SD-Fabric Tutorial

Carmelo Cascone
Charles Chan

Cloud Software Development Engineers, Intel
SD-Fabric Project TST Members, ONF

github.com/opennetworkinglab/sdfabric-tutorial
SD-Fabric: a Platform to Realize Your P4 Ideas

- Data center fabric entirely defined by software
  - Extend and modify to suit your needs

- Building blocks
  - Starter P4 program – with basic forwarding and advanced capabilities such as 5G UPF, INT, slicing, and more (if you need them)
  - Production-grade SDN control plane – with high-level APIs to build new control apps, while re-using built-in apps for common features
  - Integration with cloud-native and CI/CD tools – to make it easier to deploy and monitor the full stack in production
What is This Tutorial About?

- Learn about the architecture
- Deep-dive into existing features
  - Basic: L2/L3 forwarding
  - Advanced: P4-UPF, INT, slicing & QoS, ...
- Hands-on exercises based on an emulated environment
Resources

- **SD-Fabric 1.1 release**
  - Apache 2.0 open-source license
- **Learn more**
  - [Website](#), [Whitepaper](#), [Techinar](#), [Wiki](#)
- **Stay in touch**
  - [Mailing list](#), [Slack](#) (register)

https://docs.sd-fabric.org
Today’s Agenda

- Part 1 – Introduction to SD-Fabric: motivation, architecture, use cases
- Part 2 – Basics & Configuration + hands-on lab
- Part 3 – P4 User Plane Function (UPF) + hands-on lab
- Part 4 – In-band Network Telemetry (INT)
- Part 5 – Extending SD-Fabric
- Part 6 – Slicing & QoS
- Part 7 – Advanced Connectivity
- And more...

More sessions and labs on the way!
Make sure to watch the GitHub repo
github.com/opennetworkinglab/sdfabric-tutorial
Introduction to SD-Fabric

SD-Fabric Tutorial – Part 1
Era of the Multi-Cloud Connected Edge

- Multi-Cloud Applications
- IoT and 1000x Endpoints
- 5G Connected Devices
- AI and ML

Hyperscaler Cloud

Telco Edge

Enterprise Edge

IoT

Sensors

Surveillance

Multimedia

Employees

Visitors

SD-Fabric Tutorial

github.com/opennetworkinglab/sdfabric-tutorial
Needs of the New Datacenter (1/2)

- **Lines between servers and networks are blurring**
  - Rapidly increasing data demands
  - Programmable cluster of computing

- **Developer optimized**
  - Deeply Programmable via unified APIs
    - Distribute workloads on CPU, IPU or Switch
Needs of the New Datacenter (2/2)

- Needs to be **cloud managed**
  - Deployed and managed from the cloud
  - Easy to use gauges and dials for application visibility and control

- Need to consider traffic **end-to-end**
  - Include container network, software switch and IPUs

- Need to **orchestrate all these components**
  - Opportunity to build more resilient, secure and self-healing solutions
A Little Bit of History…
Evolution of ONF's open-source network fabric

- 2015 - Project Atrium
  - Demonstrate the feasibility of SDN and OpenFlow solutions by providing a complete software stack (app, controller, switch OS, SDK, ASIC...)

- 2016 - CORD network infrastructure
  - Focused on telco data center use cases
  - Introduced many new features

- 2017 - Trellis, Comcast collaboration started
  - Hardened platform, scaled up

- 2018 ~ 2019 - Field trial ~ Production deployment
  - Based on Broadcom OF-DPA switches
  - Start supporting programmable ASICs
    (Key design principle: keep controller app the same with Flow Objective API)

- 2020 ~ present - SD-Fabric
  - Integrate fully-programmable, fully-visible data plane
  - Aether as the driving use case (private-5G edge)
SD-Fabric Overview

Cloud-managed SDN Fabric as a Service

1. Cloud managed
   - Automatic provisioning, lifecycle management and failure recovery for all servers and switches
   - Manage multiple sites with state-of-the-art cloud-native CI/CD, monitoring, telemetry, logging and alert system

2. SDN Control Plane + Fabric Apps
   - No distributed protocol overhead
   - Global optimization
   - Control and data plane redundancy
   - One big router abstraction
   - Bridging, ECMP routing, ACL, LLDP, DHCP...

3. Fully programmable & visible data plane
   - Powered by Intel® Tofino™ Intelligent Fabric Processors (and potentially IPU later)
   - Network function offloading (SASE, tunnel termination, FWaaS, 5G UPF...)
   - In-band network telemetry & closed loop control
   - Resource optimization per use case
Two options to program/configure devices

- ONL + Stratum (via P4RT/gNMI)
  - Fully customizable P4 pipeline
  - Config, managed and monitored by SD-Fabric cloud management platform (more in the next few pages)

- Roadmap: SONiC + PINS (via P4RT/gNMI)
  - Require custom P4 table support to unleash the full potential of programmable data plane
SDN Control Plane & Fabric Apps

- Production grade, open source SDN control plane
  - Scalable & redundant
  - Support various southbound protocols (e.g., P4RT and gNMI)

- Example Fabric Apps
  - Forwarding
    - L2 Bridging, L3 ECMP routing, ACL
    - IPv4/IPv6 multicast
    - DHCP relay, dual-homing
  - 4G/5G User Plane
    - GTP-U tunnel termination
    - QoS
    - INT Watchlist

- Roadmap: hybrid SDN
  - Some features controlled by switch-local control plane (e.g., SONiC)
  - Some features controlled by remote SDN controller (ONOS)
Cloud Managed

- SDN Controller (ONOS)
- Fabric Apps
- API Gateway
- Control
- Telemetry
- Switches
- Servers
- Access (Wi-Fi, 5G, Broadband)
- Internet

- Helm (K8s package mgmt.)
- Jenkins/Terraform/Fleet (CI/CD automation) → Git (config repository)
- Config / Provision
- Telegraf (metric collector)
- Prometheus (time series database)
- Grafana (visualization)
- Monitoring
- Fluentbit (log collector)
- Elasticsearch (full-text search engine)
- Kibana (visualization)
- Logging / Alert
- intel® Deep Insight Network Analytics Software
- Verification Engine [WIP]

- Powered by state-of-the-art cloud native solutions
  - Adopt modern DevOps approaches
  - All operational information in a single pane of glass
- Backed by tremendous open-source ecosystem
Scalability and Redundancy

Single Switch
Minimal setup

Single Leaf Pair
Minimal HA setup

Leaf-Spine Fabric
With ECMP and N-way redundancy

Scale

github.com/opennetworkinglab/sdfabric-tutorial
SD-Fabric provides a solid foundation for developers to take advantage of programmable data planes to deliver innovative features
Vision: End-to-End Programmable Data Plane

**fabric.p4**
- Underlay forwarding
- UPF
- INT
- Slicing & QoS
- And more...

**nic.p4**
- Overlay/network virtualization
- Crypto/storage offloading
- UPF
- INT
- Slicing & QoS
- And more...

**app.p4**
- Container networking (K8s CNI)
- QoS classification
- INT
- And more...

**Linux netstack, P4-DPDK, etc.**

Switches | IPUs/DPUs | Host
--- | --- | ---

**Today** | **Roadmap**
Current Use Case: Private-5G Connected Edge

5G Core

Operation & Control
Site, subscribers, and apps configuration

ONOS

Verification Engine (roadmap item)

Smart alarms

Configuration state

Network state

Anomalies and per-flow stats

Smart alarms

Data plane telemetry (INT)

Intel® Deep Insight Network Analytics Software

End-to-end INT visibility for tracking path, drops, latency, and congestion (prototype eBPF impl. for host, NIC in roadmap)

Distributed UPF data path fully offloaded to Tofino: GTP-U tunnel termination, app filtering, and more.

DPDK-based UPF for horizontal scalability

5G Core

5G small cells

Edge Apps

Edge Apps

Self-adaptation and healing

ONOS

UPF App
One-Big-UPF API (PFCP)

Slicing App
Classification and QoS API

INT App
Anomaly thresholds and watchlist API

Cloud / On-Prem

On-Prem

Site, subscribers, and apps configuration
Next

- Part 1 – Introduction to SD-Fabric: motivation, architecture, use cases
- **Part 2 – Basics & configuration + hands-on lab**
- Part 3 – P4 User Plane Function (UPF) + hands-on lab
- Part 4 – In-band Network Telemetry (INT)
- Part 5 – Extending SD-Fabric
- Part 6 – Slicing & QoS
- Part 7 – Advanced Connectivity
- And more…

More sessions and labs on the way. Make sure to watch the GitHub repo [github.com/opennetworkinglab/sdfabric-tutorial](https://github.com/opennetworkinglab/sdfabric-tutorial)
Notices & Disclaimers

- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure.
- Your costs and results may vary.
- © Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.