

OpenFlow, Controllers, and Intents

Raouf Boutaba

David R. Cheriton School of Computer Science
University of Waterloo

Rogers TEP Workshop 3: Transport Networks
Concepts #2

The SDN Control Plane

Recap of SDN planes

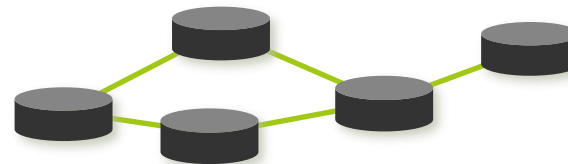
Control apps



Control Plane

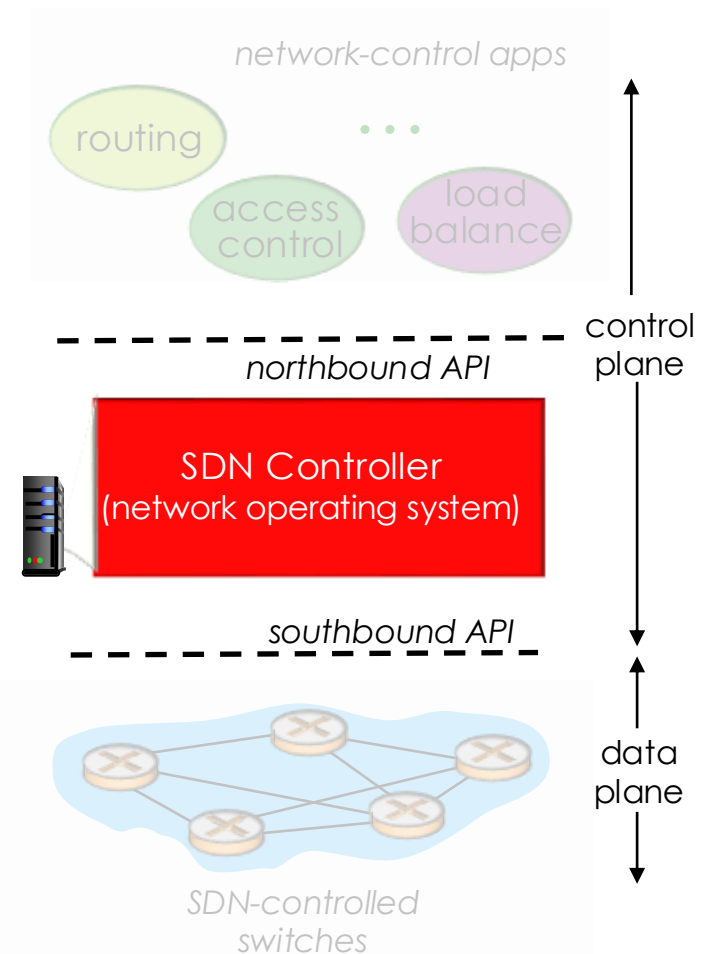


Data Plane

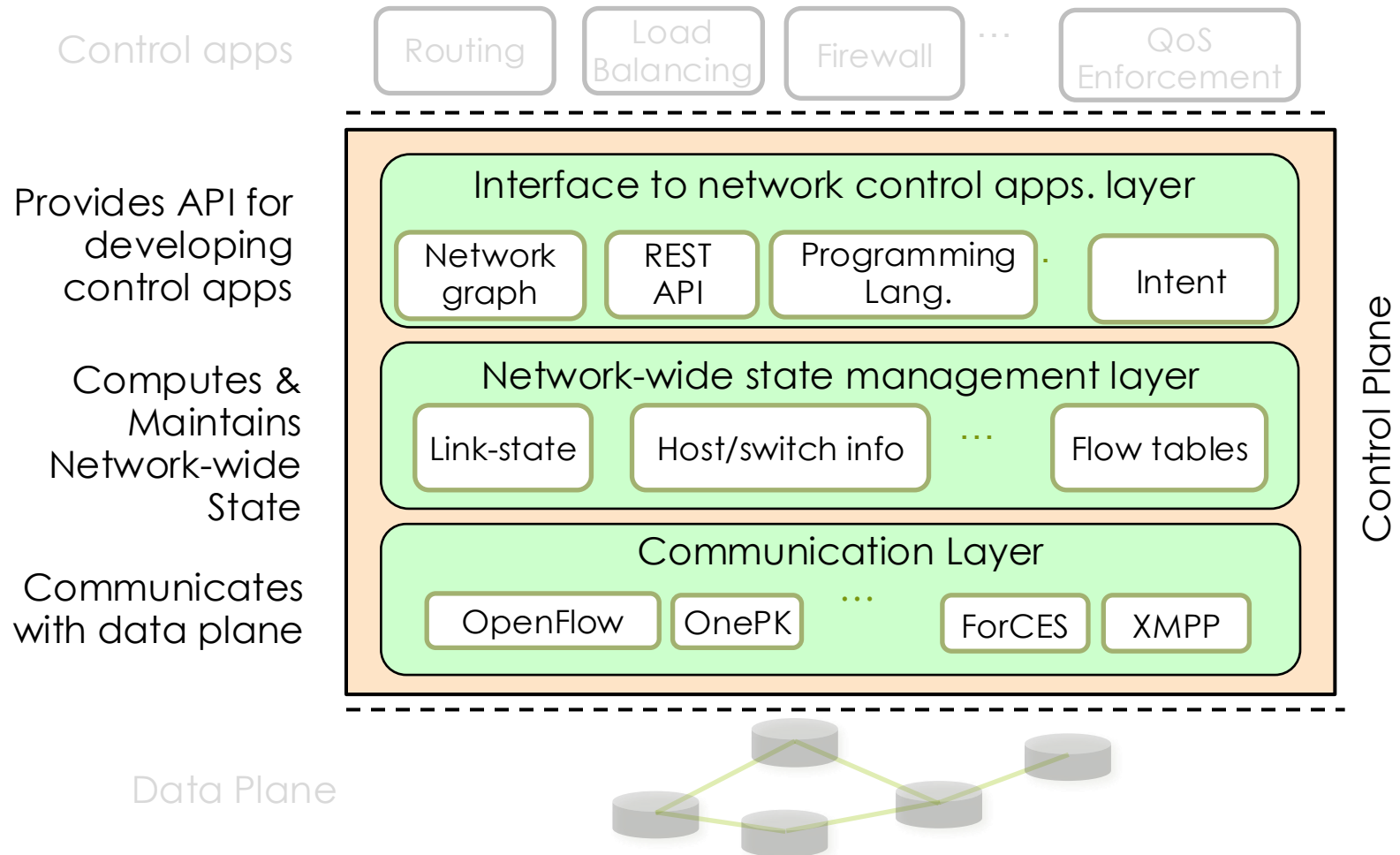


SDN Controller

- Network OS
 - logically centralized software to control data plane switches
 - maintain network state information
 - interacts with network control applications “above” via northbound API
 - interacts with network switches “below” via southbound APIs
 - supports one or more SB-APIs
- Controllers
 - OpenFlow: NOX/POX, Floodlight, OpenDaylight, ONOS, etc.
 - Other: Path Computation Element (PCE) in MPLS/GMPLS, ForCES, etc.

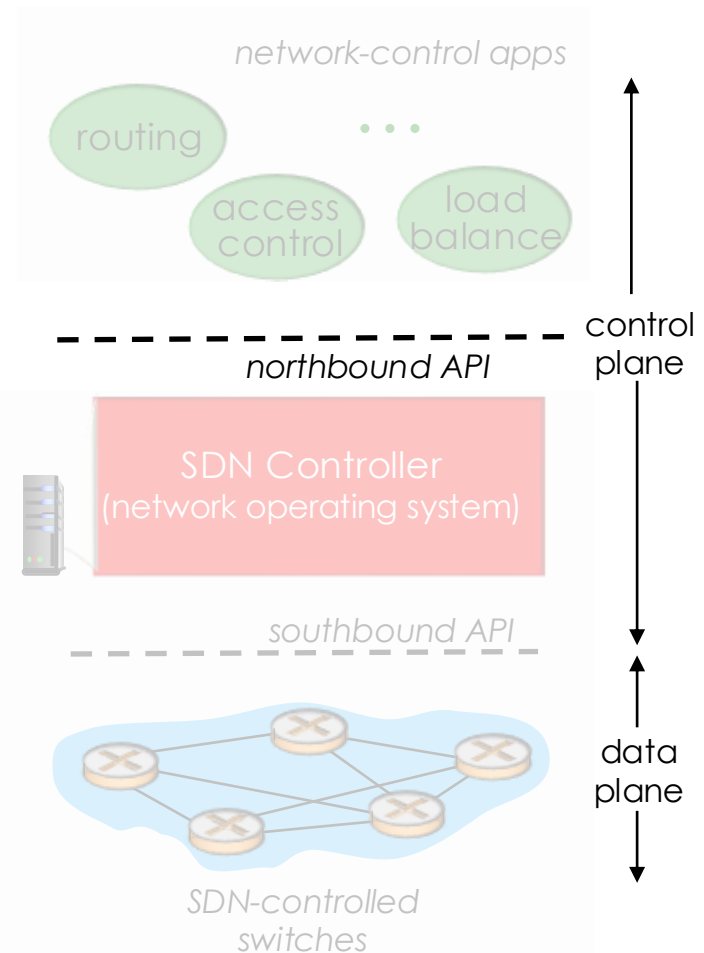


Components of SDN Controller



North Bound API

- Used by control apps to communicate with the controller
 - obtain network state
 - flow table content, counters, device health, etc.
 - modify network state
 - add/remove flow table entry, turn on/off a switch port, etc.
- Used by controller to provide network-wide view to control apps



What does a Controller do?

Control Plane Functions

- ▣ Topology Management
 - ▣ runs topology discovery protocol to discover network devices and their interconnections
- ▣ Path computation
 - ▣ uses topology information to compute shortest path and setup necessary forwarding rules for routing flows.
- ▣ Notification Management
 - ▣ receives, processes and forwards network event information to applications
- ▣ Statistics Collection
 - ▣ collects statistics from network devices by reading different counters
- ▣ Device Manager
 - ▣ configures network devices

Control Channel

- Each OpenFlow switch connects to the SDN controller over a TCP connection
- OpenFlow protocol defines a set of messages between the switch and the controller that allow:
 - Controller to program flow tables on the switches
 - Controller to query information from switches
 - Switches to send asynchronous notifications to the controller
- Control channel can be
 - In-band: Switches have default rules to forward control packets
 - Out-of-band: Control network can be a non-SDN network

OpenFlow key messages

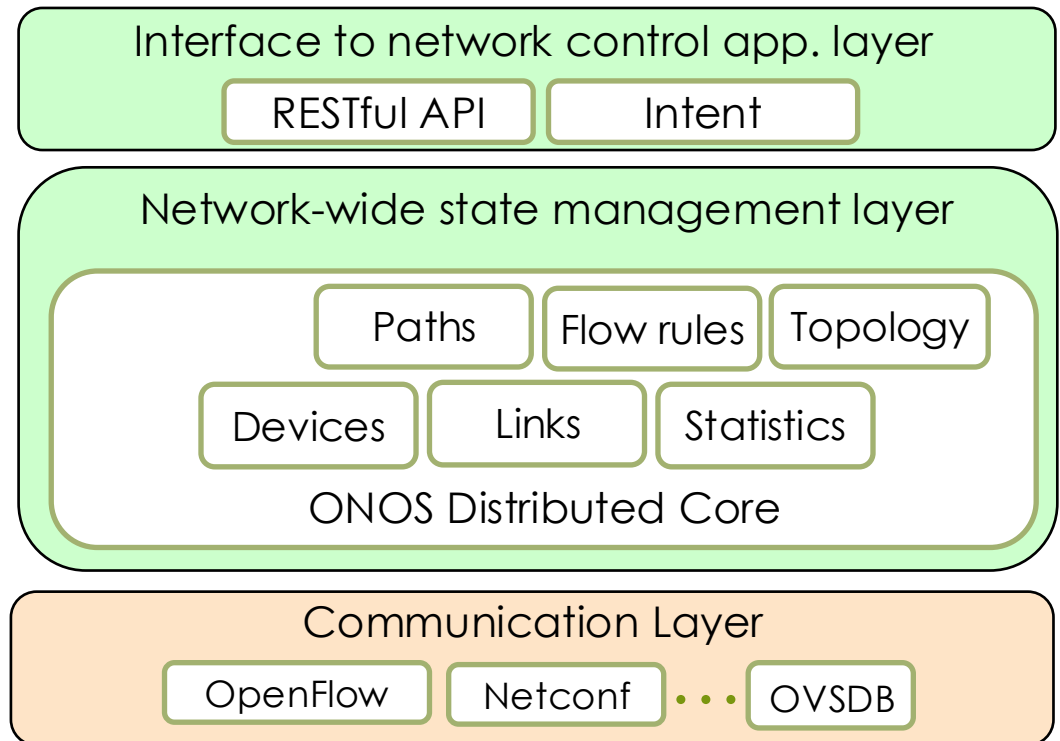
■ Switch-to-controller messages

- *PacketIn*: When a packet does not match any flow table entry the switch encapsulates the packet in a PacketIn message and sends it to the controller. This message is a request from the switch to the controller for path setup
- *FlowRemoved*: When a flow table entry expires a FlowRemoved message is sent to the controller
- *PortStatus*: A switch uses this message to inform controller about any status change of a port (e.g., port went down)

The ONOS Controller

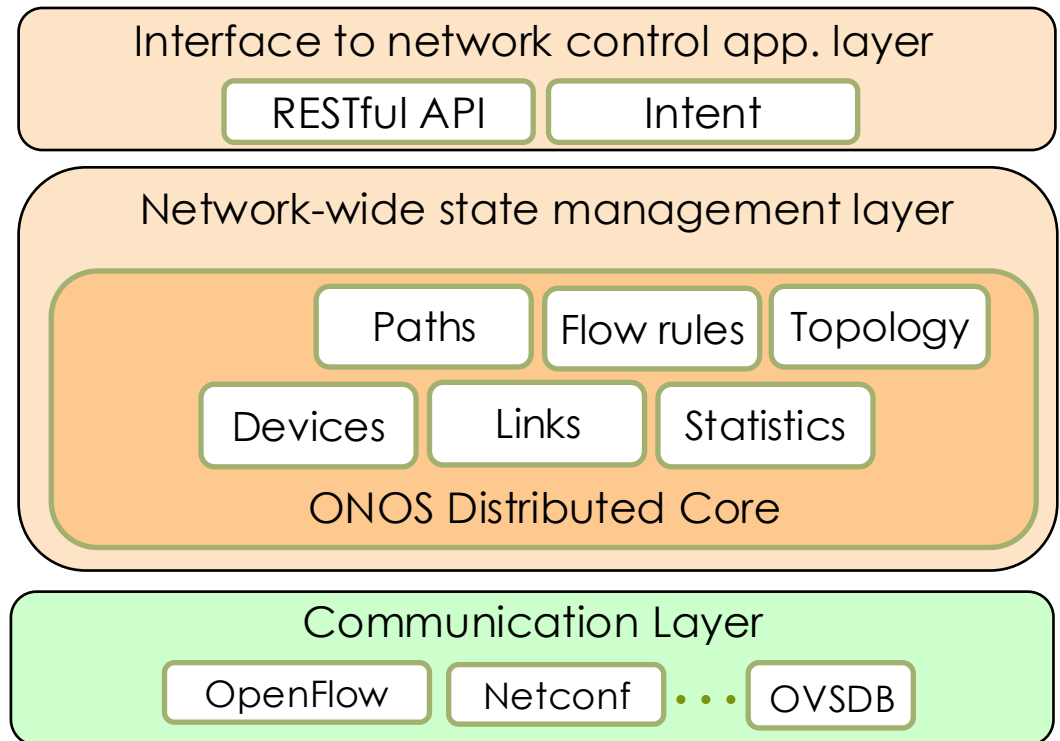
ONOS Controller

- designed as a distributed controller
- Communication layer
 - mostly supports OpenFlow
 - limited support for other southbound APIs



ONOS Controller

- ONOS ships with some basic network services
- Control apps external to controller
- Intent framework
 - high-level specification of service
 - *what* rather than *how*
- Distributed Core
 - service reliability
 - replication
 - performance scaling



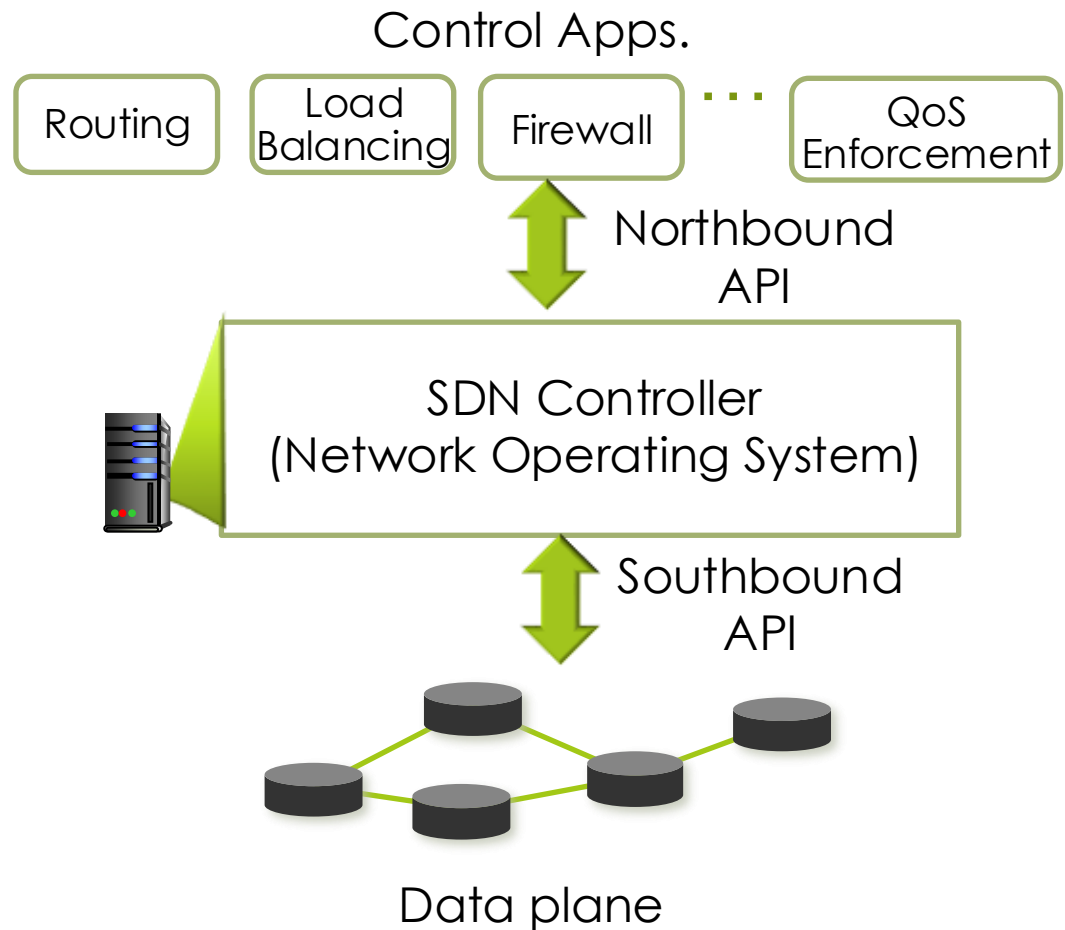
SDN API Overview

- **Northbound API** is for communication between control plane and control apps

- No standards yet

- **Southbound API** facilitates communication between SDN control and data plane

- OpenFlow is the *de facto* standard



Northbound API

- Northbound API exists in many forms
 - Programming Language
 - Intent Language
 - RESTful API/Low level sockets programming
 - ...

Intent

- Intent: Policy based derivatives describing “*what*” needs to be performed rather than “*how*” it needs to be performed

What:

Ensure connectivity between switch A and switch B

How:

- Setup flow between switch A and B
- Continuously monitor all switches on the path
- If something fails re-compute and reinstall flow rules

- Intent abstracts low level complexity from control applications
- Control applications specify their desired behavior through an intent framework supported by the controller
- Controller compiles intents to low level operations (e.g., flow installation) and takes the necessary actions

Example: ONOS Intent Framework

- allows control applications to express intent by specifying different constraints on network resources
- has two major components
 - Intent Compiler: Translates intents to installable actions on the network
 - Intent Coordinator: Determines how the network will be programmed/monitored to implement the intent

Traffic Engineering

- Traffic Engineering (TE) is done for different objectives
 - Maximizing network utilization
 - Ensuring QoS
 - Load balancing
 - Minimizing power consumption
- SDN is a very good fit for TE as
 - the controller maintains the state of the entire network
 - the optimal path for a traffic flow can be determined at a central point
 - traffic can be steered through specific paths i.e., the basis of transport slicing

From Theory to Practice

- You now have the full picture
 - Data plane: OVS switches forwarding packets
 - Control plane: ONOS managing the network automatically
 - Northbound API: REST interface to query and program ONOS
 - Intents: high-level connectivity expressed as "what, not how"
- Lab 2: Network programmability with ONOS
 - Connect Mininet topology to ONOS
 - Query topology via REST API in Python
 - Use ONOS to reroute traffic on link failure