ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Software-Defined Networking Challenges and Future Direction: A Case Study of Implementing SDN Features on OpenStack Private Cloud

Arham Salman Farooqi

Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana, India

ABSTRACT

Software-Defined Networking (SDN) has emerged as a transformative paradigm in the evolution of modern networks by decoupling the control plane from the data plane. This paper explores the integration of SDN within an OpenStack-based private cloud environment. The research identifies the challenges faced during implementation, such as scalability, security, and interoperability, and proposes solutions based on current academic and industrial advancements. A detailed case study of an SDN deployment on an OpenStack cloud is presented, offering insights into architecture, tools, and performance trade-offs. The paper concludes with a discussion on the future direction of SDN technologies in private cloud ecosystems.

KEYWORDS

Software-Defined Networking (SDN), OpenStack, Private Cloud, Network Virtualization, Neutron, OpenFlow, Network Function Virtualization (NFV), Controller, Scalability, Orchestration.

I. INTRODUCTION

Cloud computing has revolutionized IT infrastructure by enabling dynamic scalability, resource optimization, and efficient service delivery. OpenStack, an open-source cloud computing platform, plays a pivotal role in deploying Infrastructure as a Service (IaaS). However, the growing demand for agility and network programmability necessitates a more flexible network management model. Software-Defined Networking (SDN) answers this call by centralizing network control and simplifying configuration through programmable interfaces.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

This paper investigates the integration of SDN into an OpenStack-based private cloud, aiming to improve network automation, reduce complexity, and enhance scalability. It highlights practical challenges, implementation strategies, and performance evaluations in real-world deployments.

II. LITERATURE REVIEW

Numerous studies have explored SDN's capabilities in cloud environments. Major research contributions include:

- **Kreutz et al. (2015)** discussed SDN's architecture and security implications.
- **OpenStack Neutron** was identified as a critical networking component for enabling SDN features via plug-ins (e.g., Open vSwitch, OVN).
- **Hu et al. (2017)** investigated controller placement problems in SDN and suggested optimization models.
- **ONOS and OpenDaylight** were evaluated as open-source SDN controllers integrated with OpenStack in enterprise scenarios.

Despite advances, integration challenges such as limited orchestration between SDN controllers and OpenStack, performance bottlenecks, and vendor-specific implementations remain.

III. METHODOLOGY

This study employed a practical approach to implementing SDN in a private OpenStack cloud environment. The methodology includes:

- **Infrastructure Setup**: An OpenStack cluster with Neutron networking configured using Open vSwitch.
- **SDN Controller**: Integration of OpenDaylight as the external SDN controller.
- **Topology**: A three-node OpenStack deployment (controller, compute, network nodes).
- **Evaluation Metrics**: Network latency, throughput, controller response time, and orchestration effectiveness.
- **Tools Used**: Wireshark, iperf, and OpenStack Horizon dashboard for visualization.

IV. CASE STUDY TABLE: SDN INTEGRATION COMPONENTS

Component	Technology Used	Purpose	Challenges
Network	OpenDaylight	Centralized SDN control	Integration with

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Component	Technology Used	Purpose	Challenges
Controller			Neutron
Virtual Switch	Open vSwitch (OVS)	Packet forwarding and OpenFlow support	Performance at scale
Orchestrator	OpenStack Neutron	Network-as-a-Service	Limited SDN awareness
Monitoring	Wireshark, iperf	Performance and traffic analysis	Overhead in large-scale setups

---

SDN (Software-Defined Networking) Integration Components are the essential building blocks that enable the design, deployment, and operation of SDN-based networks. These components work together to separate the **control plane** from the **data plane**, enabling centralized network control and programmability.

Here's an overview of the **key SDN integration components** and how they interact:

1. SDN Controller (Control Plane)

Role:

The brain of the SDN architecture. It manages the network by maintaining a global view and issuing instructions to network devices (switches, routers) in the data plane.

Functions:

- Network topology discovery
- Flow rule installation
- Policy enforcement
- Path computation and traffic engineering

Examples:

- **OpenDaylight**
- **ONOS (Open Network Operating System)**
- **Ryu**
- **Floodlight**
- **Cisco APIC-EM**

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

□ 2. Southbound Interfaces (SBI)

Role:

Protocols and APIs that enable communication between the **SDN controller** and **network devices** in the data plane.

Common Southbound Protocols:

- **OpenFlow**: Most widely used; allows flow rule definition.
- **NETCONF/YANG**: Used for configuration management.
- **OVSDB**: Used for managing Open vSwitch instances.
- **gNMI/gNOI**: Google's telemetry and operations interface.
- **P4 Runtime**: Supports programmable data planes using the P4 language.

□ 3. Network Devices (Data Plane)

Role:

These are the **forwarding elements** (switches, routers) that execute instructions from the controller, such as forwarding packets based on flow table rules.

Types:

- **OpenFlow switches**
- **Virtual switches (e.g., Open vSwitch)**
- **Programmable hardware (e.g., using P4)**

Functions:

- Forwarding and dropping packets
- Matching packets against flow tables
- Reporting flow stats to controllers

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

□ 4. Northbound Interfaces (NBI)

Role:

APIs and protocols that allow **applications and services** to communicate with the SDN controller.

Purpose:

- Abstract network complexity
- Enable app developers to write logic for traffic shaping, security, or QoS without deep networking knowledge

Common APIs:

- **RESTful APIs**
- **gRPC**
- **GraphQL**

□ 5. SDN Applications (Application Plane)

Role:

Software programs that define network behavior and policies. These run **above the controller** and use NBIs to interact with it.

Examples of SDN Applications:

- **Load Balancers**
- **QoS Managers**
- **Security Monitoring Systems (e.g., IDS/IPS)**
- **Traffic Engineering Tools**
- **Firewall Policy Engines**
- **Network Virtualization (e.g., VXLAN overlay control)**

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

□ 6. Network Operating System (NOS)

Role:

Acts as a platform on which the SDN controller and apps run. It provides essential services like abstraction, state management, and plugin frameworks.

Examples:

- Integrated into controllers like ONOS or OpenDaylight
- Handles distributed state management in clustered controllers

☘ 7. Orchestration and Management Systems

Role:

Used for large-scale deployments to manage virtualized network resources across cloud and edge environments.

Examples:

- **OpenStack Neutron (for SDN integration in cloud)**
- **Kubernetes CNI plugins (e.g., Calico, Cilium)**
- **NFV Orchestrators (e.g., OSM, ONAP)**

□ 8. Security and Policy Frameworks

Role:

Ensures secure interaction among SDN components and enforces network-wide security policies.

Functions:

- Role-based access control (RBAC)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

- Policy definition and enforcement (e.g., intent-based networking)
- Encryption of controller-device communications (e.g., TLS)

□ 9. Telemetry and Monitoring Systems

Role:

Provide real-time visibility into network performance and help in fault detection and analytics.

Tools & Protocols:

- **sFlow**, **NetFlow**, **IPFIX**
- **Prometheus**, **Grafana**
- **OpenConfig Telemetry**
- **SNMP for legacy integration**

□ How These Components Work Together:

1. **Applications** (e.g., firewall rules) communicate with the **SDN controller** via the **Northbound API**.
2. The **controller** processes these policies and translates them into flow rules.
3. Through the **Southbound Interface**, the controller pushes these flow rules to the **network devices**.
4. Devices execute the instructions and report status/stats back to the controller.
5. **Telemetry tools** collect real-time data for monitoring and optimization.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

□ Visual Summary of SDN Integration:

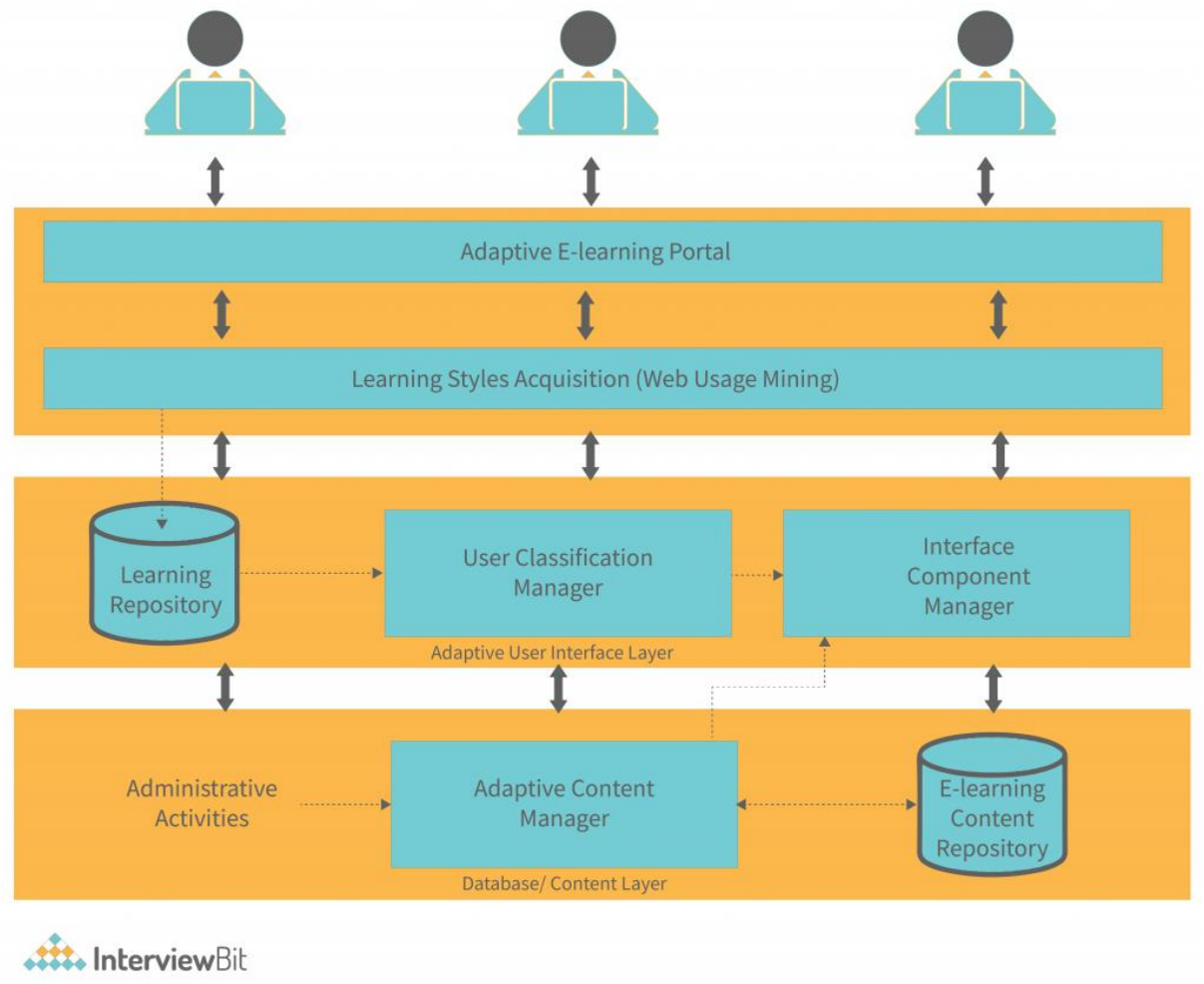


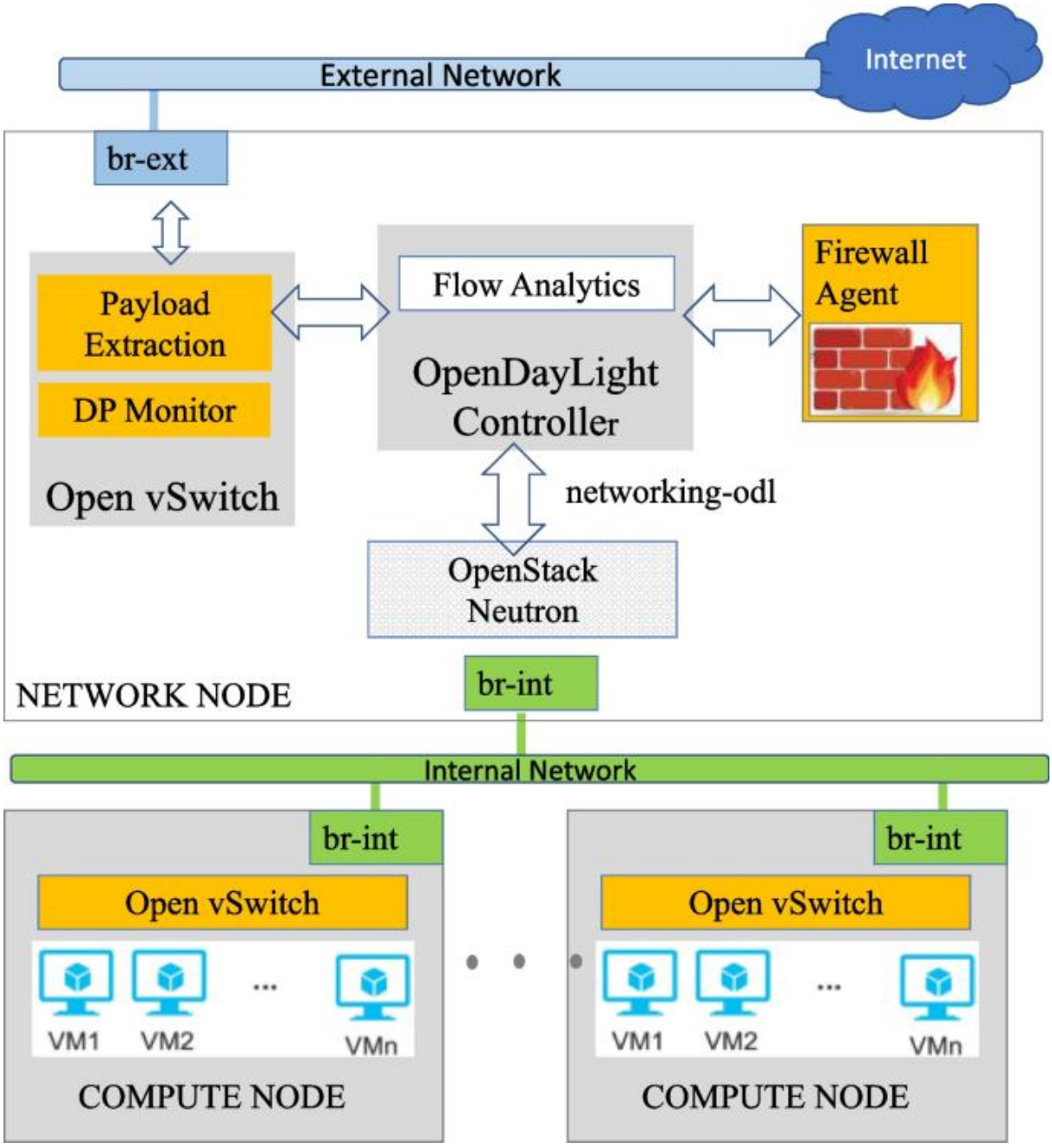
Figure 1: *SDN-enabled OpenStack Architecture*

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015



This architecture showcases how OpenDaylight communicates with Neutron via the ML2 plugin to control Open vSwitches across compute and network nodes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

VI. CONCLUSION

The integration of SDN with OpenStack private clouds brings significant benefits, including dynamic network programmability, reduced manual intervention, and better scalability. However, real-world implementations face hurdles like controller compatibility, security enforcement, and performance overheads. Future work should focus on better orchestration frameworks, AI-driven network optimization, and cross-platform standardization to fully realize SDN's potential in private clouds.

REFERENCES

1. **McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., & Turner, J. (2008).**
OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69–74. https://doi.org/10.1145/1355734.1355746
2. **Open Networking Foundation. (2012).**
Software-Defined Networking: The new norm for networks. *ONF White Paper*. https://opennetworking.org
3. **Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., & Shenker, S. (2008).**
NOX: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3), 105–110. https://doi.org/10.1145/1384609.1384625
4. **Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014).**
A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3), 1617–1634. https://doi.org/10.1109/SURV.2014.012214.00180
5. **Kempf, J., Johansson, O., Pettersson, S., Holler, J., & Martikainen, H. (2012).**
Moving the mobile Evolved Packet Core to the cloud. *Proceedings of the 2012 IEEE International Conference on Cloud Computing (CLOUD)*, 784–791. https://doi.org/10.1109/CLOUD.2012.113
6. **Mohit, Mittal (2013).** The Rise of Software Defined Networking (SDN): A Paradigm Shift in Cloud Data Centers. International Journal of Innovative Research in Science, Engineering and Technology 2 (8):4150-4160.
7. **Blenk, A., Basta, A., Zerwas, J., & Kellerer, W. (2013).**
Survey on network virtualization hypervisors for SDN. *IEEE Communications Surveys & Tutorials*, 18(1), 655–685. https://doi.org/10.1109/COMST.2015.2451851
8. **Secci, S., Tornatore, M., & Pattavina, A. (2012).**
A survey on the role of SDN for optical networking. *IEEE Communications Surveys & Tutorials*, 15(4), 1620–1638. https://doi.org/10.1109/SURV.2013.051313.00034
9. **Lantz, B., Heller, B., & McKeown, N. (2010).**
A network in a laptop: Rapid prototyping for software-defined networks. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets-IX)*, 1–6. https://doi.org/10.1145/1868447.1868466