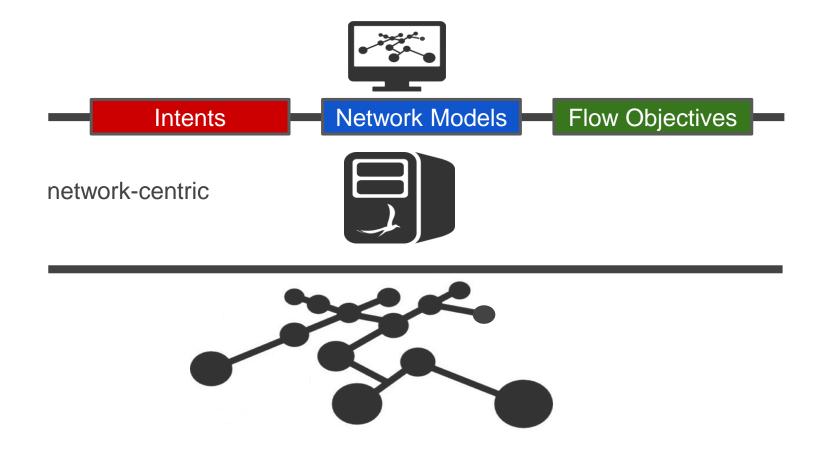# ONOS Northbound

James Won-Ki Hong, Jian Li, Seyeon Jeong

Dept. of Computer Science & Engineering

POSTECH

http://dpnm.postech.ac.kr/~jwkhong
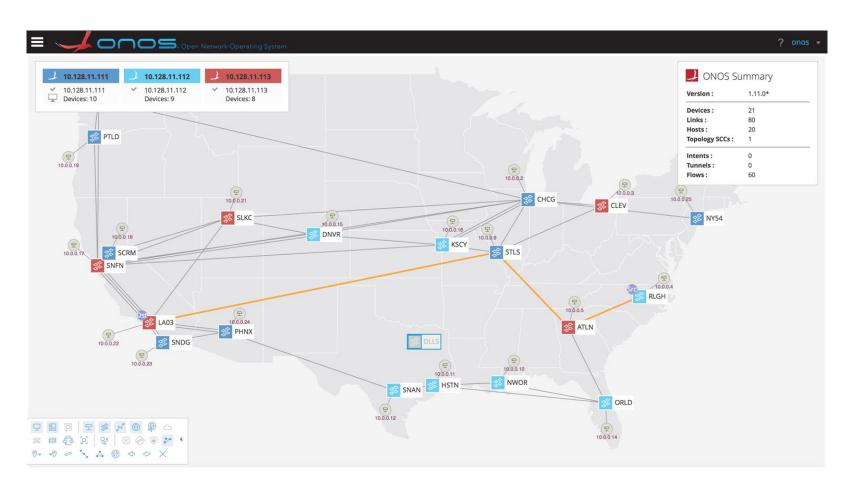
jwkhong@postech.ac.kr

Northbound

Interact with GUI

❖ **ONOS Web GUI (http://<onos-ip>:8181/onos/ui)**

- ▪ A single-page web application

Interact with ONOS CLI

❖ **ONOS CLI**
- An extension of Karaf's CLI
- Leverage features such programmatic extensibility

❖ **Access CLI**
- Local installation
  - Access with the command **onos-karaf clean**
- Remote installation
  - Deploy with **onos-package** and **stc setup**
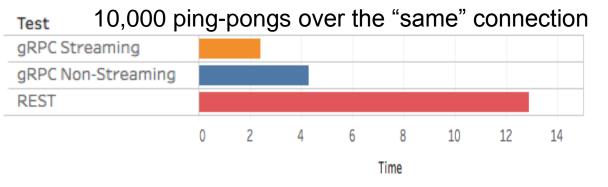 - Access with the command **onos**

Interact with ONOS REST and gRPC

❖ **REST API**

- Provides a way to interact with off-platform applications
- JSON, HTTP/1.1 based communication
- Swagger based REST documents

❖ **gRPC**

- Faster access than REST calls by using HTTP/2 connection multiplexing and bidirectional streaming
- Significantly reduced data size with binary formatted data using protobuf model
- Remote access to service interfaces with granularity similar to Java APIs

flows : Query and program flow rules

DELETE	/flows/application/{appId}
GET	/flows/application/{appId}
DELETE	/flows
GET	/flows
POST	/flows
DELETE	/flows/{deviceId}/{flowId}
GET	/flows/{deviceId}/{flowId}
GET	/flows/{deviceId}
POST	/flows/{deviceId}

10,000 ping-pongs over the "same" connection

Key Northbound Abstractions

❖ **Network Graph**
  ▪ Directed, cyclic graph comprising of infrastructure devices, infrastructure links and end-station hosts

❖ **Flow Objective**
  ▪ Device-centric abstraction for programming data-plane flows in table pipeline-independent manner
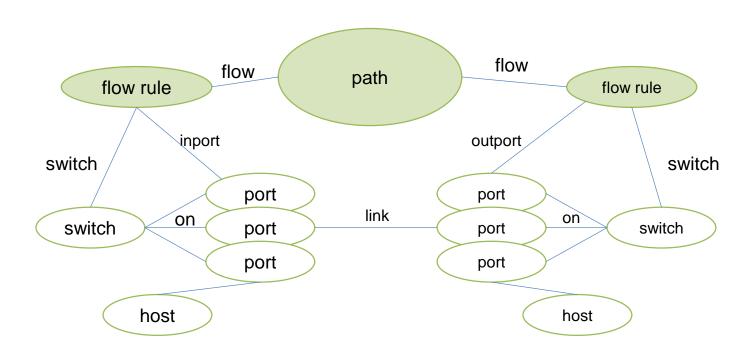
❖ **Intent**
  ▪ Network-centric abstraction for programming data-plane in topology-independent manner

Key Northbound Abstractions (1/2)

❖ **Network Graph**

- Directed, cyclic graph comprising of infrastructure devices, infrastructure links and end-station hosts
- Abstract the protocol-specific network element into protocol-agnostic network element (referred as **Model Objects**)
- Applications are only exposed to Model Objects
- Model Objects
  - Topology
    - Device, port, hosts, link, etc.
  - Control
    - Flow rule, role value
  - Packets
    - Outbound/inbound packet
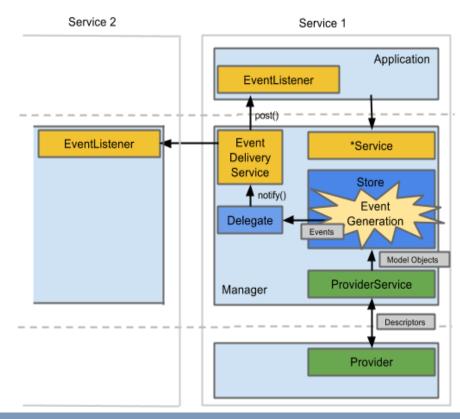
Key Northbound Abstractions (2/2)

❖ **Descriptions**
- Used to pass information about an element across the SBI
  - Made up of one or more model objects
  - E.g., HostDescription contains information about a host's MAC and IP address

❖ **Events**
- Used by managers to notify its listeners about changes in the network
- Structure
  - Event type
  - Subject built of model objects
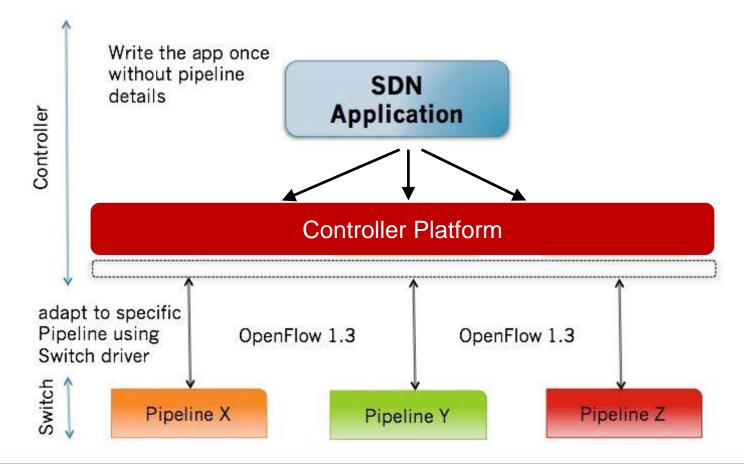- Event dispatch
- Event listeners

❖ **Problem**

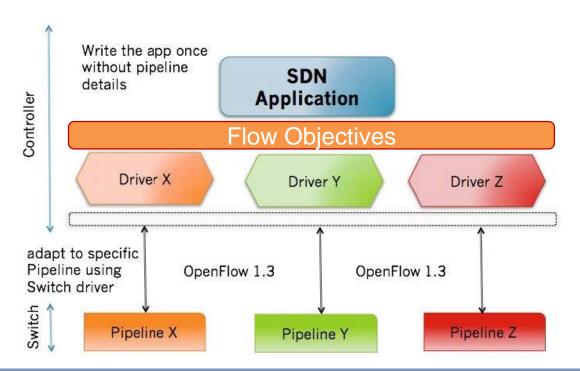- Applications must be pipeline aware, make them applicable to specific HW

Flow Objective Subsystems (2/2)

❖ **Flow Objective**

  ▪ Flow Objectives enable developers to write applications once for all pipelines

  ▪ Flow Objectives describe a SDN application's objective behind a flow it is sending to a device

  ▪ Three Types of Flow Objectives

    • Filtering objective

    • Forwarding objective

    • Next objective
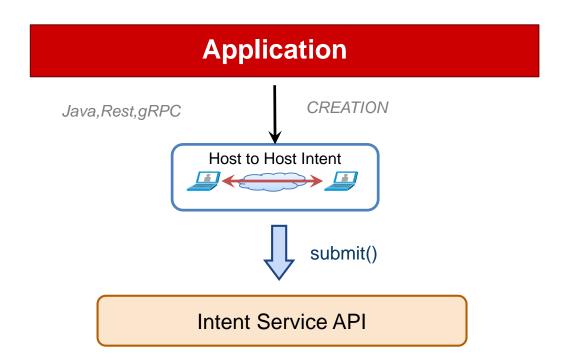
Intent Framework (1/2)

❖ **Intent Framework**

- Providers high-level, network-centric interface that focuses on **what** should be done rather than **how** it is specifically programmed
- Abstracts unnecessary network complexity from applications
- Maintains Requested semantics as network changes

❖ **Abstracts**

- Programming abstraction
  - Intents
  - Compilers
  - Installers
- Execution framework
  - Intent service
  - Intent store

Application

Java,Rest,gRPC   *CREATION*

Host to Host Intent
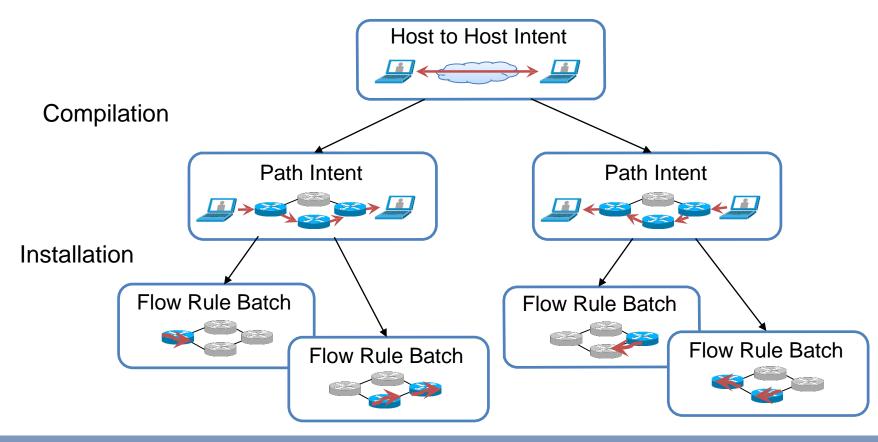
submit()

Intent Service API

Intent Framework (2/2)

❖ **Compiler & Installer**
- Compiler: produce more specific Intents given the environment
- Installer: transform Intents into device commands

Network Programming

Abstract to concrete

Intent

DC Clos Fabric	Packet/Optical WAN	Enterprise Campus

Flow Objective

OFDPA Pipeline	Single Table Pipeline	SpringOpen Pipeline

Flow Rule

OF 1.0	OF 1.3	Netconf	TL1

Configuration

❖ **Network Configuration (netcfg)**
  - Provides mechanism for any service to register and receive configuration

❖ **Device Configuration**
  - Behaviors abstract the management and configuration aspects of a device

❖ **Dynamic Configuration (WIP)**
  - Enable YANG-based service models to be introduced at runtime
  - Allow applications to implement dynamic services