Open Networking Foundation

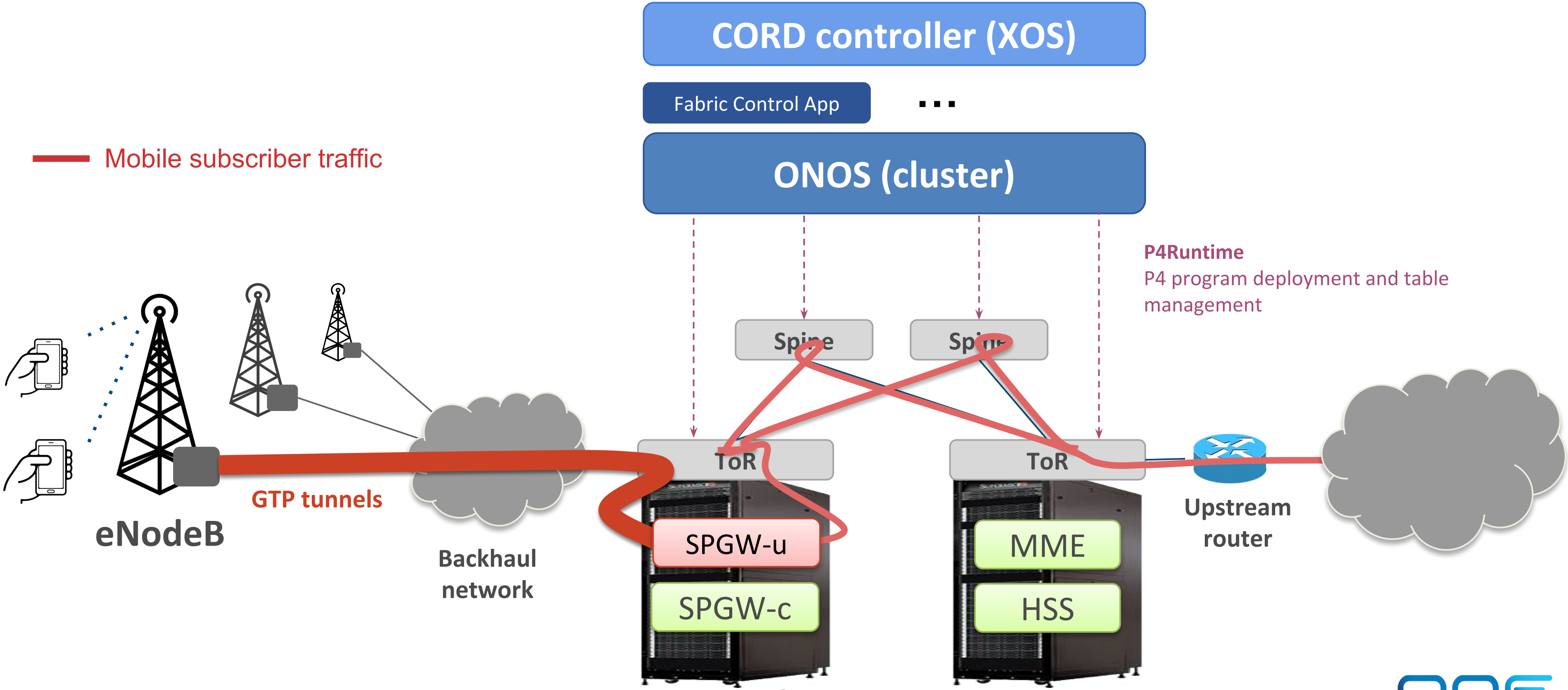
Offloading VNFs to programmable switches using P4

Carmelo Cascone, Uyen Chau

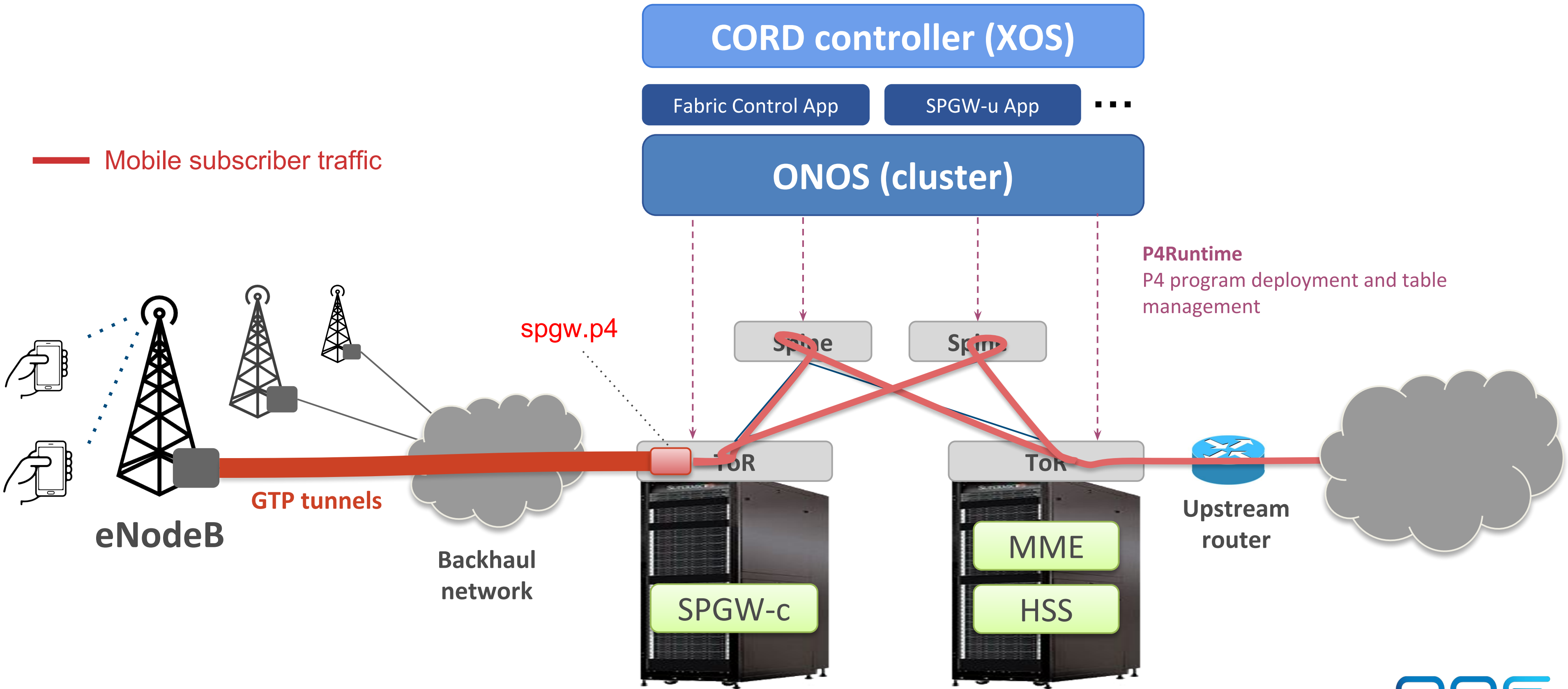
ONS North America, March 27, 2018

An Operator Led Consortium

This talk in a nutshell



This talk in a nutshell



CORD controller (XOS)

Fabric Control App

SPGW-u App

...

ONOS (cluster)

Mobile subscriber traffic

P4Runtime

P4 program deployment and table management

spgw.p4

Spine

Spine

ToR

ToR

GTP tunnels

eNodeB

Backhaul network

SPGW-c

MME

HSS

Upstream router

ONF – An Operator Led Consortium

AT&T

China unicom中国联通

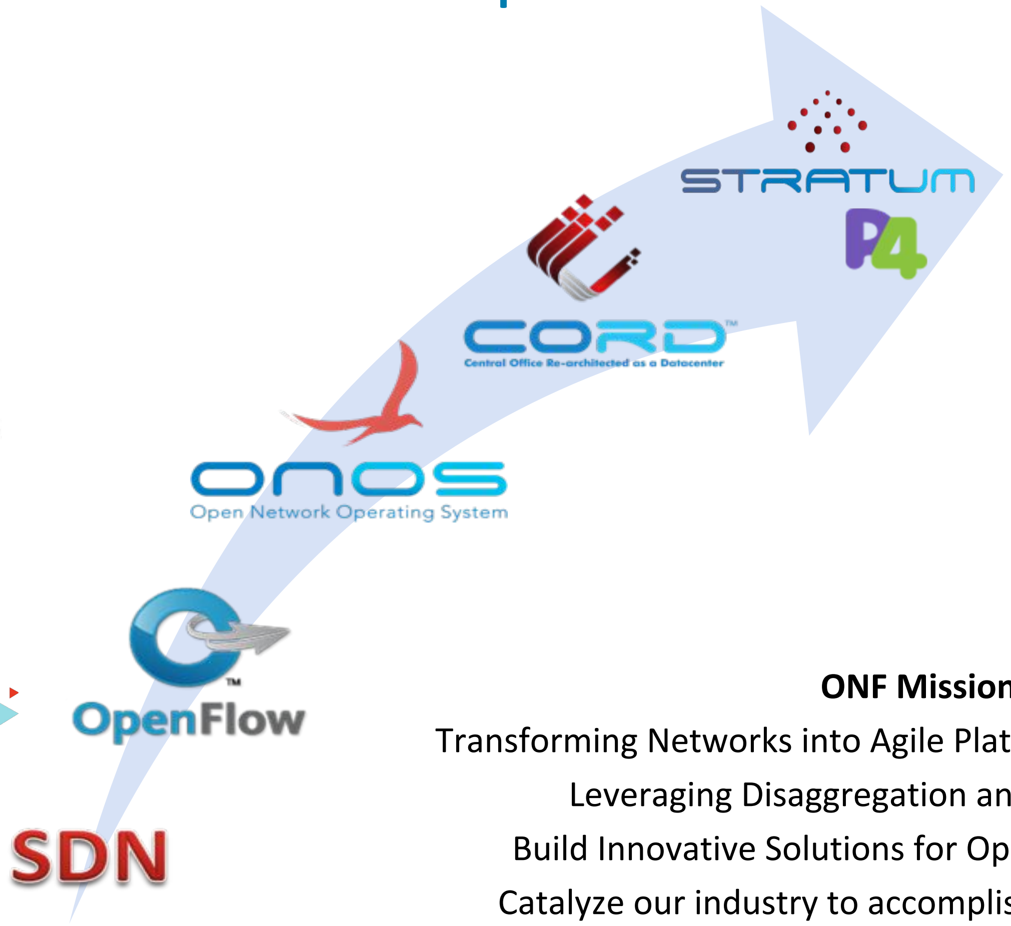
COMCAST

Deutsche Telekom

Google

NTT Group

Türk Telekom


SDN
OpenFlow
ONOS
Open Network Operating System
CORD
Central Office Re-architected as a Datacenter
STRATUM
P4

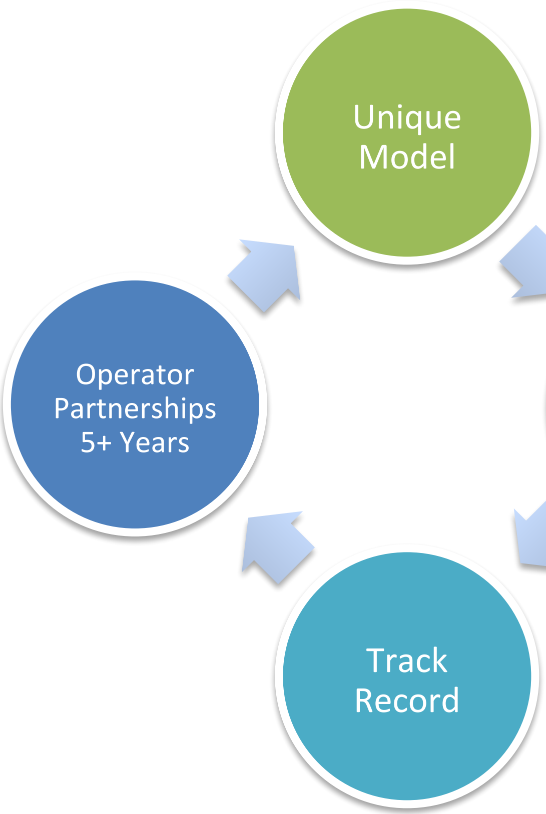
> "Nearly 40% of all end-customers will have service provided by … CORD by mid-2021"
>
> Roz Roseboro
> Heavy Reading

> "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

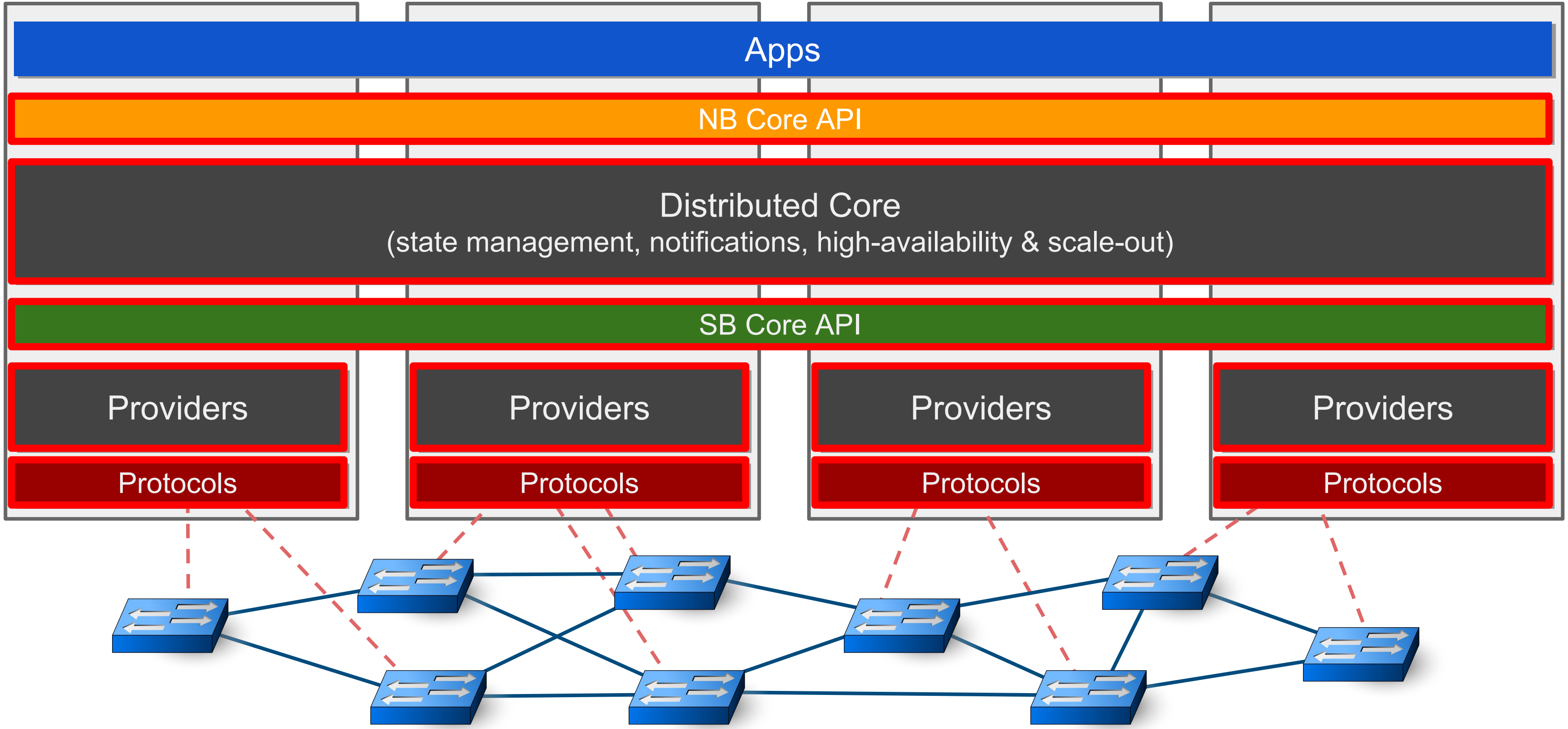


- Blending Software and Standards
- Blending Research and Vendor Innovations
- Delivering Deployable Solutions (not just piece parts)

- 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

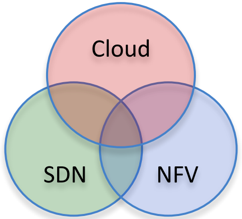
Protocols

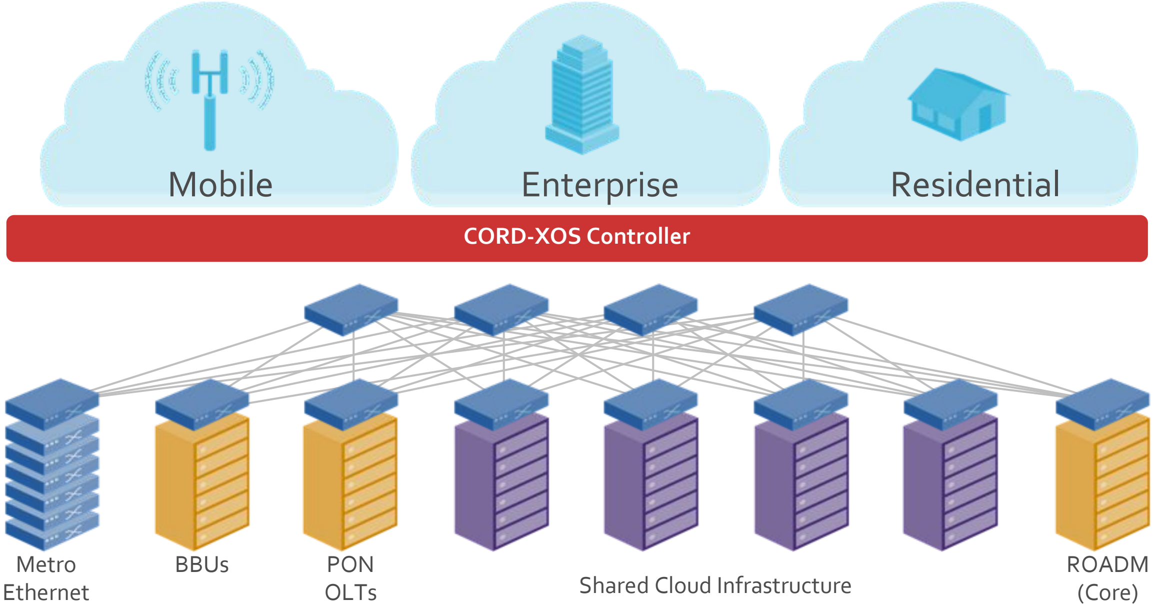
CORD High Level Architecture

Large number of COs

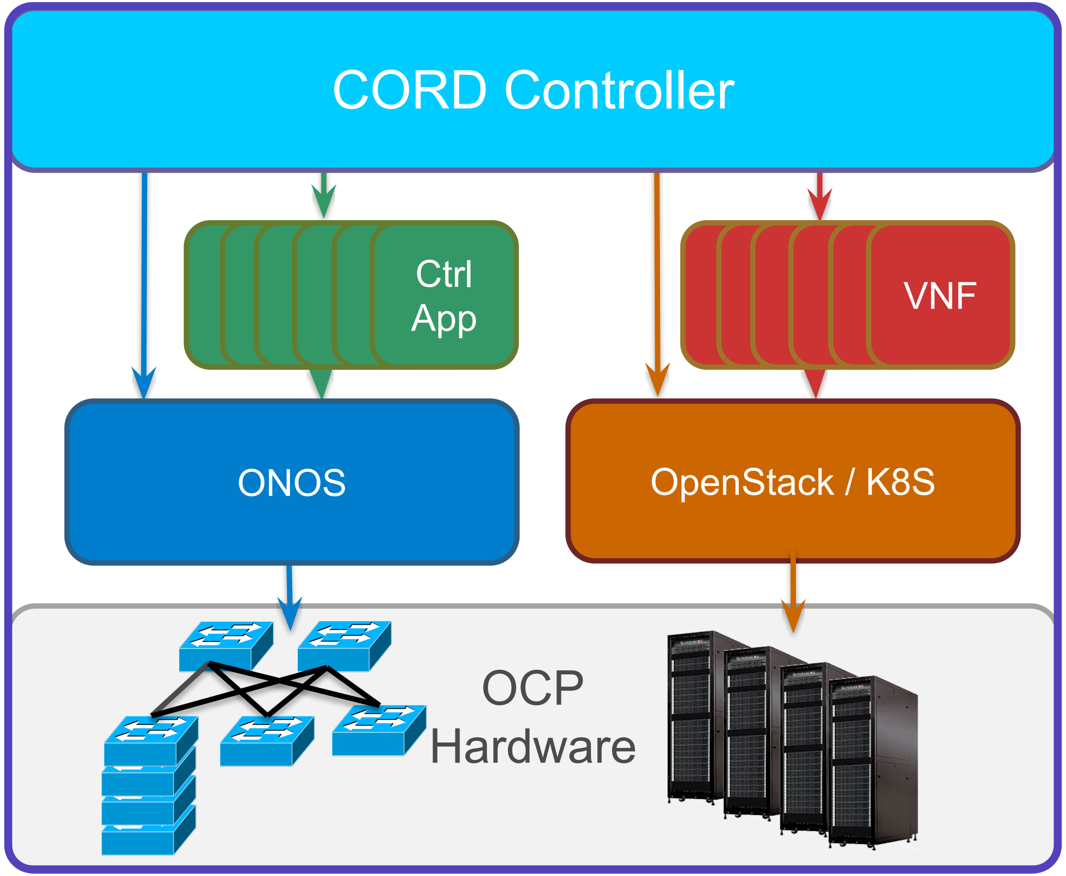
Evolved over 40-50 years

300+ Types of equipment
Huge source of CAPEX/OPEX





CORD Architecture

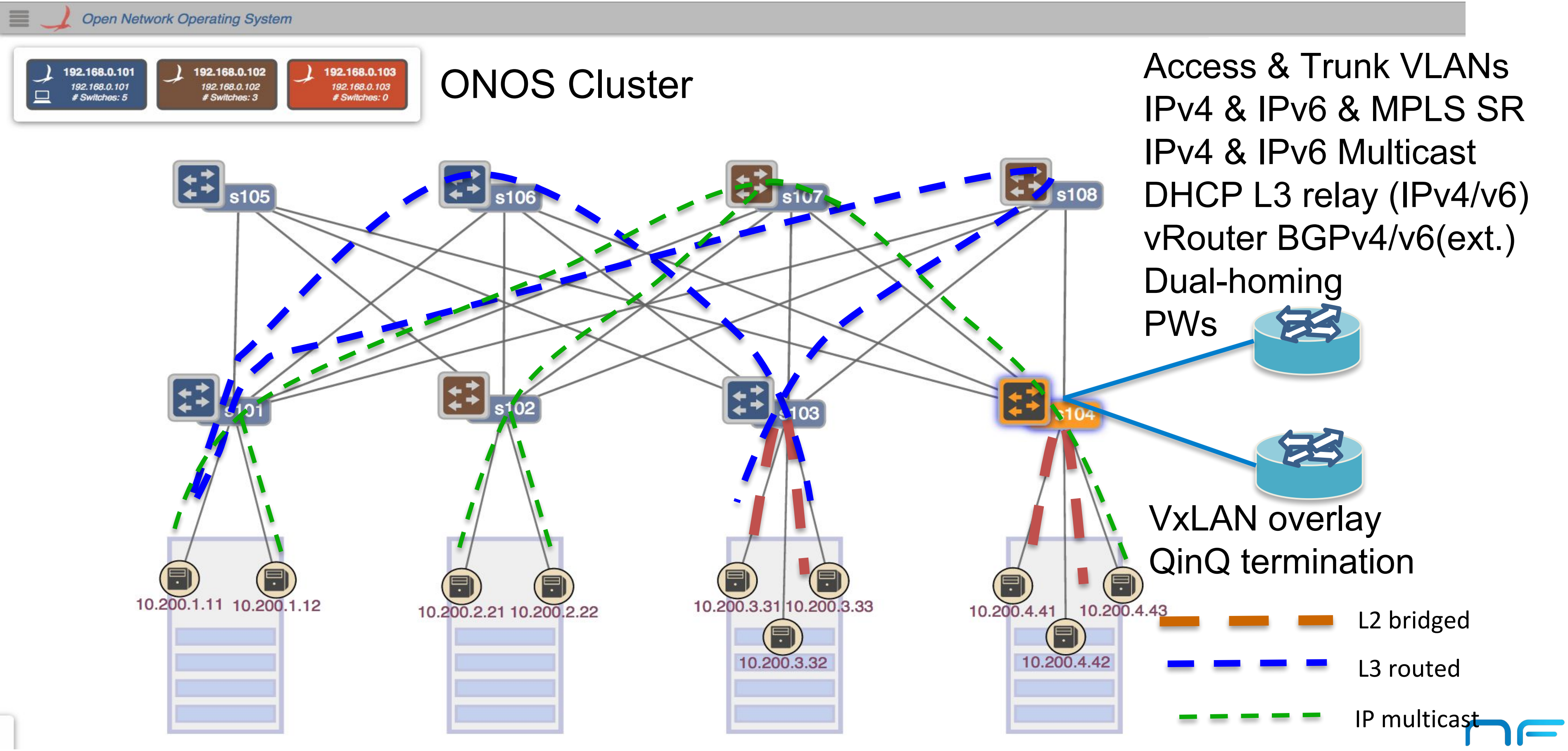


Control applications on ONOS (Trellis)

VNFs on Compute Nodes (vSG)

Commodity Servers
White-Box Switches
Merchant Silicon
Access

Trellis – Multi-purpose Leaf-Spine Fabric



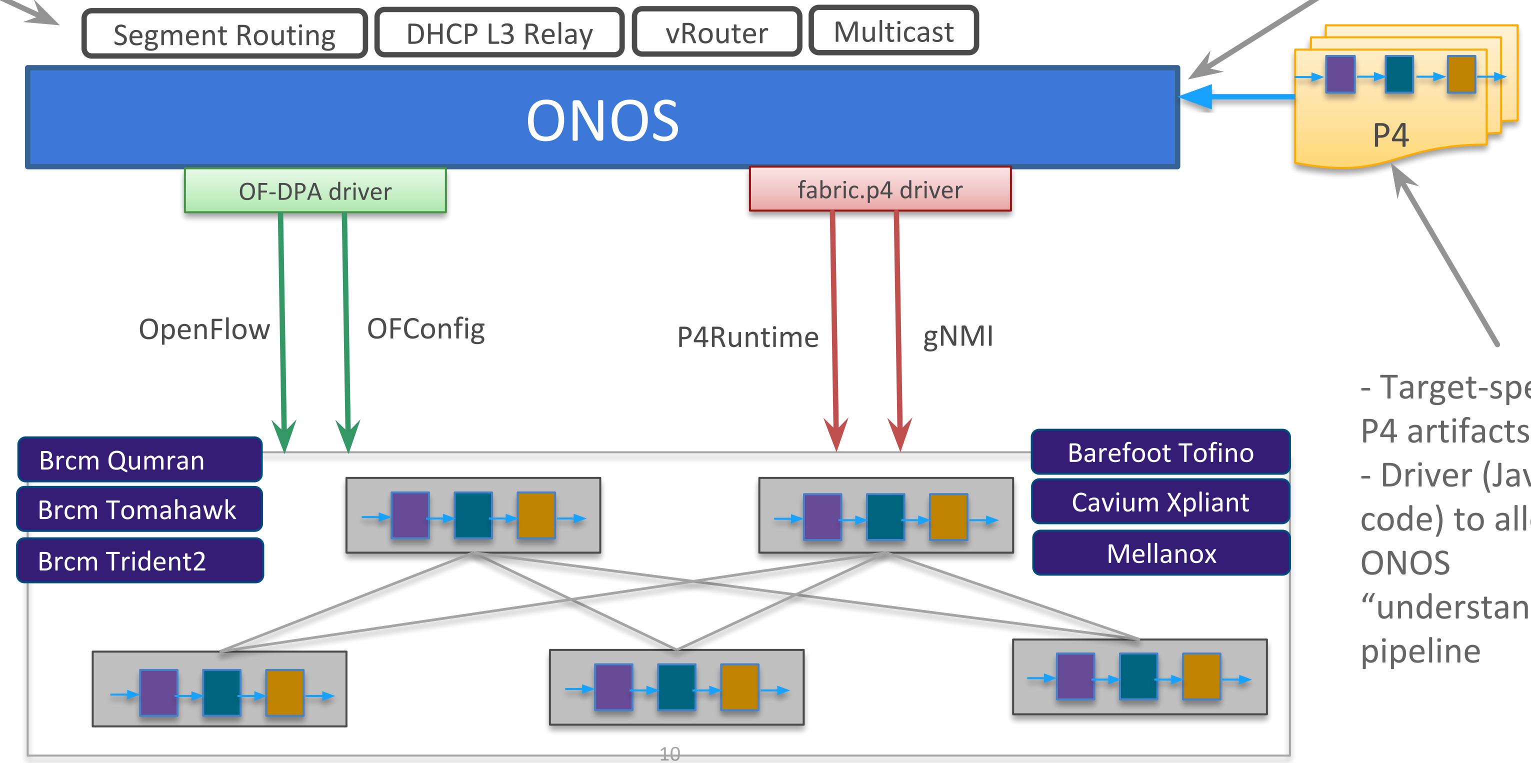
Access & Trunk VLANs
IPv4 & IPv6 & MPLS SR
IPv4 & IPv6 Multicast
DHCP L3 relay (IPv4/v6)
vRouter BGPv4/v6(ext.)
Dual-homing
PWs

VxLAN overlay
QinQ termination

Trellis & P4

Same set of Trellis applications on ONOS

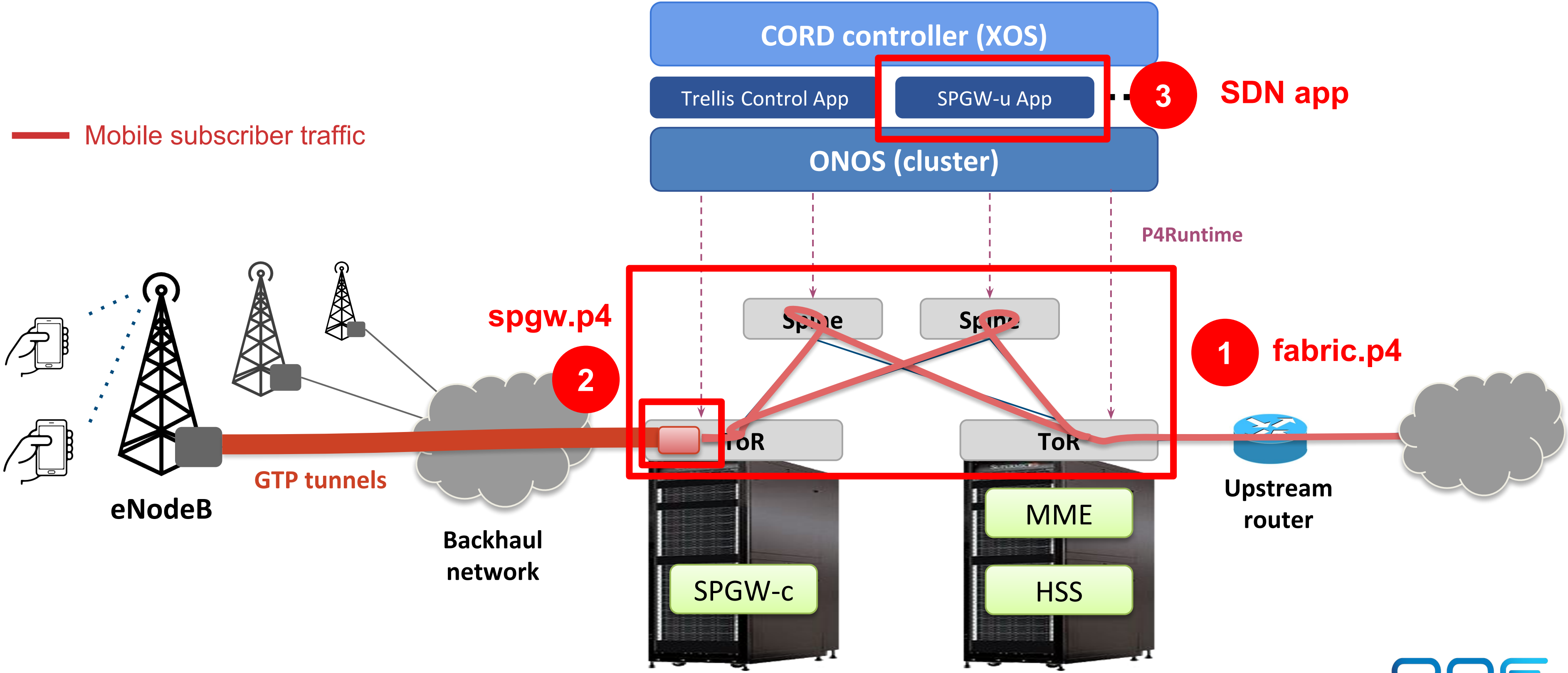
ONOS extended with P4/P4Runtime support: control **any** pipeline, with **any** app



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

Offloading the SPGW-u VNF to the P4 fabric

M-CORD with P4 fastpath



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 a 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**

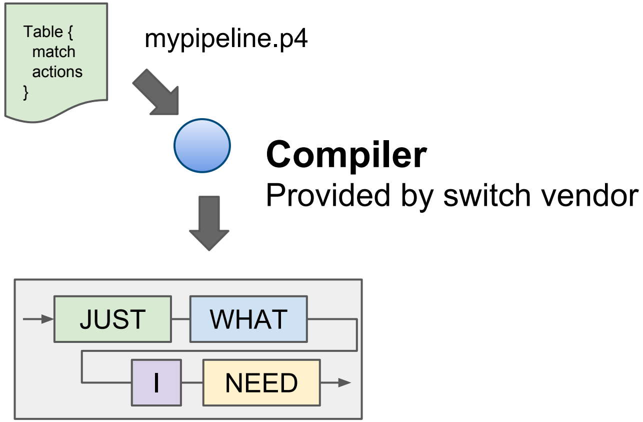


Table {
match
actions
}

mypipeline.p4

Compiler
Provided by switch vendor

JUST WHAT
I NEED

Programmable or fixed-function
data plane pipeline

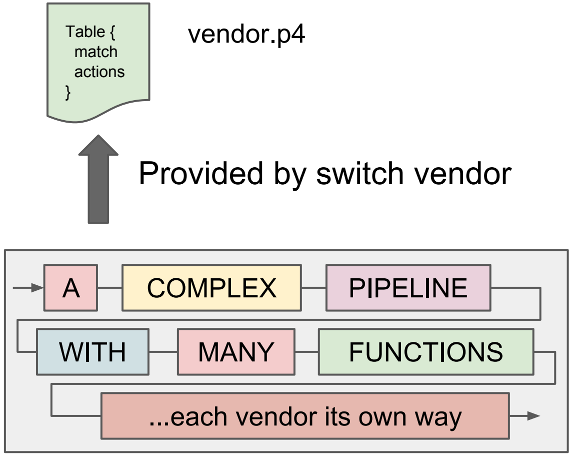


Table {
match
actions
}

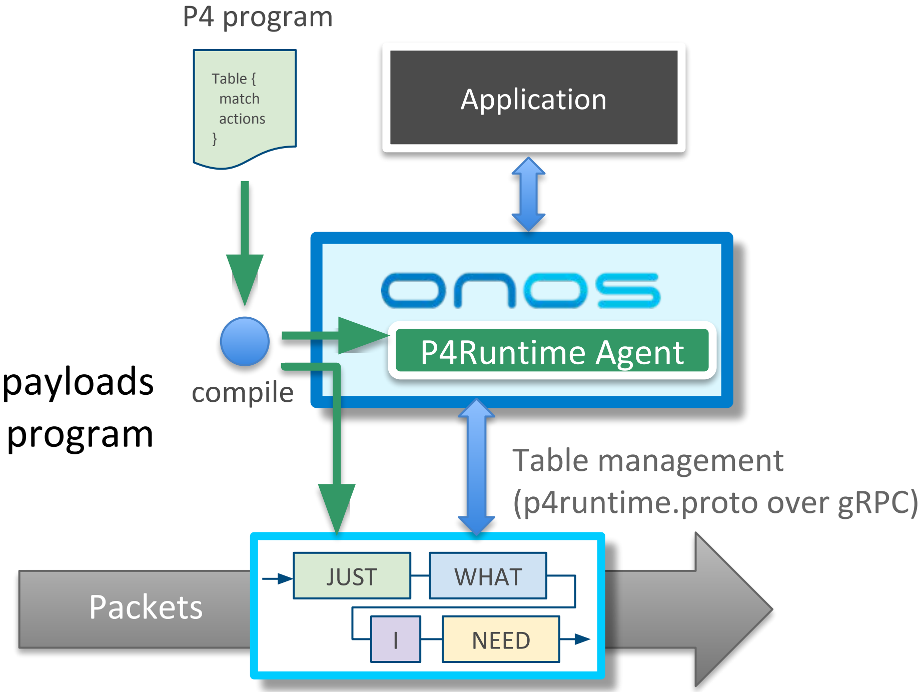
vendor.p4

Provided by switch vendor

A COMPLEX PIPELINE
WITH MANY FUNCTIONS
...each vendor its own way

Fixed-function
data plane pipeline

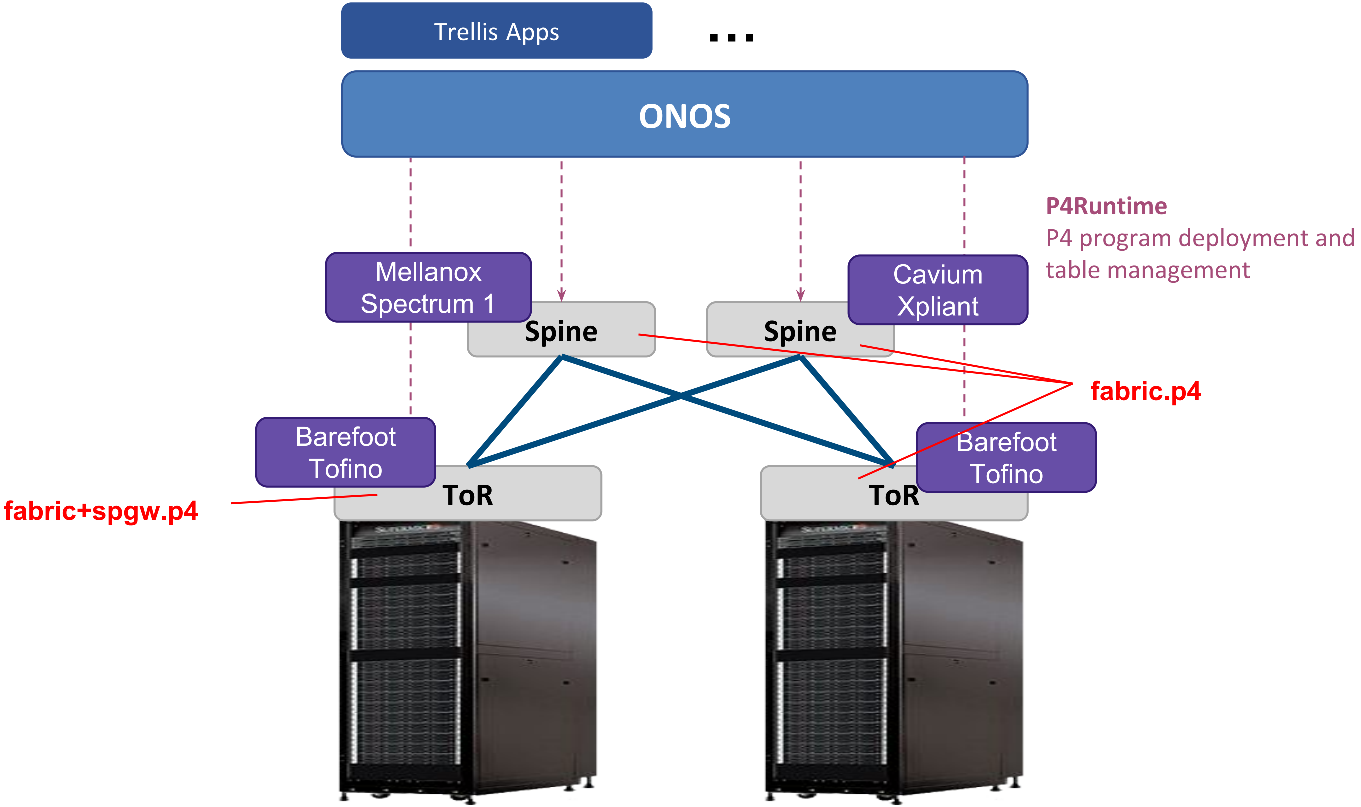
P4Runtime recap



P4Runtime message payloads derived from P4 program

Programmable or fixed-function pipeline

ONS demo: P4Runtime-enabled multi-vendor fabric

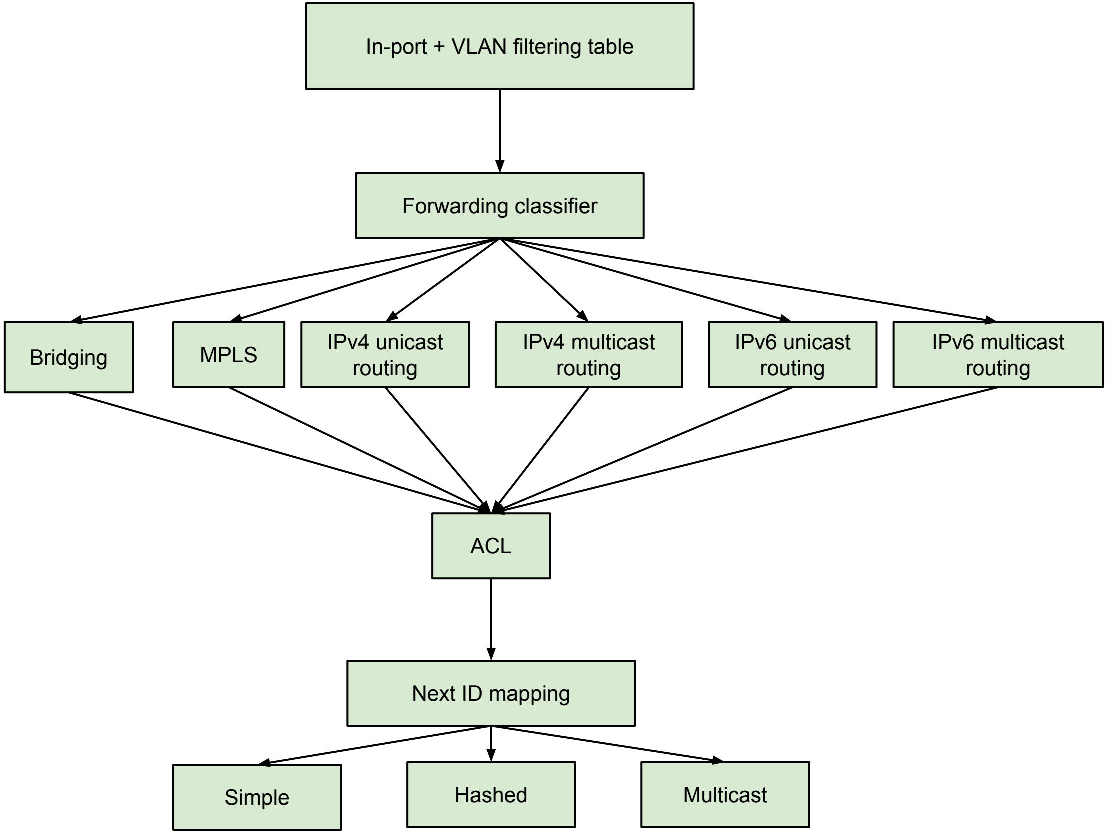


P4Runtime
P4 program deployment and table management

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

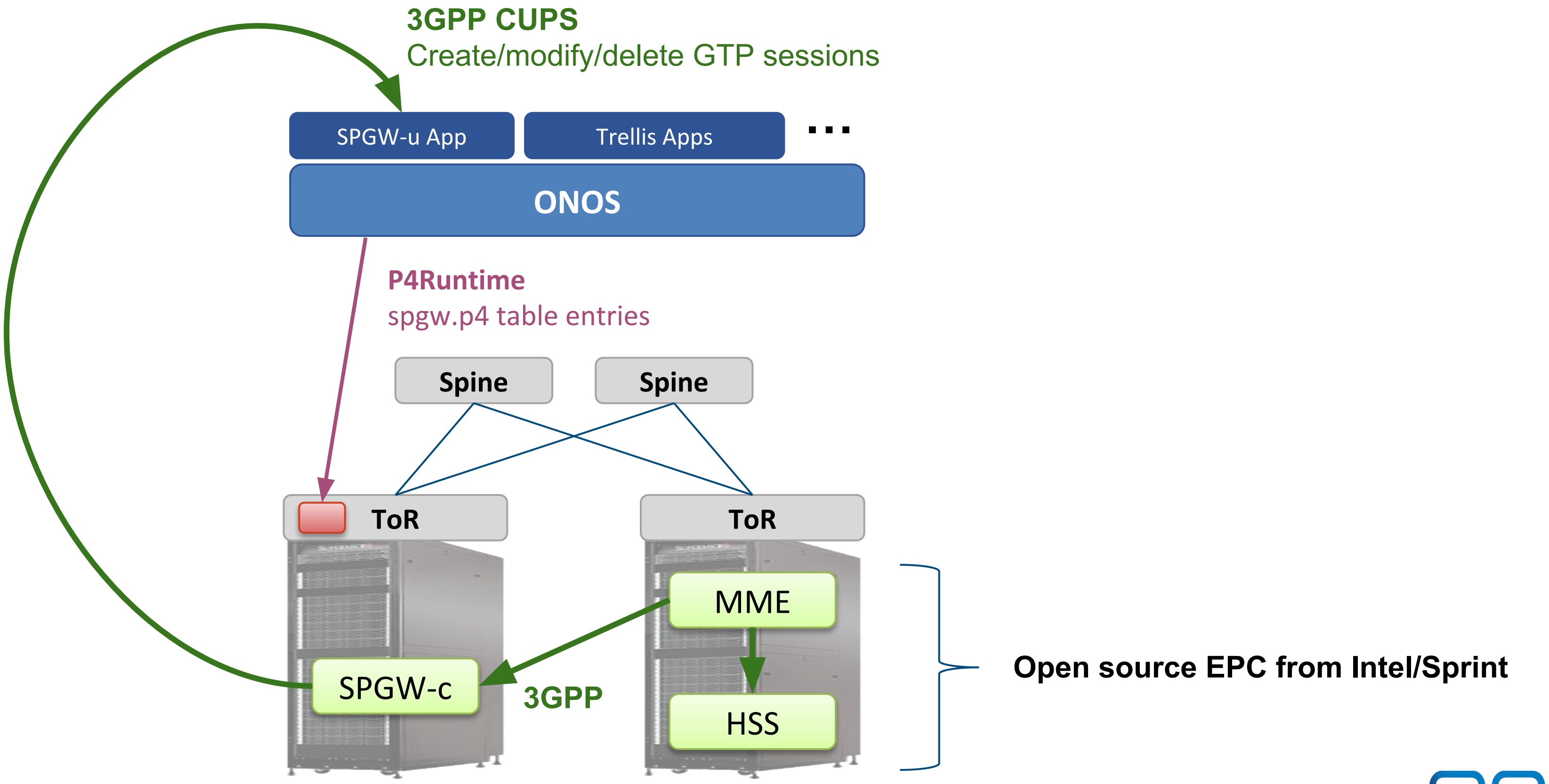
Hashed

Multicast

spgw.p4

- **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
SPGW-u App
Trellis Apps
...
ONOS
P4Runtime
spgw.p4 table entries
Spine
Spine
ToR
ToR
MME
SPGW-c
3GPP
HSS
Open source EPC from Intel/Sprint

Switching ASIC vs CPU - What are the benefits?

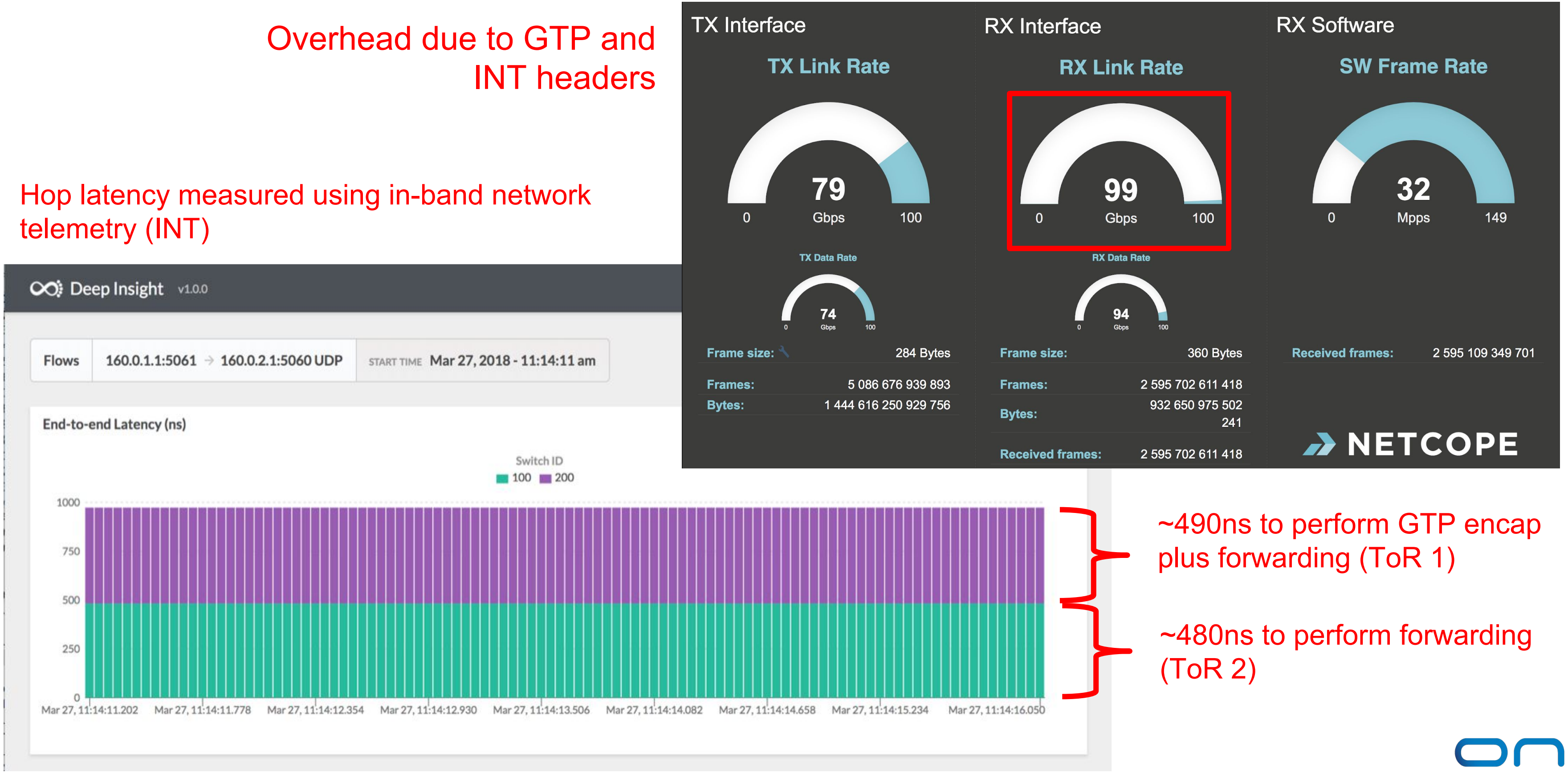
- **Maximized, deterministic throughput**
  - Always process traffic at line rate, with any traffic workload
- **Minimized, deterministic processing latency (and jitter)**
  - In the order of nanoseconds, with any traffic workload
- **Reduced power consumption**
  - Use less CPU resources, instead use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



TX Interface

TX Link Rate

79 Gbps (0–100)

TX Data Rate

74 Gbps (0–100)

Frame size: 284 Bytes

Frames: 5 086 676 939 893

Bytes: 1 444 616 250 929 756

RX Interface

RX Link Rate

99 Gbps (0–100)

RX Data Rate

94 Gbps (0–100)

Frame size: 360 Bytes

Frames: 2 595 702 611 418

Bytes: 932 650 975 502 241

Received frames: 2 595 702 611 418

RX Software

SW Frame Rate

32 Mpps (0–149)

Received frames: 2 595 109 349 701

NETCOPE

Deep Insight v1.0.0

Flows | 160.0.1.1:5061 → 160.0.2.1:5060 UDP | START TIME Mar 27, 2018 - 11:14:11 am

End-to-end Latency (ns)

~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?

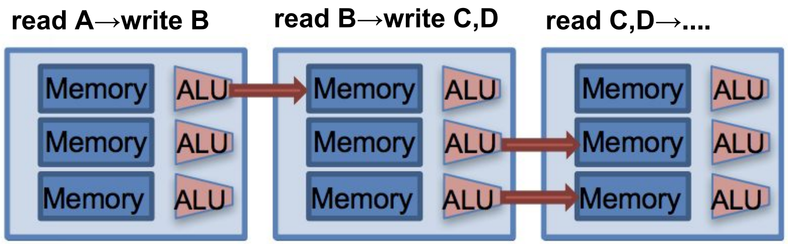
- **Per-subscriber state in SPGW**
  - GTP tunnel info, counters (billing), bearer mapping rules, etc.
- **Limited ASIC memory, allocated by the P4 compiler**
  - Number of subscribers depends on memory available, compiler optimizations
- **Size-speed trade-off in memories**
  - Fast on-chip memories 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?

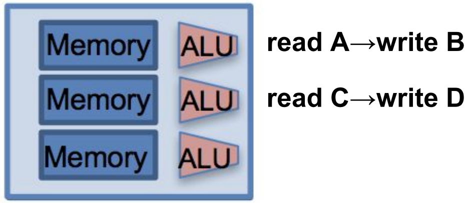
- **P4 chips have a fixed number of match-action stages**
  - Multiple simultaneous lookups and actions can be supported on each stage
  - Match/action dependencies call for sequential or parallel execution
- **Number of VNFs depends on match action dependencies**
  - ...compiler optimizations, and memory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

Sequential execution



3 stages

Parallel execution



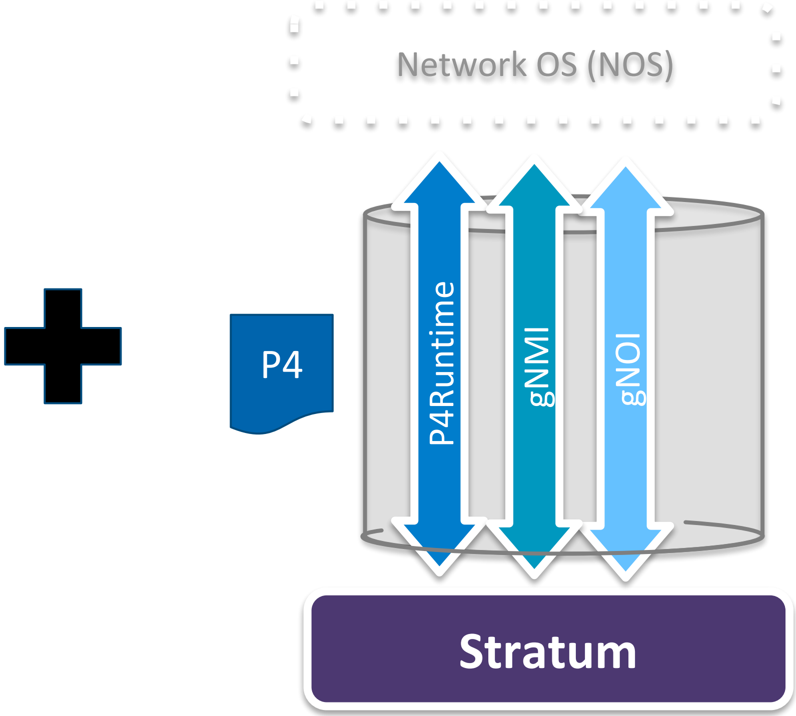
1 stage

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/20S9

P4 Brigade mailing list:

brigade-p4@onosproject.org

ONOS Code

https://github.com/opennetworkinglab/onos

ONOS wiki:

https://wiki.onosproject.org