SESSION 4

Use cases of ONOS+Stratum
Use cases

1. Trellis: Silicon-independent fabric
2. BNG offloading in SEBA
3. S/PGW offloading in M-CORD
Silicon-independent fabric
Trellis – Multi-purpose Leaf-Spine Fabric

- Prominent example of ONOS apps
  - In production at tier-1 operator in the US

- Designed for NFV and access/edge applications
  - Built with white-box switches, open source software, SDN based

- Extensive feature set
  - Bridging/VLANs, IPv4/v6 unicast and multicast routing, DHCP-relay, pseudowires, QinQ, vRouter & more

- Initially designed to work with Broadcom silicon
  - Using the OF-DPA OpenFlow-based agent
Trellis & P4

Pipeline-agnostic apps - use ONOS FlowObjective API

Trellis apps
- Segment Routing
- DHCP L3 Relay
- vRouter
- Multicast
- ...

ONOS

OF-DPA driver

fabric.p4 driver

OpenFlow
Flow table/group mgmt

P4Runtime
Deploy pipeline config
Flow table/group mgmt

Broadcom Qumran
Broadcom Tomahawk
Broadcom Trident2

Barefoot Tofino
Mellanox Spectrum 1

White-box switches

Fabric.p4 pipeconf
Trellis with mixed fabric demo (2018)

Mixed P4/OpenFlow multi-vendor white-box switches
- Broadcom, Barefoot, Edge-Core, Inventec, Delta

End hosts

ONOS

Trellis apps
(Segment routing, multicast, fpm... etc)

P4Runtime

OpenFlow

Upstream BGP routers

Internet

Central office

Field office

Multicast video source

Stratum

OF-DPA

ONOS

Copyright © 2018 - Open Networking Foundation
**fabric.p4**

- **P4 implementation of the Trellis reference pipeline**
  - Inspired by Broadcom OF-DPA pipeline
  - Tailored to Trellis needs (fewer tables, easier to control)

- **Bring more heterogeneity in Trellis with P4-capable silicon**
  - Works with both programmable and fixed-function chips
  - Logical simplified pipeline of L2/L3/MPLS features
  - Any switch pipeline that can be mapped to fabric.p4 can be used with Trellis

- **Extensible open-source implementation**
Design rationale: simplify control plane development

ONOS FlowObjective API (3-stage logical pipeline)

Filtering
- VLAN-based port filtering
- Permit/deny

Forwarding
- FIB for different behaviors (bridging, routing, MPLS SR).
  Set Next ID.

Next ID
- Next ID implementation: rewrite headers, push VLAN/MPLS, replication (broadcast/multicast) and ECMP.

Next
- To ports

V1Model P4 architecture

Ingress pipeline
- filtering.p4
- forwarding.p4
- next.p4
- Traffic manager (replication engine, buffering)

Egress pipeline
- next.p4
P4 provides “easier” silicon independence

- Mapping FlowObjective to new HW is hard!
  - Underspecified/ambiguous pipeline abstraction
- Any switch ASIC that can be mapped to fabric.p4 can be used with Trellis
  - Both programmable and fixed function
- With P4, the mapping effort left to compilers, not ONOS drivers
  - E.g. using Stratum p4c-fpm backend for Broadcom ASICs
OF-DPA vs fabric.p4 mapping driver in ONOS

fabric.p4

$ cd onos/pipelines/fabric/.../pipeliner
$ wc -l *.java
  106 AbstractObjectiveTranslator.java
  284 FabricPipeliner.java
  58 FabricPipelinerException.java
  237 FilteringObjectiveTranslator.java
  252 ForwardingFunctionType.java
  43 ForwardingFunctionTypeCommons.java
  284 ForwardingObjectiveTranslator.java
  498 NextObjectiveTranslator.java
  209 ObjectiveTranslation.java
  20 package-info.java
1991 total

OF-DPA

$ cd onos/drivers/.../pipeline/ofdpa/
$ wc -l Ofdpa*.java
  1985 Ofdpa2GroupHandler.java
  1933 Ofdpa2Pipeline.java
  514 Ofdpa3GroupHandler.java
  913 Ofdpa3Pipeline.java
  49 Ofdpa3QmxPipeline.java
  772 OfdpaGroupHandlerUtility.java
6166 total

x3 more LOCs
BNG offloading in SEBA
Residential Access Recap

Many functions:
- Subscriber tunnel termination (QinQ, PPPoe)
- Accounting
- Hierarchical QoS
- Lawful intercept (wiretap)
- Wholesale tunnel relay (e.g. L2TP)
- Multicast
- Routing

RG – Residential Gateway
OLT – Optical Line Termination
BNG – Broadband Network Gateway

$2B+ market
SEBA 2.0 Today
On path to production

How to disaggregate and embed in SEBA POD?
Plan for BNG embedding in SEBA

• **BNG user plane (BNG-u)**
  - Implement “in-fabric” using P4 and merchant silicon
  - Functional distribution over different chipsets
    • Barefoot Tofino, Broadcom Qumran

• **BNG control plane (BNG-c)**
  - App running on top of ONOS
  - Integrate with existing control planes when possible
    • e.g. external PPPoE server, BGP speaker
Disaggregated BNG

Control plane (BNG-c)

UPSTREAM

- Classification
- Policing
- Double-VLAN termination
- Anti-spoofing
- PPPoE term.
- Routing
- ACL
- Accounting
- Lawful intercept

DOWNSTREAM

- Lawful intercept
- Accounting
- H-QoS
- PPPoE aggr.
- Double-VLAN aggregation
- Mcast replication
- ACL
- Routing
- Classification

User plane (BNG-u)

Barefoot Tofino (P4)

Broadcom Qumran

Credits: Deutsche Telekom
Access 4.0 project
Initial P4 implementation

Demonstrated today
Work in progress
SEBA 3.0 (WIP)
With embedded BNG

**What’s new?**

- P4-programmable Tofino-based switch (instead of Tomahawk)
- Stratum with P4 program as the switch stack (instead of OF-DPA)
- BNG-C app on top of ONOS
ONF Connect 2019 - Demo setup

Subscriber with PPPoE client

OLT

Switch (P4)

ONU

Kubernetes cluster (compute node)

NEM

SEBA Apps

Trellis

BNG-c Relay

VOLTHA

Stratum

External PPPoE Server

Handles authentication, address assignment (IPCP), keep-alive, etc.

Free, unmodified implementation from rp-pppoe project, or bring your own

Internet

Handles authentication, address assignment (IPCP), keep-alive, etc.
ONF Connect 2019 - Demo setup

1 - PPPoE session establishment

BNG-c Relay learns subscriber info by looking into PPPoE control packets relayed via packet-in/out

PPP CONTROL PKTS

ONF
ONF Connect 2019 - Demo setup

2 - User plane termination setup
ONOS apps write P4Runtime entries to terminate and route PPPoE data packets to/from the subscriber

Subscriber with PPPoE client
ONU
OLT
Switch (P4)
Stratum
VOLTHA
SEBA Apps
Trellis
BNG-c Relay
NEM
External PPPoE Server
Internet

Routing flows
PPPoE termination flows
ONF Connect 2019 - Demo setup

3 - Subscriber is connected
P4 switch performs termination and routing of PPPoE data packets
S/PGW offloading in M-CORD
NFV-based mobile access

- **CORD**: ONF NFV platform for the Telco central office
- **M-CORD**: CORD Mobile profile (other profiles exist, e.g. for *residential* access)

Traffic processing intensive VNF:
- GTP tunnel terminations
- Filtering and lawful interception
- Downlink QoS
- Downlink buffering during handover
- Charging

Mobile subscriber traffic

*Traffic processing intensive VNF:*
- GTP tunnel terminations
- Filtering and lawful interception
- Downlink QoS
- Downlink buffering during handover
- Charging

Copyright © 2018 - Open Networking Foundation
M-CORD with offloaded SPGW-u VNF

Mobile subscriber traffic

SPGW-u executed directly on the switching ASIC

ONOS

Trellis Apps

SPGW-u App

P4Runtime
P4 program deployment and table management

eNodeB

GTP tunnels

Backhaul network

spgw.p4

TOR

Spine

Spine

MME

HSS

Upstream router

Copyright © 2018 - Open Networking Foundation
PoC P4 implementation of the SPGW-u data plane
- ~300 lines of P4_16 code
- Integrated with fabric.p4
- [https://github.com/opennetworkinglab/onos/…/spgw.p4](https://github.com/opennetworkinglab/onos/…/spgw.p4)

Good enough to demonstrate end-to-end connectivity
- Support GTP encap/decap, filtering, charging functionalities

Important missing features
- Downlink buffering during handovers
- Downlink QoS
SPGW-u App

3GPP Control and User Plane Separation (CUPS) protocol
Create/modify/delete GTP sessions

ONOS

P4Runtime
spgw.p4 table entries

Spine  Spine

ToR  ToR

SPGW-c

3GPP

Open source EPC from Intel/Sprint
Exercise 4: Modify code to enable IPv6 routing
Exercise 3 steps

We want the topology to behave like a traditional IPv6 fabric, but... Implementation is broken, your task is to fix it!

Steps:

● Debug the issue (with step-by-step instructions)
● Update P4 program
● Run PTF unit tests to validate P4 changes
● Update ONOS app implementation
● Test connectivity between hosts on different IPv6 subnets
PTF overview

- Python-based dataplane test framework
- Similar to OFTest framework
  - But focuses on the dataplane and is independent of OpenFlow/P4Runtime
- High-level P4Runtime lib provided with starter code
  - Add/remove table entries, groups, packet-in/out, etc.

PTF-based test

Produce input (packet)

Stratum

Bmv2 simple_switch

Verifies output

P4Runtime
Bmv2.json, P4Info
Table entries
IPv6 fabric recap

- Leaf switches should behave as a traditional router (simplified)
  - i.e., with IPv6 configuration on interfaces (address and subnet)
- Hosts configured with “gateway” IPv6 address the leaf switch one
- Hosts should be able to resolve the MAC address of their gateway
  - i.e. the leaf switch should reply to NDP Neighbor Solicitation messages sent by the hosts
- Not all packets need to be “routed” by leaf switches
  - Only those with destination MAC address the “gateway” one (myStationMac)
- Switch maps IPv6 prefixes (LPM) to next hops (routing table)
- Support ECMP when forwarding to spine switches
netcfg.json (fabricDeviceConfig)

```json
{
    "devices": {
        "device:leaf1": {
            "basic": {
                "managementAddress": "grpc://mininet:50001?device_id=1",
                "driver": "stratum-bmv2",
                "pipeconf": "org.onosproject.ngsdn-tutorial"
            }
        },
        "fabricDeviceConfig": {
            "myStationMac": "00:aa:00:00:00:01",
            "isSpine": false
        }
    }
}
...
```
..."ports": {
  "device:leaf1/3": {
    "interfaces": [
      {
        "name": "leaf1-3",
        "ips": ["2001:1:1::ff/64"]
      }
    ]
  },
...

IPv6 unicast routing

Host 1 2001:1::/64 Router 1 Router 2 2001:3::/64 Host 2

Packet

Eth Src: Host 1
Eth Dst: Router 1
IP Src: Host 1
IP Dst: Host 2
IP unicast routing

Host 1 - 2001:1::/64 → Router 1 - Packet → Router 2 - 2001:3::/64 → Host 2

Eth Src: Router 1
Eth Dst: Router 2
IP Src: Host 1
IP Dst: Host 2
IP unicast routing

Host 1 2001:1::/64 Router 1 Router 2 2001:3::/64 Host 2

Packet

Eth Src: Router 2
Eth Dst: Host 2
IP Src: Host 1
IP Dst: Host 2
Neighbor Discovery Protocol (NDP)

Default gateway is 2001:1::ff

Host 1  2001:1::1a/64  2001:1::ff/64  Router 1
Neighbor Discovery Protocol

Host 1 2001:1::1a/64

NDP

2001:1::ff/64

Router 1

Router/Neighbor Solicitation
Who has MAC of 2001:1::ff?
Neighbor Discovery Protocol

How to generate NDP replies from the switch?
Same-leaf routing

**Diagram:**
- **Spine1**
- **Spine2**
- **Leaf1**
  - 2001:1:1::/64
  - 2001:1:2::/64
  - h1a
  - h1b
  - h1c
  - h2
- **Leaf2**
  - 2001:2:3::/64
  - 2001:2:4::/64
  - h3
  - h4

**Network Ranges:**
- 2001:1:1::/64
- 2001:1:2::/64
- 2001:2:3::/64
- 2001:2:4::/64
Same-leaf routing

- **Leaf1**: 2001:1:1::/64, 2001:1:2::ff
- **Leaf2**: 2001:2:3::/64, 2001:2:4::/64

- **Spine1**
- **Spine2**

- **Hosts**: h1a, h1b, h1c, h2, h3, h4

Simple IP unicast on same leaf
Route to other leaves via ECMP

- **spine1**
  - 2001:1:1::/64
  - 2001:1:2::/64
  - h1a
  - h1b
  - h1c
  - h2

- **spine2**
  - 2001:2:3::/64
  - 2001:2:4::/64
  - h3
  - h4

- **leaf1**
  - 2001:1:1::/64
  - 2001:1:2::/64

- **leaf2**
  - 2001:2:3::/64
  - 2001:2:4::/64
Ipv6RoutingComponent.java

- **Listens to device and topology events**

- **For each device, provision flow rules to:**
  - Match on `myStationMac` to enable routing table
  - Route packets to attached hosts (/128)
  - Route packets to spines when matching on other leaves IPv6 subnets (from interface config)
  - Groups used for both attached hosts (one next hop) and ECMP (multiple next hops, one per link/spine)

- **Looks at topology information (leaf-spine links) and netcfg to compute path, generate flow rules, and groups**
Exercise 4: Get Started

Open lab README on GitHub:

Or open in text editor:
~/ngsdn-tutorial/README.md
~/ngsdn-tutorial/EXERCISE-4.md

Solution:
~/ngsdn-tutorial/solution

Before starting!
Update tutorial repo
(requires Internet access)
cd ~/ngsdn-tutorial
git pull origin master
make pull-deps

P4 language cheat sheet:
http://bit.ly/p4-cs

You can work on your own using the instructions.
Ask for instructors help when needed.
Wrap Up
Recap

● Domain specific languages
  ○ P4 (pipeline modeling), YANG (configuration modeling)
● Models
  ○ Tutorial P4 program (IPv6 router), OpenConfig
● APIs
  ○ P4Runtime, gNMI, gNOI
● Switch OS
  ○ Stratum - implementation of P4Runtime, gNMI, and gNOI
● Controller platforms
  ○ ONOS - with support for Stratum
Learn more @ ONF Connect 2019

● **Talks at Next-Gen SDN Track:**
  ○ Operator’s update on P4 use cases for the Edge Cloud
  ○ P4 compiler for fixed-function switches
  ○ Validation of fixed-function switches against a P4 Program
  ○ Refactoring OpenFlow solutions to P4Runtime
  ○ µONOS project overview
  ○ and more…

● **Demos**
  ○ BNG disaggregation with Stratum and SEBA
  ○ Stratum interoperability: Broadcom Tomahawk and Barefoot Tofino
  ○ µONOS demo with Stratum
Thanks!