-switch-model-00

Abstract

[Insert abstract here]

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

Disclaimer - this is a -00 draft.

This is a normative profile for Baseline Switch Profile (send into IETF RTG) intended to be published as RFC on completion of DMTF spec to wrap Baseline Switch Profile.

2. What is a Redfish Baseline Switch?

The baseline switch profile contains basic system, interface, L2, and L3 configuration elements sufficient to set up the device for use in a controller based converged infrastructure environment.

The following list of IETF drafts, RFCs, and Redfish models will constitute the management interface to the baseline switch.

3. Core YANG RFCs

RFC6020 [1] provides the YANG modeling language definition.

RFC6991 [2] provides the Common YANG Data Types used by many other IETF YANG modules.

Interface management requires at set of RFCs to provide all relevant capabilities:

https://tools.ietf.org/html/rfc7223
https://tools.ietf.org/html/rfc7277
https://tools.ietf.org/html/rfc7224
https://tools.ietf.org/html/rfc7317

3.1. RFC7223 provides:

```
     +--rw interfaces
     |  +--rw interface* [name]
     |     +--rw name                        string
     |     +--rw description?                string
     |     +--rw type                        identityref
     |     +--rw enabled?                    boolean
     |     +--rw link-up-down-trap-enable?   enumeration
     +--ro interfaces-state
        +--ro interface* [name]
           +--ro name               string
           +--ro type               identityref
           +--ro admin-status       enumeration
           +--ro oper-status        enumeration
           +--ro last-change?       YANG:date-and-time
           +--ro if-index           int32
           +--ro phys-address?      YANG:phys-address
           +--ro higher-layer-if*   interface-state-ref
           +--ro lower-layer-if*    interface-state-ref
           +--ro speed?             YANG:gauge64
           +--ro statistics
              +--ro discontinuity-time    YANG:date-and-time
              +--ro in-octets?            YANG:counter64
              +--ro in-unicast-pkts?      YANG:counter64
              +--ro in-broadcast-pkts?    YANG:counter64
              +--ro in-multicast-pkts?    YANG:counter64
              +--ro in-discards?          YANG:counter32
              +--ro in-errors?            YANG:counter32
              +--ro in-unknown-protos?    YANG:counter32
              +--ro out-octets?           YANG:counter64
              +--ro out-unicast-pkts?     YANG:counter64
              +--ro out-broadcast-pkts?   YANG:counter64
              +--ro out-multicast-pkts?   YANG:counter64
              +--ro out-discards?         YANG:counter32
              +--ro out-errors?           YANG:counter32
```

3.2. RFC7277 adds:

```
     +--rw if:interfaces
       +--rw if:interface* [name]
          ...
          +--rw ipv4!
          |  +--rw enabled?            boolean
          |  +--rw forwarding?         boolean
          |  +--rw mtu?                uint16
          |  +--rw address* [ip]
          |  |  +--rw ip               inet:ipv4-address-no-zone
          |  |  +--rw (subnet)
          |  |     +--:(prefix-length)
```

```
          |  |     |  +--rw ip:prefix-length?   uint8
          |  |     +--:(netmask)
          |  |        +--rw ip:netmask?         YANG:dotted-quad
          |  +--rw neighbor* [ip]
          |     +--rw ip                    inet:ipv4-address-no-zone
          |     +--rw link-layer-address    YANG:phys-address
          +--rw ipv6!
             +--rw enabled?            boolean
             +--rw forwarding?         boolean
             +--rw mtu?                uint32
             +--rw address* [ip]
             |  +--rw ip               inet:ipv6-address-no-zone
             |  +--rw prefix-length    uint8
             +--rw neighbor* [ip]
             |  +--rw ip                    inet:ipv6-address-no-zone
             |  +--rw link-layer-address    YANG:phys-address
             +--rw dup-addr-detect-transmits?   uint32
             +--rw autoconf
                +--rw create-global-addresses?        boolean
                +--rw create-temporary-addresses?     boolean
                +--rw temporary-valid-lifetime?       uint32
                +--rw temporary-preferred-lifetime?   uint32
```

AND

```
 +--ro if:interfaces-state
    +--ro if:interface* [name]
       ...
       +--ro ipv4!
       |  +--ro forwarding?   boolean
       |  +--ro mtu?          uint16
       |  +--ro address* [ip]
       |  |  +--ro ip               inet:ipv4-address-no-zone
       |  |  +--ro (subnet)?
       |  |  |  +--:(prefix-length)
       |  |  |  |  +--ro prefix-length?   uint8
       |  |  |  +--:(netmask)
       |  |  |     +--ro netmask?         YANG:dotted-quad
       |  |  +--ro origin?          ip-address-origin
       |  +--ro neighbor* [ip]
       |     +--ro ip                    inet:ipv4-address-no-zone
       |     +--ro link-layer-address?   YANG:phys-address
       |     +--ro origin?               neighbor-origin
       +--ro ipv6!
          +--ro forwarding?   boolean
          +--ro mtu?          uint32
          +--ro address* [ip]
          |  +--ro ip               inet:ipv6-address-no-zone
```

```
             |  +--ro prefix-length    uint8
             |  +--ro origin?          ip-address-origin
             |  +--ro status?          enumeration
             +--ro neighbor* [ip]
                +--ro ip                    inet:ipv6-address-no-zone
                +--ro link-layer-address?   YANG:phys-address
                +--ro origin?               neighbor-origin
                +--ro is-router?            empty
                +--ro state?                enumeration
```

3.3. RFC7224 provides:

The set of YANG identity statement for the IANA defined interface types.

3.4. RFC7317 provides:

- o System Identification
- o System Time Date
- o NTP
- o DNS Client

System Identification

```
  +--rw system
  |  +--rw contact?          string
  |  +--rw hostname?         inet:domain-name
  |  +--rw location?         string
  +--ro system-state
     +--ro platform
        +--ro os-name?       string
        +--ro os-release?    string
        +--ro os-version?    string
        +--ro machine?       string
```

System Time

```
     +--rw system
     |  +--rw clock
     |  |  +--rw (timezone)?
     |  |     +--:(timezone-name)
     |  |     |  +--rw timezone-name?     timezone-name
     |  |     +--:(timezone-utc-offset)
     |  |        +--rw timezone-utc-offset?   int16
     |  +--rw ntp!
     |     +--rw enabled?   boolean
     |     +--rw server* [name]
     |        +--rw name                string
     |        +--rw (transport)
     |        |  +--:(udp)
     |        |     +--rw udp
     |        |        +--rw address    inet:host
     |        |        +--rw port?      inet:port-number
     |        +--rw association-type?   enumeration
     |        +--rw iburst?             boolean
     |        +--rw prefer?             boolean
     +--ro system-state
        +--ro clock
           +--ro current-datetime?      YANG:date-and-time
           +--ro boot-datetime?         YANG:date-and-time
```

DNS Client

```
     +--rw system
        +--rw dns-resolver
           +--rw search*    inet:domain-name
           +--rw server* [name]
           |  +--rw name    string
           |  +--rw (transport)
           |     +--:(udp-and-tcp)
           |        +--udp-and-tcp
           |           +--rw address    inet:ip-address
           |           +--rw port?      inet:port-number
           +--rw options
              +--rw timeout?    uint8
              +--rw attempts?   uint8
```

User Authentication

```
+--rw system
   +--rw authentication
      +--rw user-authentication-order*   identityref
      +--rw user* [name]
         +--rw name        string
         +--rw password?   ianach:crypt-hash
         +--rw authorized-key* [name]
            +--rw name         string
            +--rw algorithm    string
            +--rw key-data     binary
```

4. Additional YANG models

In addition to the above RFCs, the baseline switch models needs to cover:

- o VLANs
- o ACLs
- o Syslog

The following lists of IETF drafts sets our recommendation to cover the above three areas.

4.1. VLAN and interface extensions:

To handle VLANs and with related interface configuration the following YANG models are under evaluation.

- o https://tools.ietf.org/html/draft-ietf-netmod-intf-ext-yang-03
- o https://tools.ietf.org/html/draft-wilton-intf-vlan-yang-00.txt ## ACL To handle ACL configuration the following YANG model is under consideration.
- o https://tools.ietf.org/html/draft-ietf-netmod-acl-model-09

4.2. Syslog

To handle configuration and access to syslog the following YANG model is under consideration.

- o https://tools.ietf.org/html/draft-ietf-netmod-syslog-model-11

5. Applicable Redfish system management models

The following standard Redfish systems management models apply to the baseline network switch profile. Reference: Redfish schema index [3]. The use of these Redfish management models allows a converged infrastructure manager to have a consistent view of server, storage and network systems.

- o Chassis
- o ComputerSystem
- o Manager
- o ManagerAccount
- o Power
- o Thermal
- o SoftwareInventory plus UpdateService
- o Event configuration using Event, EventDestination, and Event Service
- o Access to logs using LogEntry, and LogService
- o Management interface configuration using EthernetInterface and related
- o Console configuration using SerialInterface
- o PrivilegeRegistery and Privileges

Where YANG and Redfish overlap, the commonality of YANG vs Redfish is TBD.

6. Overall Baseline Switch Profile Structure

```
./redfish/v1/Systems
./redfish/v1/Chassis
./redfish/v1/NetworkDevices/BaselineSwitch/...
... other redfish resource blocks...
(resource from RFCs and Redfish bullet list, above)
```

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. URIs

[1] https://tools.ietf.org/html/rfc6020

[2] https://tools.ietf.org/html/rfc6991

[3] http://redfish.dmtf.org/redfish/schema_index

Authors' Addresses

Joe White
Dell EMC

Email: john.white@dell.com

David Black
Dell EMC

Email: david.black@dell.com

John Leung
Intel Corporation

Email: john.leung@intel.com