SDN Protocols in Internet

Josef Ungerman
Consulting Systems Engineer
Global SP, CCIE #6167
Agenda

• Segmentation of SDN Protocols and API’s
• Controlling Switches and Routers
  – Openflow driven
  – Data Model driven
• IETF protocols SDN evolution
  – BGP
  – PCE
  – IGP and MPLS
• Summary
SDN Protocols – Focus on SP (Internet & Carrier Ethernet)

Application Frameworks, Management Systems, Controllers, ...

Management
Orchestration
Network Services
Control
Forwarding
Device

Protocols
C, Java, Py
NETCONF
REST
OpenFlow
ACI Fabric
OpenStack
DevOps

BGP, IGP, TE, ...

I2RS
IPSE
I2RS

OpFlex
Neutron

API’s (OnePK)
Data Models (Yang)

SSH
HTTP
Plug-in’s

Operating Systems – IOS / NX-OS / IOS-XR

where SDN started
Openflow

Pipeline – this is how L2/L3 ASIC works inside – series of TCAM lookups

- Symmetric Sync Messages (Hello, Echo, Vendor…)
- Async Messages (Port-Status, Flow-Removed, Error…)
- Forwarding Control & Stats Collection

OF 1.1+: New ASIC design required
OF 1.5+: Egress pipeline

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public
OF: describe generalized forwarding primitives
Made for a pipelining L2/L3-switching ASIC (common in DC and LAN)

<table>
<thead>
<tr>
<th>Match Fields</th>
<th>Counters &amp; Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OXM_OF_IN_PORT</td>
<td>Output</td>
</tr>
<tr>
<td>OXM_OF_IN_PHY_PORT</td>
<td>Port</td>
</tr>
<tr>
<td>OXM_OF_METADATA</td>
<td>OFPP_IN_PORT</td>
</tr>
<tr>
<td>OXM_OF_ETH_DST</td>
<td>OFPP_NORMAL</td>
</tr>
<tr>
<td>OXM_OF_ETH_SRC</td>
<td>OFPP_FLOOD</td>
</tr>
<tr>
<td>OXM_OF_ETHETYPE</td>
<td>OFPP_ALL</td>
</tr>
<tr>
<td>OXM_OF_IP_DSCP</td>
<td>OFPP_CONTROLLER</td>
</tr>
<tr>
<td>OXM_OF_IP_PROTO</td>
<td>OFPP_LOCAL</td>
</tr>
<tr>
<td>OXM_OF_IP_PROTO</td>
<td>Set-Queue</td>
</tr>
<tr>
<td>OXM_OF_IPV4_SPC</td>
<td>Drop</td>
</tr>
<tr>
<td>OXM_OF_TCP_SRC</td>
<td>Group</td>
</tr>
<tr>
<td>OXM_OF_TCP_DST</td>
<td>Push-Tag/Pop-Tag</td>
</tr>
<tr>
<td>OXM_OF_UDP_SRC</td>
<td>Pop VLAN header</td>
</tr>
<tr>
<td>OXM_OF_UDP_DST</td>
<td>Push MPLS header</td>
</tr>
<tr>
<td>OXM_OF_SCTP_SRC</td>
<td>Pop MPLS header</td>
</tr>
<tr>
<td>OXM_OF_SCTP_DST</td>
<td>Push PBB header</td>
</tr>
<tr>
<td>OXM_OF_ICMPV4_TYPE</td>
<td>Change-TTL</td>
</tr>
<tr>
<td>OXM_OF_ICMPV4_CODE</td>
<td>Set MPLS TTL</td>
</tr>
<tr>
<td></td>
<td>Decrement MPLS TTL</td>
</tr>
<tr>
<td></td>
<td>Set IP TTL</td>
</tr>
<tr>
<td></td>
<td>Decrement IP TTL</td>
</tr>
<tr>
<td></td>
<td>Copy TTL outwards</td>
</tr>
<tr>
<td></td>
<td>Copy TTL inwards</td>
</tr>
</tbody>
</table>

Very low-level and complex
- start from scratch

What if I want just something simple?
- look for something else

Example: Catalyst 3850, 6800
- UADP ASIC – OF1.3 h/w support
- OF: 17K entries (Flows)
- FIB: 80K entries (IP Routes)
Software can’t control what hardware can’t deliver.
Internet Router

Programmable NPU – not pipeline

- Table Memory
- sTCAM
- DDR3
- Traffic Manager
- Packet Buffers
- Cores Array
- MAC
- Ports PHY
- ASIC
- Fabric Intf

Yes, it can emulate the OF pipeline
- NPU is fully programmable and has TCAM

Cisco ASR9000 – XR 5.1.2
- 8 OF switches
- 4 types: L2, L3, IPv4, dual-stack

Example: ASR 9000
- Typhoon NPU: OF1.3 h/w support
- OF: 30K entries per LC (Flows)
- FIB: 4M entries (IP Routes)
Internet Router – Specific API’s & Protocols

Router has different tables to control

• Higher-level abstraction

Routing Protocols

SDN Controller

If I want something simple…

• add route, delete route
• add VRF, add ACL
• create new LSP/Tunnel…

Platform Specific Control Plane

Drivers, HAL, SDK

Features (QoS, ACL, Netflow…)

Interface Mgmt.
OnePK (Open Networking Environment – Platform Kit)

Open API and SDK for C and Java (ev. Python)

- API’s (functions) grouped in Service Sets (libraries)
- Low-level programming

Thrift
RPC message framework (communication library)

Features
(QoS, ACL, Netflow…)

Policy SS
(ACL, QoS, …)

Element & Interface SS
(interfaces, CPU, …)

Data Path SS
(match, set, punt…)

Routing SS
(RIB read, write, notify)

Topology SS
(discovery)

Platform Specific Control Plane

Drivers, HAL, SDK

ROUTER FORWARDING ENGINE (NPU)

BGP
OSPF
ISIS
LDP
... X
Y
Z

RIB

FIB

Features
(QoS, ACL, Netflow…)

Routing Protocols
SDN Protocols: Agenda

Application Frameworks, Management Systems, Controllers, ...

- **Protocols**
  - C, Java, Py
  - NETCONF
  - REST
  - OpenFlow
  - ACI Fabric
  - OpenStack
  - DevOps

- **Enterprise**
  - Management
  - Orchestration
  - Network Services: BGP, IGP, TE...
  - Control
  - Forwarding
  - Device

- **Service Provider**
  - Puppet/Chef
  - Neutron
  - OpFlex
  - Plug-in’s

- **Operating Systems**
  - IOS / NX-OS / IOS-XR
  - API’s (OnePK)
  - Data Models (Yang)
  - I2RS
  - IPSE
  - SSH
  - HTTP

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public 10
Programmable Interface to Router

**IETF**

- **I2RS (Interface to the Routing System) – RIB & Co.**
  - IETF (I2RS WG): *draft-clemm-i2rs-yang-network-topo*, etc.
  - Asynchronous, high throughput, multi-channel, full-duplex…
  - Data Models: converged to *Yang* (self-describing semantics)

- **IPSE (Interface to Packet Switching Element) – FIB & Co.**
  - IETF: *draft-rfernando-ipse*
  - Interface between CP (Control Plane) and DP (Data Plane)
    - implemented on Virtual PE (VTS), usable with ASIC’s as well
  - Data Models: *Yang* modules for FIB and other fwd’ing tables
    - interface-table
    - l2-table, l2tp-table, flow-table
    - ip-unicast-table, label-table, context-selector-table (VRF)
    - arp-table, arp-proxy-table
    - pse-oam
Network Configuration – Cross-Domain Orchestration

Yang: IETF NETMOD WG
Netconf: IETF NETCONF WG

eg. E-Line Service Attributes
– Endpoints, UNI VLAN’s, CIR, PDU…

eg. E-Line configuration
– EFP, xconnect, class-map, policy-map,…
SDN Protocols: Agenda

IETF Open Standards

Application Frameworks, Management Systems, Controllers, ...

- Protocols
- C, Java, Py
- NETCONF
- REST
- OpenFlow
- ACI Fabric
- OpenStack
- DevOps
- Management
- Orchestration
- Network Services
- BGP, MPLS, ...
- Control
- Forwarding
- Device
- IPSE
- I2RS
- Plug-in’s
- Neutron
- Puppet/Chef
- Operating Systems – IOS / NX-OS / IOS-XR
- API’s (OnePK)
- Data Models (Yang)
- SSH
- HTTP
- C, Java, Py
- RESTful
- OpenFlow
- OpFlex
- NETCONF
- RESTful
- OpenFlow
How open is “Open”? 

• **Open Standard** is not a buzzword
  – collaborative and consensus-driven process, publicly available

• **Open != Open Source**
  – it may contain commercial parts as well as open source

• **Open can be mean an “open invitation to vendor lock-in” 😊**
  – many “open” projects **dominated by a single vendor**
    – eg. Open Contrail, Open vSwitch, proprietary OVSDB extensions…
  – look for a **foundational governance**, diverse community with major industry players
    – eg. OpenStack, OpenDaylight…
MPLS/CarrierE Network Programmability
The Existing Solution and Programmability Enhancements

Motivation: Real-time Network Optimization
• utilize unused links, find non-SPF resources

PCE (Path Computation Element)
– SDN extensions: LSP control delegation and state synchronization
– Stateful PCEP (PCE Protocol)

BGP-LS (Link State)
– SDN Controller gets an accurate topology information & instant updates
– New BGP AF: redistribute ISIS/OSPF topology and SR labels

BGP-FS (FlowSpec)
– SDN-driven innovations: iBGP, Set Next-Hop/TE tunnel, IPv6 support…
– Granular (ACL based) traffic selection and action set

Segment Routing
– SDN-driven evolution of MPLS – ISIS/OSPF enhancements
– Simplification (no RSVP-TE, no LDP), State is moved to a controller
How to get the topology? (and SR labels)

**BGP-LS**

- BGP Link-State (BGP-LS) – *draft-ietf-idr-ls-distribution*
- Redistribute IGP LSDB into per-domain BGP speaker
- Advantages:
  - Single upstream topology feed (BGP)
  - IGP isolated from external entities
  - Leverage well-known BGP security, transport and policy knobs
  - Enables operator control, responsibility border

- IGP LSDB can contain more information than just cost
  - Delay, Jitter, Loss, Residual bw, Available bw
  - Extensions to ISIS/OSPF (new TLVs)
    - *draft-previdi-isis-te-metric-extensions*
    - *draft-ietf-ospf-te-metric-extensions*
BGP-LS Example

IOS-XR 5.1.1

- BGP speakers express their BGP-LS support in capabilities
- LSDB carried in MP_REACH_NLRI
- Link-State Attribute
  - LS NLRI: link, node or prefix (IPv4/IPv6)
  - LS Attribute: Describes a topology element

Prefix string format (sh bgp link link)

[NLRI-Type][Area][Protocol-ID][Local node descriptor][Remote node descriptor][Attributes]/prefix-length
How to program the explicit path?

Stateful PCEP

- **PCE (Path Computation Element) Architecture**
  - Centralized Computation Model for MPLS TE (2006)
  - Cisco innovation, originally for Inter-AS TE (stateless)

- **PCE Architecture Elements**
  - Controller = PCE Server (PCS)
  - Agent = Path Computation Client (PCC) on the router
  - PCE Protocol (PCEP) = protocol between PCC & PCS (RFC5440)
  - Traffic Engineering Database (TED) – contains topology and resource information (LSDB etc.)

- **Stateful PCE (RFC4655)** – introduces PCEP extensions for:
  - LSP state synchronization between PCCs and PCEs
  - PCC delegation of LSP control to PCE (active PCE)
PCE (Path Computation Element) – SDN style

Config Example – IOS-XR 5.1.2

```
! ipv4 unnumbered mpls traffic-eng Loopback0
!
mpls traffic-eng
pce
peer ipv4 172.16.255.3
   stateful-client
       instantiation
!
auto-tunnel pcc
   tunnel-id min 1000 max 5000
!
```

Verification & Troubleshooting
- show mpls traffic-eng auto-tunnel pcc [...] 
- show mpls traffic-eng pce peer [ipv4, node-id,...]
- show mpls traffic-eng pce tunnels <id>
- show mpls traffic-eng pce trace [...] 

Other options – delegation of LSP to Controller (future)

```
interface tunnel-te 101
   pce delegation

interface tunnel-te 102
   path-option 10 dynamic segment-routing PCE addr ipv4 1.1.1.1
```
How to steer traffic to the new path?

Multiple Options

• Per-destination granularity (XR 5.1.1): Use standard routing tools
  - Autoroute Announce or FA (granularity per egress router)
  - Static route or Autoroute Destination (granularity per prefix)
  - PBTS (set forward-class) can add CoS awareness
  - l2vpn preferred path (per pseudowire)
  - ABF (Access-list Based Forwarding)
  - Enhanced Policy Based Routing – PBF, PBTS, Flow-tag (5.2.2)

• Per-app granularity: Controller can use BGP Flowspec to program classifiers on the Edge
  - SET Next-Hop (can resolve over Tunnel, like PBF)
  - MATCH Application (L3/L4 header fields)
BGP Flowspec (RFC5575)
Dissemination of Flow Specification Rules

New NLRI (AFI=1, SAFI=133) – components are optional, order dependent

<table>
<thead>
<tr>
<th>NRLI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Destination IP address</td>
</tr>
<tr>
<td>2</td>
<td>Source IP address</td>
</tr>
<tr>
<td>3</td>
<td>IP protocol</td>
</tr>
<tr>
<td>4</td>
<td>Port</td>
</tr>
<tr>
<td>5</td>
<td>Destination port</td>
</tr>
<tr>
<td>6</td>
<td>Source port</td>
</tr>
<tr>
<td>7</td>
<td>ICMP type</td>
</tr>
<tr>
<td>8</td>
<td>ICMP code</td>
</tr>
<tr>
<td>9</td>
<td>TCP flags</td>
</tr>
<tr>
<td>10</td>
<td>Packet length</td>
</tr>
<tr>
<td>11</td>
<td>DSCP</td>
</tr>
<tr>
<td>12</td>
<td>Fragment</td>
</tr>
</tbody>
</table>

Match = MP_REACH_NLRI (RFC4760)

Action = Extended Community (RFC4360)
BGP Flowspec improvements towards SDN

Original BGP-FS was for DDoS protection
• only policing or set VRF actions

SDN-driven enhancements were needed:
• Redirect IP extension: draft-simpson-idr-flowspec-redirect
• IBGP extension: draft-ietf-idr-bgp-flowspec-oid
• Persistence Support: draft-uttaro-idr-bgp-persistence
• IPv6 extensions: draft-ietf-idr-flow-spec-v6
• HA/NSR Support

Redirect IP details:
• The redirect nexthop can be explicitly configured and can possibly resolve over IP/MPLS/tunnel
• Redirect nexthop is encoded as the MP_REACH nexthop in the BGP flowspec NLRI along with associated extended community.
• The MP_REACH nexthop can be preserved through the use of the “unchanged” knob.
BGP Flowspec Configuration Example
IOS-XR 5.2.0

- Unique capabilities
  - ipv4/vpnv4 and ipv6/vpnv6
  - Redirect IP

**Internet Flowspec**

```
router bgp 100
  address-family ipv4 flowspec !
  address-family ipv6 flowspec !
neighbor 1.1.1.1
  remote-as 100
  address-family ipv4 flowspec !
  address-family ipv6 flowspec !
  vrf foo
    address-family ipv4 flowspec !
    address-family ipv6 flowspec !
  !
  vrf foo
    address-family ipv4 flowspec !
    address-family ipv6 flowspec !
neighbor 10.10.10.10
  remote-as 100
  address-family ipv4 flowspec !
  address-family ipv6 flowspec !
```

**VPN Flowspec**

```
router bgp 100
  address-family vpnv4 flowspec !
  address-family vpnv6 flowspec !
neighbor 1.1.1.1
  remote-as 100
  address-family vpnv4 flowspec !
  address-family vpnv6 flowspec !
  vrf foo
    address-family ipv4 flowspec !
    address-family ipv6 flowspec !
  !
  vrf foo
    address-family ipv4 flowspec !
    address-family ipv6 flowspec !
neighbor 10.10.10.10
  remote-as 100
  address-family vpnv4 flowspec !
  address-family vpnv6 flowspec !
```

**Local Policies**

```
flowspec
  [local-install interface-all]
  address-family ipv4|ipv6
    service-policy type pbr <policy-name>
      [{local | remote}]
    service-policy type pbr <policy-name>
      [{local | remote}]
  !
  vrf <vrf-name>
    address-family ipv4|ipv6
      service-policy type pbr <policy-name>
        [{local | remote}]
      service-policy type pbr <policy-name>
        [{local | remote}]
    !
  interface <interface-name>
    [ipv4|ipv6] flowspec disable
```

- show flowspec summary [vrf {<vrf> | all}] {ipv4 | ipv6 | afi-all} summary
- show flowspec [vrf {<vrf> | all}] {ipv4 | ipv6 | afi-all}
- show flowspec [vrf {<vrf> | all}] {ipv4 | ipv6 | afi-all} detail
Simple extension to IS-IS or OSPF, automatically builds and maintains Segments

Prefix Segment – A Shortest path to the related prefix
Adjacency Segment – One hop through the related adjacency

In MPLS environment: 
Segment == Label
Normal forwarding behavior (push, pop, swap), ECMP, PHP, etc.
No need for LDP, RSVP, etc.
Combining Segments to Engineer Path
“MPLSDN” – MPLS evolution towards SDN

- Software Controls, Network Delivers
  The Controller decides which path is the best
  Shortest Path and ECMP are the default
- The state is in the packet, not in the network
  Great scale – no midpoints, no tailends
  Headend can be a virtual router
- Technology agnostic
  Carries MPLS (service labels), IPv4, IPv6
  Transport over MPLS (stack) or IPv6 (SRH)
  Easy migration – no need for LDPv6, RSVPv6…
Segment Routing – IGP Enhancements

IOS XR 5.2.2

- What we need to advertise
  - SR Capabilities
  - SID (label) – Prefix SID, Adjacency SID
  - SRGB (Segment Routing Global-label Block) – label indexing

- ISIS: draft-ietf-isis-segment-routing-extensions
  - IS-IS Router Capability TLV (242) – SR Capability sub-TLV (2) [SRGB]
  - Extended IP reachability TLV (135) – Prefix-SID sub-TLV (3)
  - SID/Label Binding TLV (149) – Prefix-SID sub-TLV (3)
  - Extended IS Reachability TLV (22) – Adjacency-SID sub-TLV (31), LAN-Adjacency-SID sub-TLV (32)

- OSPF: draft-ietf-ospf-prefix-link-attrib-01, draft-ietf-ospf-segment-routing-extensions-02
  - Router Information Opaque LSA (type 4) – SR-Algorithm TLV (8), SID/Label Range TLV (9) [SRGB]
  - OSPFv2 Extended Prefix Opaque LSA (type 7) – OSPFv2 Extended Prefix TLV (1), Prefix SID Sub-TLV (2)
Segment Routing – CLI examples (XR 5.2.2)

```
router isis 1
  is-type level-2-only
  net 49.0000.1720.1625.5001.00
  address-family ipv4 unicast
  metric-style wide
    segment-routing mpls

  interface Loopback0
  passive
  address-family ipv4 unicast
    prefix-sid {absolute|index} <value> [explicit-null]

```

- Enable SR on all IPv4 interfaces in this IS-IS instance.
- Ipv4 Prefix-SID value for loopback0.

```
router ospf 1
  router-id 100.0.0.1
  segment-routing mpls
  segment-routing forwarding mpls
  area 0
    interface Loopback0
    passive enable
    prefix-sid index <value> [explicit-null]
```

- Enable SR on all areas (can be disabled per area).
- Enable SR forwarding on all interfaces.
- Prefix-SID index for loopback0.
SR EPE – Egress Peer Engineering (future)

AS1
BGP-LU
RFC3107

AS2
BGP-LS

AS3

AS4
9.9.9.9/32

AS3

Controller

B

A

B

C

D

E

16003
24003

Payload

Payload

Payload

router bgp 1

! neighbor RTR_E
address-family ipv4 unicast

sr-epe

! neighbor Controller
address-family link-state link-state

Across AS’s: BGP Prefix-SID draft-keyupdate-idr-bgp-prefix-sid
Other benefits of Segment Routing
Scale, Simplicity, Programmability

- **TI-LFA: automatic <50ms convergence**
  Topology Independent Loop-Free Alternate
  100%-coverage <50ms link and node protection
  Automatic, Simple to operate and understand
  Uses the post-convergence path – no transient congestion and suboptimal routing
  Incremental deployment, protects all traffic

- **Anycast: automatic node redundancy**
  Multiple nodes advertise the same Segment Identifier (anycast segment) in addition to their Prefix-SID
  Traffic is forwarded to the closest node based on IGP best path (automatic rerouting)
  No need for PW Redundancy, VPLS full-mesh, etc.
**Motivation:** Solution for the SDN & NfV Era
- Simple & Programmable, De-Layered, Guaranteed SLA
- All the complexity moves to the Controller and Cloud
- Minimal but “Sufficient” network intelligence

**Software Controls:** YANG
- Network API using standard Network and Service Models, the OSS programs services and paths in real-time

**Network Delivers:** Autonomic Segment Routing
- Devices automatically form an “unbreakable” secure channel (IPv6 RPL) and self-configure for Cisco Validated Design (SDN & Segment Routing)
SDN Protocols: Summary

Application Frameworks, Management Systems, Controllers, ...

- Protocols
  - C, Java, Py
  - NETCONF
  - REST

- OpenFlow
- ACI Fabric
- OpenStack
- DevOps

Management

Orchestration

Network Services

- BGP, IGP, TE, ...

Control

Forwarding

Device

- API’s (OnePK)
- Data Models (Yang)
- SSH
- HTTP

Operating Systems – IOS / NX-OS / IOS-XR

© 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public 32
VIRL (CML) – get your hands dirty!