Controlling a Software-Defined Network via Distributed Controllers

Volkan Yazıcı, Oğuz Sunay, Ali Özer Ercan
Özyeğin University, İstanbul, Turkey
{volkan.yazici,oguz.sunay,ali.ercan}@ozyegin.edu.tr

2012 NEM Summit
October 16-18, 2012
İstanbul, Turkey
Outline

● Problem Statement
● Traditional Networks & Software-Defined Networks
● Motivation & Challenges
● The Proposed Architecture
  – Controller Service Discovery
  – Inter-Controller Routing
  – High-Availability
● Experimental Setup & Results
● Conclusions
Problem Statement

how to architect a **software-defined network**

distributed across wide-area regions

subject to

→ minimal or no changes at the switching hardware
→ compatible with OpenFlow protocol
Traditional Computer Networks

Switching equipments decide & route individually

- Partial view of the network
- Sub-optimal performance

Every new feature is implemented on vertically integrated machines

- High hardware costs
- Almost impossible to innovate
- No room for customized solutions
Software-Defined Networks

Control Plane

Data Plane

Network Operating System

- Metering
- Routing
- QOS
- Monitoring

Openflow
- Simple Packet Forwarding Hardware

Openflow
- Simple Packet Forwarding Hardware

Openflow
- Simple Packet Forwarding Hardware

Nick McKeown, “How SDN will Shape Networking”, ONS’11
Motivation

A logically centralized control-plane can be implemented using a single controller or multiple controllers due to wide-area geolocations. We focus on distributed multiple controllers.

Concerns:
- reliability
  - single-point-of-failure
- scalability
  - handling the traffic of hundreds of switches
- performance
  - observed latency in the wide-area
Challenges

Scalability

- add new controllers to the cluster on-the-fly

Reliability

- save switches connected to failed controllers

Hard to achieve in practice

- synchronization of individual states
- distributed coordination & cooperation
- controller network failures & recovery (split-brain problem)

In domain challenges

- how many changes do you need to make in the
- controller software,
- OpenFlow protocol,
- switch?
The Proposed Architecture

Cluster Layer

Controller Machine
Controller Machine
Controller Machine
Controller Machine

Controller Software
Controller Software
Controller Software
Controller Software

Switch
Switch
Switch
Switch
Switch
Switch
Switch
Switch
Switch

Cluster Comm. Layer
backed by IGMP and Hazelcast

OpenFlow Protocol
Controller Service Discovery

Each switch is initially assigned a static IP address denoting the controller it is supposed to connect.
Controller Service Discovery

Using inter-controller communication, controllers assign these virtual IP addresses to their switching network interfaces.
Controller Service Discovery

when a switch tries to connect to its controller, it gets redirected to the corresponding controller assigned to that virtual IP address.
Controller Service Discovery

when a switch tries to connect to its controller, it gets redirected to the corresponding controller assigned to that virtual IP address.

scheme requires no software/hardware changes.
Controller Service Discovery

When a switch tries to connect to its controller, it gets redirected to the corresponding controller assigned to that virtual IP address.

A controller can have many virtual IP addresses.

The scheme requires no software/hardware changes.
Inter-Controller Routing

Controller Machine

Controller Software

Controller Machine

Controller Software

Controller Machine

Controller Software

Controller Machine

Controller Software

a flow hits to the network
Inter-Controller Routing

a flow hits to the network

reactive approach: just program the switches accessible by the current controller
Inter-Controller Routing

A flow hits to the network

Proactive approach: using inter-controller comm. program all switches end-to-end
High-Availability

Controller Machine

Controller Software

Controller Machine

Controller Software

Virtual Controller
IP Address

Switch

Controller Software
High-Availability

controller failure
High-Availability

controller failure

another controller gets attached to the virtual IP address(es) of failed controller
Experimental Setup

- 4 controller machines
- Unmanaged gigabit switch
- 5 PCs, each emulating traffic of 64 switches using cbench to emulate the switches
- An in-house developed distributed OpenFlow controller forked from Beacon and based on Java
Experimental Results

Time it takes to move switches from a controller to another.

Number of responses processed per second per switch.
Conclusions

• In a wide-area SDN deployment, distributed controllers are preferred due to reliability, scalability and performance (latency) constraints.

• Globally optimal routing decisions are possible with globally shared network map.

• The network map is distributed using (peer-to-peer and 1-hop) distributed data structures.

• Switch migration scheme without any software/hardware changes is proposed.

• Showed that switch migration is fast.

• A strong candidate as a distributed SDN architecture.
Thanks...

Questions?