Offloading VNFs to programmable switches using P4

Carmelo Cascone, Uyen Chau

ONS North America, March 27, 2018
This talk in a nutshell

CORD controller (XOS)

Fabric Control App

ONOS (cluster)

Fabric Control App... ONOS (cluster)

P4Runtime
P4 program deployment and table management

Mobile subscriber traffic

eNodeB

GTP tunnels

Backhaul network

Spine

Spine

ToR

ToR

SPGW-c

SPGW-u

MME

HSS

Upstream router

Mobile subscriber traffic
This talk in a nutshell

CORD controller (XOS)
- Fabric Control App
- SPGW-u App

ONOS (cluster)

P4Runtime
P4 program deployment and table management

P4 program deployment and table management

Cord controller (XOS)
- Fabric Control App
- SPGW-u App

Mobile subscriber traffic

GTP tunnels

SPGW-c

MME

HSS

Upstream router

Backhaul network

eNodeB

GTP tunnels

spgw.p4

Mobile subscriber traffic
ONF – An Operator Led Consortium

“Nearly 40% of all end-customers will have service provided by … CORD by mid-2021”
Roz Roseboro
Heavy Reading

“70% of operators worldwide are planning to deploy CORD”
Michael Howard
IHS Markit

ONF Mission
Transforming Networks into Agile Platforms for Service Delivery
Leveraging Disaggregation and Open Source to Build Innovative Solutions for Operator Networks and Catalyze our industry to accomplish this transformation
ONF Unique Approach

- Blending Software and Standards
- Blending Research and Vendor Innovations
- Delivering Deployable Solutions (not just piece parts)

- Small independent engineering team
- Dedicated to the mission with no legacy to protect

- SDN -> open source -> ONOS -> CORD -> Stratum
- Creating ‘Killer Solution’ for Network Edge (CORD)
  - Demonstrated Impact - 70% of Operators Planning to Deploy CORD (IHS survey)
ONOS Distributed Architecture

- Apps
  - NB Core API
    - Distributed Core
      - (state management, notifications, high-availability & scale-out)
  - SB Core API
    - Providers
      - Protocols
    - Providers
      - Protocols
    - Providers
      - Protocols
    - Providers
      - Protocols
CORD High Level Architecture

- Large number of COs
- Evolved over 40-50 years
- 300+ Types of equipment
- Huge source of CAPEX/OPEX

CORD-XOS Controller

- Mobile
- Enterprise
- Residential

- SDN
- NFV

- Cloud

- Metro Ethernet
- BBU
- PON OLTs
- Shared Cloud Infrastructure
- ROADM (Core)
CORD Architecture

- **ONOS**
- **OpenStack / K8S**
- **VNFs on Compute Nodes (vSG)**
- **Commodity Servers**
- **White-Box Switches**
- **Merchant Silicon**
- **Control applications on ONOS (Trellis)**

- **Access**
- **Hardware**
- **OCP**
- **VNF**
- **Ctrl App**
Trellis – Multi-purpose Leaf-Spine Fabric

ONOS Cluster

Access & Trunk VLANs
IPv4 & IPv6 & MPLS SR
IPv4 & IPv6 Multicast
DHCP L3 relay (IPv4/v6)
vRouter BGPv4/v6(external)
Dual-homing
PWs

VxLAN overlay
QinQ termination

L2 bridged
L3 routed
IP multicast
Trellis & P4

ONOS extended with P4/P4Runtime support: control any pipeline, with any app

- Target-specific P4 artifacts
- Driver (Java code) to allow ONOS “understand” the pipeline

Same set of Trellis applications on ONOS

ONOS

Segment Routing  DHCP L3 Relay  vRouter  Multicast

OF-DPA driver  fabric.p4 driver

OpenFlow  OFConfig  P4Runtime  gNMI

Brcm Qumran  Brcm Tomahawk  Brcm Trident2

Barefoot Tofino  Cavium Xpliant  Mellanox

Brcm Qumran

Brcm Tomahawk

Brcm Trident2

OpenFlow  OFConfig  P4Runtime  gNMI
Offloading the SPGW-u VNF to the P4 fabric
M-CORD with P4 fastpath

- CORD controller (XOS)
  - Trellis Control App
  - SPGW-u App
- ONOS (cluster)
- SDN app

- Mobile subscriber traffic
  - eNodeB
  - GTP tunnels
  - Backhaul network
  - Upstream router
  - eNodeB
  - GTP tunnels
  - Backhaul network

- SDN app
  - fabric.p4
  - spgw.p4

- Mobile subscriber traffic
  - eNodeB
  - GTP tunnels
  - Backhaul network
  - Upstream router
  - eNodeB
  - GTP tunnels
  - Backhaul network
P4 recap

- Domain-specific language to formally define the logical pipeline behavior
  - Describe headers, lookup tables, actions, etc.
  - Can describe fast pipelines (e.g. ASIC, FPGA) as well as slower ones (e.g. SW switch)
- Good for programmable switches, as well as fixed-function ones
- Defines “contract” between the control plane and data plane

![Diagram showing P4 pipeline and compiler process](image-url)
P4Runtime recap

P4 program

Table { match actions }

Application

P4Runtime Agent

compile

Table management (p4runtime.proto over gRPC)

P4Runtime message payloads derived from P4 program

Packets

Programmable or fixed-function pipeline
ONS demo: P4Runtime-enabled multi-vendor fabric

ToR

ToR

Spine

Spine

ONOS

Trellis Apps

Mellanox Spectrum 1

Cavium Xpliant

Barefoot Tofino

Barefoot Tofino

P4Runtime
P4 program deployment and
table management

fabric.p4

fabric+spgw.p4

Barefoot

Cavium

Mellanox
fabric.p4

- **P4 implementation of the Trellis reference pipeline**
  - Inspired by Broadcom OF-DPA
  - Tailored to Trellis needs (fewer tables)
  - Work in progress:
    - Tested support for L2 bridging, IPv4 routing, MPLS segment routing

- **Open-source implementation based on P4_16**
  - Hosted in ONOS repository
  - Depends only on open-source libraries (v1model.p4)
  - Can compile and test on Mininet with BMv2 software switch
  - Need few private bits to be able to compile it on HW
fabric.p4 pipeline

- In-port + VLAN filtering table
- Forwarding classifier
  - Bridging
  - MPLS
  - IPv4 unicast routing
  - IPv4 multicast routing
  - IPv6 unicast routing
  - IPv6 multicast routing
- ACL
- Next ID mapping
  - Simple
  - Hashed
  - Multicast
PoC P4 implementation of the SPGW data plane
- ~300 lines of P4 code
- Hosted in the ONOS repo as part of fabric.p4
- Good enough to demonstrate end-to-end connectivity
  - Support GTP encap/decap, filtering, charging functionalities
  - Some missing features (future work):
    - **Downlink buffering during handovers**: async process, cannot describe in P4, need cooperation of CPU and external storage
    - **QoS**: easy to describe rate-limiting in P4 (for downlink), P4 cannot describe scheduling
SPGW-u App

3GPP CUPS
Create/modify/delete GTP sessions

ONOS

P4Runtime
spgw.p4 table entries

Spine

ToR

SPGW-c

3GPP

MME

HSS

Open source EPC from Intel/Sprint
Switching ASIC vs CPU - What are the benefits?

- **Maximized, deterministic throughput**
  - Always process traffic at line rate, with any traffic workload
- **Minimized, deterministic processing latency (and jitter)**
  - In the order of nanoseconds, with any traffic workload
- **Reduced power consumption**
  - Use less CPU resources, instead use switch that is there anyways

**Achieved effortlessly!**
Writing P4 code is easier than writing C code optimized for throughput/latency/power consumption
ONS demo: benefits of spgw.p4

Overhead due to GTP and INT headers

Hop latency measured using in-band network telemetry (INT)

~490ns to perform GTP encap plus forwarding (ToR 1)
~480ns to perform forwarding (ToR 2)
Challenges (1)

How many concurrent subscribers can we handle on the switch?

- **Per-subscriber state in SPGW**
  - GTP tunnel info, counters (billing), bearer mapping rules, etc.

- **Limited ASIC memory, allocated by the P4 compiler**
  - Number of subscribers depends on memory available, compiler optimizations

- **Size-speed trade-off in memories**
  - Fast on-chip memories are usually small, tens of MB for a terabit DC switch
  - Can handle tens of thousands of subscribers, but not millions (like commercial EPCs or CPU-based VNF implementations)

- **Solutions**
  - Use more switches, i.e. distribute subscriber state across the fabric
  - Wait for next-gen P4 chips: less throughput, larger memories (expandable off-chip)
Challenges (2)

How many VNFs can we execute on one switch?

- P4 chips have a fixed number of match-action stages
  - Multiple simultaneous lookups and actions can be supported on each stage
  - Match/action dependencies call for sequential or parallel execution
- Number of VNFs depends on match action dependencies
  - ...compiler optimizations, and memory
- If stage limit is hit, can distribute/split VNFs across the fabric
Takeaways

• **P4 enables open-source target-independent data plane evolution**
  • fabric.p4 and spgw.p4 available on ONOS repository
  • ONF mission to deliver reference P4 implementations

• **Great benefits when offloading VNFs to the switching fabric**
  • Throughput, latency, power consumption

• **Technical challenges that needs to be addressed**
Next steps

- Integration of P4 fabric in CORD
  - Multicast, Broadcast, ACL
- In-band Network Telemetry
- Other VNF offloading
  - BNG (QoS)
  - PPPoE termination
Further reading and contacts

P4 Brigade wiki:
https://wiki.onosproject.org/x/2oS9

P4 Brigade mailing list:
brigade-p4@onosproject.org

ONOS Code
https://github.com/opennetworkinglab/onos

ONOS wiki:
https://wiki.onosproject.org