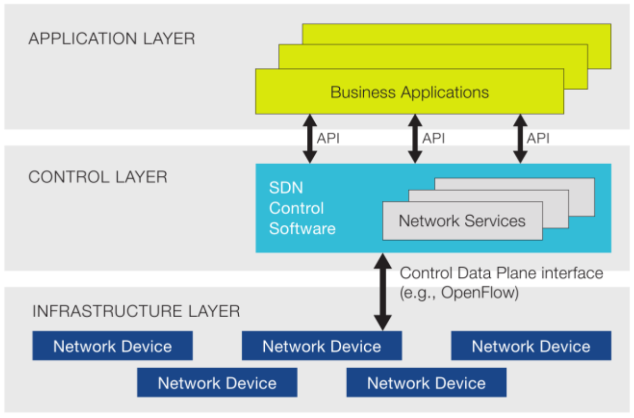
Core OpenFlow Technical Training

ONLINE COURSE INTRODUCING OPENFLOW SWITCHING AND CONTROLLER PROGRAMMING



NEW TOOLS, NEW METHODS, NEW IDEAS

Software Defined Networking (SDN) provides a whole new toolset for solving network problems. But these new tools will only generate the desired results when placed in the hands of those with the skills to use them wisely and who understand how these new tools impact the way network problems are solved. This course is intended to provide hands-on experience with key SDN tools, techniques and problem solving methodologies, using nothing more than standard PC and online resources.

KEY FEATURES

Two OpenFlow Multi-Table Pipeline Environments

- Virtual network of switches and hosts
- OpenFlow Controller
- OpenFlow application code implementing an L2 learning switch
- OpenFlow application code implementing an enhanced L2 switch (state in switch) and with security ACLs

A Complete Virtualized Development Workspace

- Implemented with Opensource technologies
- Code and algorithm compatible with NoviFlow switches

Two Levels of Certification

- SDN/OpenFlow Principles
- SDN/OpenFlow Architect Level 1

Instructor moderated

Get help if you get stuck

Printable Format Available

- Each Section in “book” form
- Over 200 pages of training material

THE VISION: VIRTUAL TO PHYSICAL

The **Core OpenFlow Technical Training** is NoviFlow training on setting up, managing and leveraging the capabilities of high-performance SDN switches based on programmable match-action pipeline technology.

Why an online course using a virtual environment on PCs?

Only two technologies implement the OpenFlow 1.3/1.4/1.5 specification at a high-compliance level – NoviFlow’s NoviWare NOS and the x86 Based Open vSwitch. Consequently, OpenFlow 1.3+ code developed in one environment will run in the other environment. This course leverages this compatibility to provide core skills training OpenFlow development on the widely-available PC environment. The knowledge acquired and code developed transfers directly to NoviFlow’s globally deployed NoviWare commercial switch environment.

By being delivered on standard PC tools, the course provides a solid foundation for higher level courses provided by NoviFlow that focus on specific capabilities of NoviFlow’s NoviWare NOS and NoviSwitch high-performance OpenFlow forwarding planes.

COURSE DESCRIPTION

This course is designed to enable students to acquire the core principles of an SDN/OpenFlow match-action pipeline technology by reading modules, installing tools, completing a series of assignments, and passing skills assessments. Students are assumed to have basic expertise in Networking and Python programming, and will additionally acquire multi-table, large rule-set system architecture skills.

NOTE: All information presented in this document is provided as is WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, and is subject to change without notice. Copyright © 2019, NoviFlow Inc.

The course embeds how to program OpenFlow switches within the broader context of crafting an SDN algorithm. It presents OpenFlow switches, OpenFlow controllers, and OpenFlow applications working together to create, deploy and manage SDN-based network solutions.

The course covers two multi-table OpenFlow applications in detail. Course Sections provide step-by-step walkthroughs for major design topics such as Pipeline Modeling, Event Driven Programming, Event Emitting Data Structures, moving a Network "State" from the Controller to the Switch, and using a REST Interface to probe a switch's internal Pipeline. The Course Section outline at the end of datasheet provides the Overview, Technology and Topic, plus Learning Outcomes for each Section.

The Course is delivered online so that students can progress at thier own pace. It is instructor moderated. Any questions can be directly communicated to the Course Leader. You will get help if you get stuck.

The course also walks the student through the construction of an x86 based Virtual Development Workspace that is consistent across x86 PC and compatible with NoviFlow switches (see below).

VIRTUAL DEVELOPMENT WORKSPACE

The *"consistent" x86* Development Workspace built over the first three Sections is a compelling aspect of this course. The consistency of the workspace across different PC configurations, enables a student's OpenFlow code and skills to be portable to any other PC running the workspace. And, the consistency of the workspace with OpenFlow standards enables the student's skills and code to transfer to high-performance NoviFlow switches.

The Workspace uses VirtualBox to isolate the development environment from the student's PC hardware environment. Ubuntu and Python provide a standard Linux OS and programming environment, and Mininet provides a network virtualization tool.

A step-by-step process across the first three of Section builds the Development Workspace. The student acquires skills in the process but does not have to be a power user on any of the technologies to be able to complete setting up the development Workspace.

WHO SHOULD TAKE THIS COURSE?

This **Core OpenFlow Technical Training** is an engineering, hands-on course that grants an **SDN/OpenFlow Principles Certification** after completion of the Assignments and passing the Skills Assesment testing.

The Course offers an optional Project for students who will be involved with daily OpenFlow algorithms design and application development coding. Completion of the Project demonstrates an advanced design and coding skill level and earns an **SDN/OpenFlow Architect Level 1 Certification**.

The course is designed for:

- Network application developers that are moving to SDN/OpenFlow
- CyberSecurity engineers interested in moving security functionality into the Network via SDN/OpenFlow
- Networking and CyberSecurity engineering management wanting to get an in-depth look at SDN/OpenFlow technology
- SDN engineers that are interested in moving to multi-table, OpenFlow 1.3/1.4/1.5 compliant technology after working with functionally limited OpenFlow technologies such as OF-DPA or OpenFlow 1.0 single table products

NoviFlow products are warranted according to the terms and conditions of the agreements under which they are provided. NoviFlow, the NoviFlow logo, noviflow.com, NoviWare, CyberMapper, NoviConnect and NoviSwitch are trademarks of NoviFlow Inc. All other product names, company names and trademarks mentioned herein are the property of their respective owners. Document #DS2017-COTT-02

COURSE DURATION

This Course covers a wealth of material. The main five Sections may take up to an estimated 15 to 30 hours to complete. Of course, that depends on your initial skill set, plus your interest in "drill down" and exploring.

The optional Project component adds an estimated 5 to 10 hours. The exact time required depends on: 1) how complete your initial algorithm design is, 2) how much Ryu Controller research you have to do to implement your design, and 3) how much Python research you will need to implement your design. i.e. the more technical learning required to implement your design the longer the Project will take.

COURSE PROGRESS AND SKILLS ASSESSMENT

Each of the main Sections of the course has an Assignment and a Skills Assessment Quiz. The cumulative grade scores from the Assignments and Assessments determine your qualification for an "*SDN/OpenFlow Principles Certification*."

The Project Section is an optional 'finals" test. It presents a required enhancement to the *"Simple Switch 2"* controller application that will require Pipeline Architectural and Python programming skills. Completion of the project determines qualification for *"SDN/OpenFlow Architect Level 1 Certification"*

REQUIREMENTS

Skills

Some familiarity with key technologies:

- Networking and switch configuration
- Scripting languages and Python
- Virtualization environment

This course works through building the development Workspace in a step-by-step process. Anyone will be able to follow along and finish with a working virtual image of the Workspace. If your technical skills are average, the process of creating the Workspace will be quick and the skills acquired deep.

If you are a novice, creating the Workspace will take a little longer than required by an active programmer. Also, in the code walkthrough assignments, you may be spending some time understanding Python as well as to build your skill in designing OpenFlow algorithms.

PC System Requirements

Minimal

- 64-bit, 4 CPU cores with hardware virtualization support
- 8GB RAM
- 100GB free disk space
- Screen/Window capture software such as Snagit

Recommended

- 64-bit, 8 CPU cores or 16 HTs with hardware virtualization support
- 16GB RAM
- 200GB free disk space
- Screen/Window capture software such as Snagit.

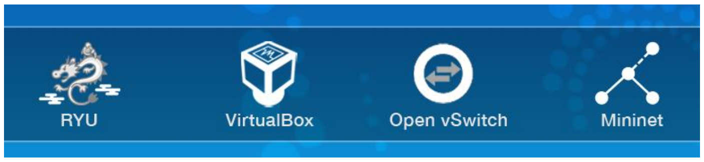
NOTE: All information presented in this document is provided as is WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, and is subject to change without notice. Copyright © 2019, NoviFlow Inc.

SECTION OVERVIEW

CREATING A DEVELOPMENT WORKSPACE

This is a “keystone” Section. A step-by-step process that guides you through building the Virtual Development Environment that will be used as the development environment for all other training modules in this course.

The unique feature of the Virtual Development Workspace is the high level of compliance with the OpenFlow 1.3/1.4/1.5 specifications. Thus skills, algorithms and code created in the virtual environment move directly to NoviFlow switches which are also highly compliant with OpenFlow 1.3/1.4/1.5.



Technologies and Topics

- VirtualBox
- Ubuntu
- Mininet
- Python
- Ryu Controller

Learning Outcomes

After working through this Section, you will be able to:

- Install and configure VirtualBox for an OpenFlow development environment
- Load Ubuntu Linux as the Operating System for the development VM
- Install full Mininet environment in developer VM
- Run installation verification for Mininet with the controller, one switch, two hosts.
- Install Ryu Controller in your development VM

INTERACTIVE RYU WITH POSTMAN

This Section provides the tools for a developer to see inside OpenFlow switches and modify switch behavior via the Ryu REST API. Querying the REST API for all OpenFlow switches in the network, a switch’s description, retrieving flow entries, etc. are presented in step-by-step detail. The REST API messages for modifying current flows, adding flows and deleting flows are also presented.

The Section presents a tool – Postman – that make the task of crafting REST API messages easy. Using Postman to create messages for all the above REST interactions is covered in detail. How to save these messages into Postman Collections is shown.

The example REST messaging is run on the MiniNet environment presented in the Creating a Development Workspace module above.

Technologies and Topics

- Postman application
- Ryu REST API
- JSON
- Postman collections
- Packaged set of Postman scripts to interact with OpenFlow switch via Ryu
- Test Suite for Postman Collection

Learning Outcomes

After working through this Section, you will be able to:

- Install the Postman application on your development VM
- Utilize the Ryu multi-component design by launching a Ryu application and the Ryu REST API at the same time
- Probe for network OpenFlow switches with a Postman crafted Ryu REST message
- Probe for switch description and flow status with Postman crafted Rye REST messages
- Add, modify, and delete flows in a switch with Postman crafted Rye REST messages
- Dynamically observe and change OpenFlow switch behavior with Postman crafted Rye REST messages
- Create a Postman collection and save it into the collection.
- Run test programs for Postman Collection of REST messages

NoviFlow products are warranted according to the terms and conditions of the agreements under which they are provided. NoviFlow, the NoviFlow logo, noviflow.com, NoviWare, CyberMapper, NoviConnect and NoviSwitch are trademarks of NoviFlow Inc. All other product names, company names and trademarks mentioned herein are the property of their respective owners. Document #DS2017-COTT-02

CUSTOM MININET TOPOLOGIES AND INTRODUCING ATOM

This Section provides the tools for creating virtual OpenFlow networks and the code that automates the process. The article specifically introduces how to virtualize realistic datacenter network topologies. Two new technologies – mininet for creating virtual networks and the atom IDE for editing code - are integrated into the Inside OpenFlow Development Workspace.

The module develops five topology models for creating virtual datacenter networks. These increase in complexity and configuration from the “Basic” configuration with one aggregator switch connected to racks with one top-of-rack switch and a configurable number of Host to the “Fully Redundant” configuration with multiple aggregator switches connected to multiple racks each containing 2 ToR switches and multiple Hosts.

Technologies and Topics

- Mininet virtual OpenFlow network environment
- Mininet Python libraries
- Python scripts that automate of building Mininet networks
- Virtual models of Datacenter networks
- Atom IDE for creation and maintenance of Python Mininet scripts

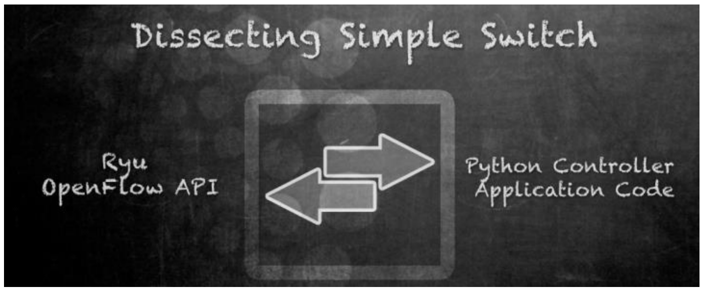
Learning Outcomes

After working through this Section, you will be able to:

- Install the Mininet virtual network platform
- Install Atom IDE and extension
- Create a Mininet project in the Atom
- Build and run a Python Ryu application script for the Simple Topology model
- Build and run a Python Ryu application script for the Basic Datacenter topology model
- Build and run a Python Ryu application script for the Configurable Datacenter topology model
- Build and run a Python Ryu application script for the Fully Redundant Datacenter topology model

UNDERSTANDING THE RYU API: DISSECTING SIMPLE SWITCH

In our previous Section, we covered everything needed to get started with SDN development using Ryu and Mininet. Now that our virtual workspace has all the development tools we need, it is now time to look into the real meat of Ryu - its API. Follow along as we discuss the detailed inner workings of Simple Switch as an OpenFlow application example programmed to the Ryu API. This Section includes detailed documentation and code references.



Technologies and Topics

- Anatomy of a Ryu Controller Application
- Structure of L2 Switch
- Terminology and Icon Legend
- Introduction of Ryu Event Handlers
- Detail links from Python application code to Ryu Modules, Classes, Attributes, Functions, etc.

Learning Outcomes

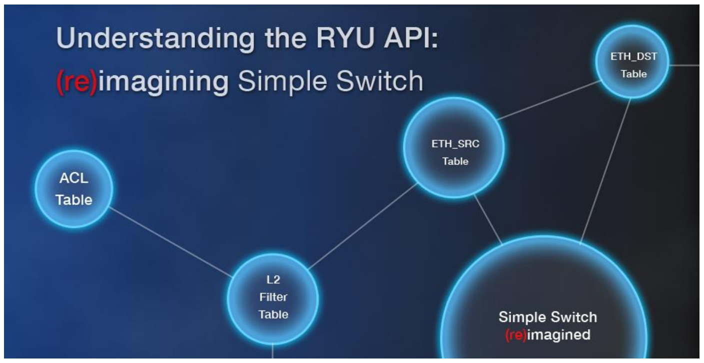
After working through this Section, you will be able to:

- Articulate the core structure of Ryu Controller Applications
- Articulate the functions of an L2 switch
- Articulate the terminology and icon used by Inside OpenFlow in dissecting Ryu applications
- Drill-down on New Switch Handler event code to find specific Ryu source code and documentation
- Drill-down on Packet-In Handler and Packet Dissection Handler to find specific Ryu source code and documentation
- Drill-down on Constructing and Adding a Flow Entry to find specific Ryu source code and documentation

NOTE: All information presented in this document is provided as is WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, and is subject to change without notice. Copyright © 2019, NoviFlow Inc.

UNDERSTANDING THE RYU API: REIMAGINING SIMPLE SWITCH

This Section "reimagines" the L2 Simple Switch using features found in the modern OpenFlow 1.3 or later controllers and switches. Two significant changes are made to the previous Simple Switch architecture: 1) a multi-table design is used, 2) the MAC learning state is completely moved to the OpenFlow switches. The multi-table design enables a more sophisticated solution that includes ACLs and packet filtering. Moving the large MAC learning tables out of the controller makes the application scalable.


Understanding the RYU API:
(re)imagining Simple Switch
ETH_DST Table
ETH_SRC Table
ACL Table
L2 Filter Table
Simple Switch (re)imagined

Technologies and Topics

- Multi-table OpenFlow pipeline
- Non-blocking packet to Controller algorithm
- Stateless L2 Controller Application

Learning Outcomes

After working through this Section, you will be able to:

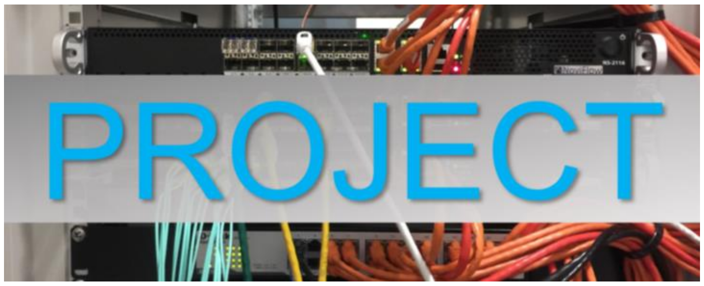
- Install "Simple Switch 2" in a "VirtualBox environment
- Create a realistic, virtual Data Center network Mininet
- Test "Simple Switch 2" using the virtual Data Center network
- Simulate a Host moving from one network segment to another and validate that "Simple Switch 2" handles the event
- Use Postman with the Inside OpenFlow queries to probe the internal state of the OpenFlow switches

PROJECT SECTION (OPTIONAL)

This is an optional Section. The Assignment is to enhance the security algorithm within "Simple Switch 2" by developing code that blocks certain types of traffic.

The project requires algorithm design and Python coding and is targeted at the student who will be SDN/OpenFlow development engineers.

The Project defines a set of new ACL requirements to be added to Pipeline. A mininet testing environment is provided to validate that the new code implements the required ACLs.



Learning Outcomes

After completing the Project you will have designed a "Simple Switch 2" algorithm enhancements and crafted new Python code that implements the following security actions:

- Drop packets with a specific MAC address
- Drop packets with a specific MAC OUI
- Drop packets with a specific IP address (requires eth_type test)
- Drop packets within a block of IP addresses (requires eth_type test)
- Drop packet with a specific TCP port, UDP port (requires ip_proto test)
- Drop ICMP echo requests, but not other ICMP messages (allows pinging in one direction, eth_type test required)

NOVIFLOW PART NUMBER

900-100-000	Core OpenFlow Technical Training (online course)

NoviFlow products are warranted according to the terms and conditions of the agreements under which they are provided. NoviFlow, the NoviFlow logo, noviflow.com, NoviWare, CyberMapper, NoviConnect and NoviSwitch are trademarks of NoviFlow Inc. All other product names, company names and trademarks mentioned herein are the property of their respective owners. Document #DS2017-COTT-02