



March 30, 2017

Pranav Deshpande  
TE3,3315

# Software-Defined Networking Using OpenFlow Protocol v1.3

---

Concept of Redefined Networking



# Motivation

---

The programmability and centralized control of the network topology in SDN allow enterprises to incorporate various applications easily to

- ▶ **Improve efficiency, reduce complexity,**
- ▶ **Streamline processes, and provide superior**
- ▶ **User experience.**



# Presentation Flow

---

- ▶ Software-Defined Networks concept.
- ▶ Difference between Traditional networks and SDN.
- ▶ Software-Defined Network Architecture.
- ▶ OpenFlow Protocol v1.3
- ▶ Open Daylight Controller
- ▶ OpenVSwitch
- ▶ About tool GNS3.
  - ▶ Docker Container.
  - ▶ Peculiar features of GNS3
  - ▶ Implementation/Simulation
- ▶ Conclusion



# Concept of Software-Defined Networks

---

- ▶ The goal of Software-Defined Networking is to enable cloud and network engineers and administrators to respond quickly to changing business requirements via a centralized control console.
- ▶ The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.
- ▶ Decouples the network control and forwarding functions.
  - ▶ Network Application
  - ▶ Control Plane
  - ▶ Forwarding/Data Plane

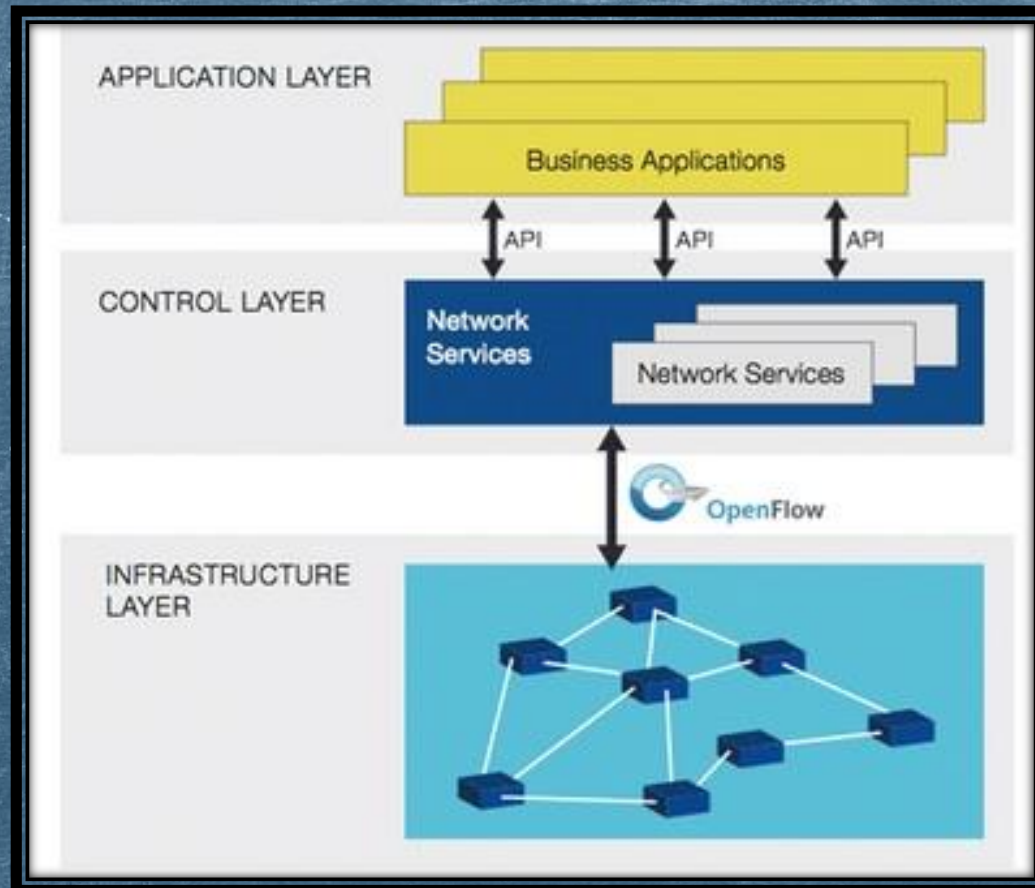


# Traditional networks vs Software-Defined N/W

Traditional Networks	Software-Defined Networks
They are Static and inflexible networks. They are not useful for new business ventures. They possess little agility and flexibility	They are programmable networks during deployment time as well as at later stage based on change in the requirements. They help new business ventures through flexibility, agility and virtualization.
They are Hardware appliances.	They are configured using open software.
They have distributed control plane.	They have logically centralized control plane.
They use custom ASICs and FPGAs.	They use merchant silicon.
They work using protocols.	They use APIs to configure as per need.



# SDN Architecture



► **Controllers**

► **Southbound APIs**

► **Northbound APIs**



# Software-Defined Networks Myth

---

## ► Software-Defined Networking is Not OpenFlow

Often people point to OpenFlow as being synonymous with software-defined networking, but it is only a single element in the overall SDN architecture.

SDN=OpenFlow -> No

SDN=Centralization of CP ->No

SDN=Southbound API ->NO

*“SDN is a framework to solve set of problems”*



# Software-Defined Networks benefits

---

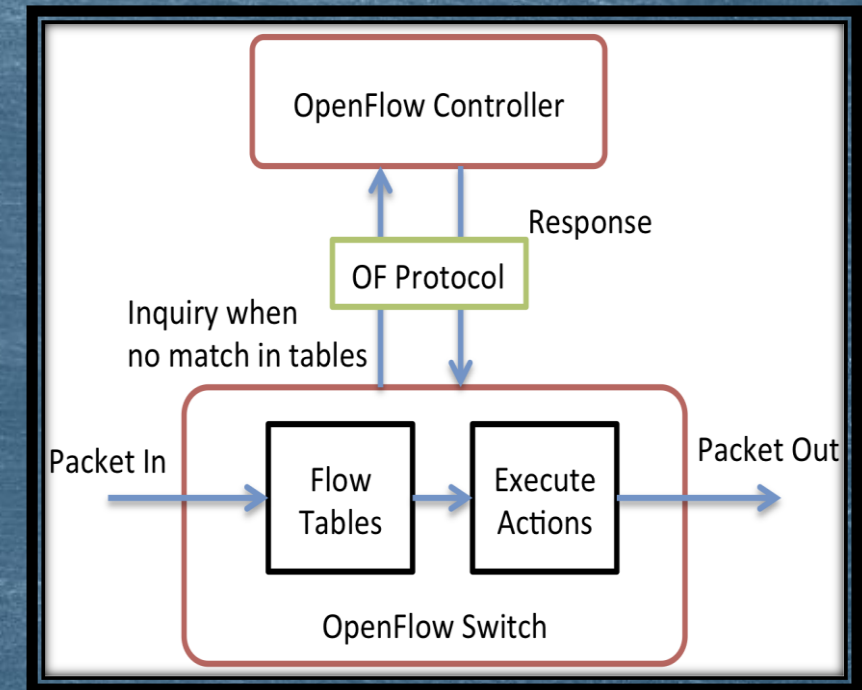
- ▶ **Directly programmable.**
- ▶ **Agile.**
- ▶ **Centrally managed.**
- ▶ **Programmatically configured.**
- ▶ **Open standards-based and vendor-neutral.**
- ▶ **Less Expensive research and Fast Upgrades.**



# OpenFlow Protocol in SDN



- OpenFlow (OF) is considered one of the first software-defined networking (SDN) standards. It originally defined the communication protocol in SDN environments that enables the SDN Controller to directly interact with the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based), so it can better adapt to changing business requirements.





# OpenFlow v1.3

---

- ▶ Refactor capabilities negotiation
- ▶ More flexible table miss support
- ▶ IPv6 Extension Header handling support
- ▶ Per flow meters
- ▶ Per connection event filtering
- ▶ MPLS BoS matching



# Open Daylight: Open Source SDN Platform

---

Open Daylight (ODL) is an Open Source platform to enable networking services and technologies.

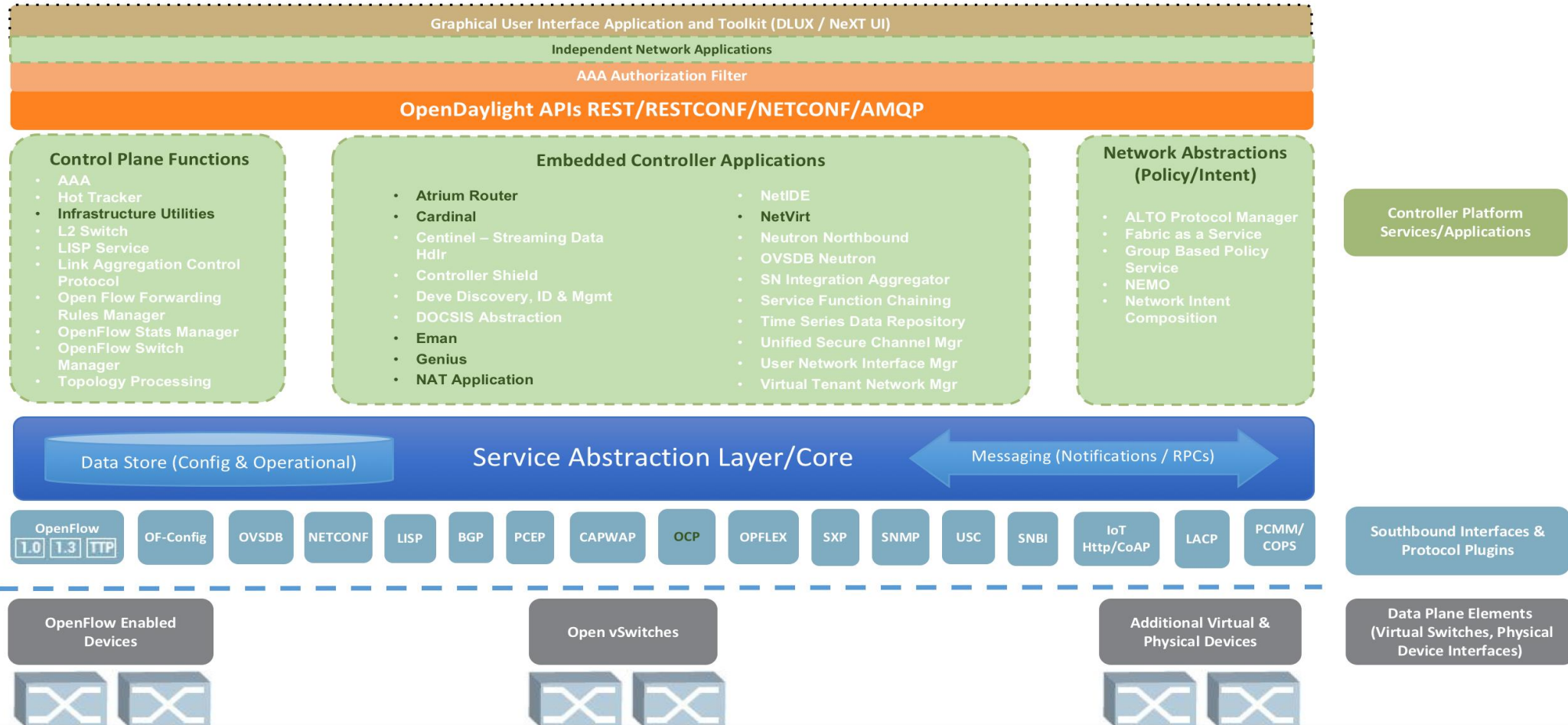
- ▶ **Common SDN toolchains**
- ▶ **Operational tooling**
- ▶ **Application developer tooling**



# ODL Boron Infrastructure



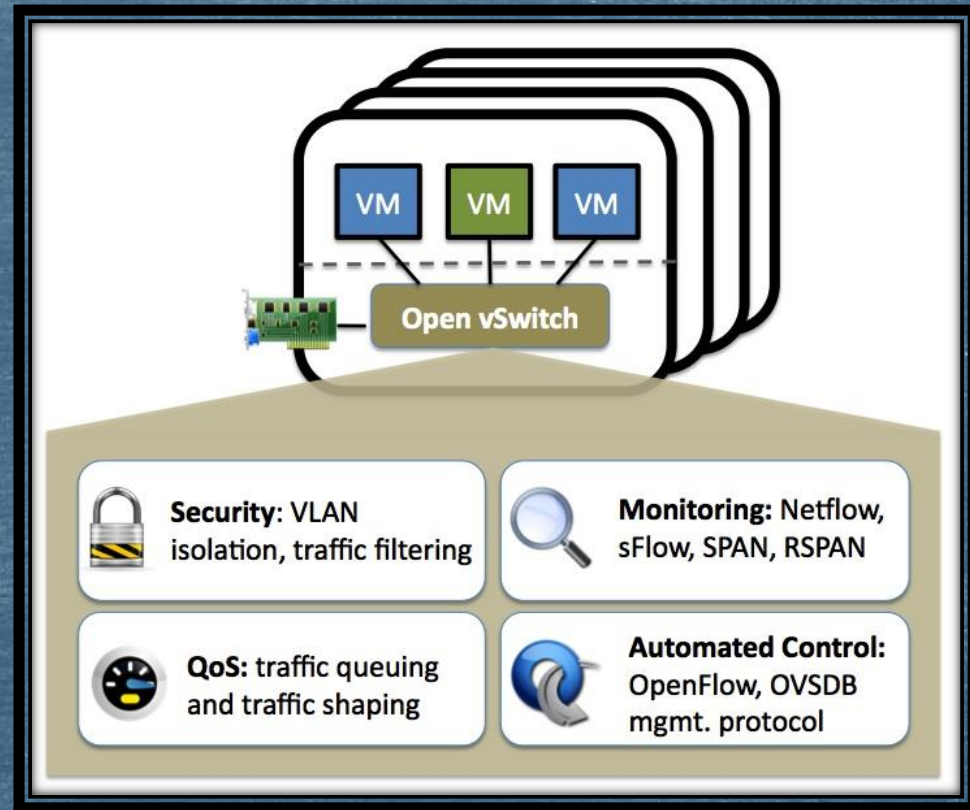
## Boron: Platform for Network-Driven Business





# OpenVSwitch (OvS)

- ▶ Multilayer virtual switch
- ▶ Automation through programmatic extension
- ▶ Supports standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag).
- ▶ Support distribution across multiple physical servers similar to VMware's vNetwork distributed vswitch or Cisco's Nexus 1000V. S





# OvS Features-

---

- ▶ Standard 802.1Q VLAN model with trunking
- ▶ Fine-grained QoS control
- ▶ IPv6 support
- ▶ Per VM interface traffic policing
- ▶ OpenFlow protocol support (including many extensions for virtualization)
- ▶ Remote configuration protocol with C and Python bindings





# Graphical Network Simulator-3

- ▶ The GNS3 network simulator is free, open source software.
- ▶ GNS3 works by using real Cisco IOS images which are emulated using a program called Dynamips.
- ▶ GNS3 is really like the GUI part of the overall product. With this GUI, users get an easy to use interface that allows them to build complex labs consisting of a variety of supported Cisco routers.
- ▶ Emulating the routers using real IOS images is Dynamips



# Features of GNS3

---

GNS3 website lists the following as some of the features provided by the simulator:

- ▶ Design of high quality and complex network topologies
- ▶ Emulation of many Cisco router platforms and PIX firewalls
- ▶ Simulation of simple Ethernet, ATM and Frame Relay switches
- ▶ Connection of the simulated network to the real world
- ▶ Packet capture using Wireshark



# Docker Container

---

- ▶ The Docker support is a new feature of GNS3 1.5. This features has been start by Goran Cetusic during the Google Summer Of Code and finished by the GNS3 core team.
- ▶ Containers use the host kernel this mean they consume, less RAM and CPU. Docker containers are available from a registry, you can fork them in order to add your own tools.





# Summary

---

This presentation concludes with following topics being covered

- ▶ Concept of SDN.
- ▶ OpenFlow Protocol
- ▶ ODL Controller
- ▶ Graphical Network Simulator 3



Thank You !

---