CS60203: Design Optimization of Computing Systems *November 17, 2024*

Software Defined Networking

Instructor: Sandip Chakraborty *Scribe-By: <Raj Pajesh Parikh >*

1 Summary

Software Defined Networking (SDN) is an innovative approach to network management that enables dynamic, programmatically efficient network configuration to improve network performance and monitoring. SDN decouples the network control and forwarding functions, allowing for direct programmability of the network control, and abstracting the underlying infrastructure for applications and network services. This architecture is ideal for the dynamic and high-bandwidth nature of today's applications, providing a flexible and efficient way to manage complex networks.

2 The Networking Stack

Traditional networking architectures comprise multiple planes, each handling different aspects of network operations.

2.1 Data Plane

The **Data Plane** handles all activities involving data packets transmitted by end-users. Its primary functions include:

- **Packet Forwarding**: Directing packets from their source to the appropriate destination.
- **Segmentation and Reassembly**: Dividing large packets for transmission and reassembling them at the destination.
- **Packet Replication for Multicasting**: Creating copies of packets to send to multiple recipients simultaneously.

These activities must operate at high speeds (line rate), necessitating specialized hardware like TCAMs. Some data plane tasks may involve CPU processing, particularly those related to broadcast services.

2.2 Control Plane

The **Control Plane** encompasses all activities necessary to support data plane functions but does not directly involve handling user data packets. It serves as the network's decision-making entity, responsible for:

- **Routing Table Construction**: Determining optimal paths for packet forwarding.
- **Policy Enforcement**: Ensuring compliance with network policies for packet handling.

- **Service Availability Signaling**: Indicating the availability of network services.

In conventional networks, the control plane is integrated within each network device, leading to decentralized control logic and increased complexity.

2.3 Management Plane

The **Management Plane** focuses on monitoring and managing the network's overall health and performance. Key activities include:

- **Fault Management**: Detecting and resolving network issues.
- **Configuration Management**: Setting up devices and network configurations.
- **Accounting Management**: Tracking resource usage for billing or analysis.
- **Performance Management**: Monitoring and optimizing network performance metrics.
- **Security Management**: Implementing security policies and responding to threats.

2.4 Service Plane

The **Service Plane** provides additional services to enhance network functionality, scalability, and security, such as:

- **Firewalls**: Controlling network access based on security policies.
- **Proxies**: Facilitating indirect network connections on behalf of clients.
- **Load Balancers**: Distributing network traffic across multiple servers.
- **Intrusion Detection Systems (IDS)**: Monitoring network traffic for malicious activities.

2.5 Data vs. Control Logic in the Networking Stack

In traditional networking devices:

- The **Data Plane** requires high-speed processing capabilities to handle traffic at line rate.
- The **Control Plane** logic is executed by the device's CPU.
- Both planes are tightly integrated within the device, leading to interdependent functionalities.

This integration results in increased capital (CAPEX) and operational expenditures (OPEX) due to the complexity and cost of advanced networking hardware. It also complicates network management and scalability.

3 Key Idea of SDN

SDN introduces a paradigm shift by separating the control plane from the data plane:

- **Control Plane**: Centralized in a software-based SDN controller, which makes decisions about traffic routing and policy enforcement.
- **Data Plane**: Remains in the network devices (switches), responsible for forwarding packets based on the controller's instructions.

This separation allows for centralized network management, programmability, and greater flexibility in configuring network behavior without modifying individual devices.

4 Control Logic

The SDN controller acts as the network's brain, providing:

- **Global Network View**: Maintains an overall perspective of the network's state.
- **Centralized Decision-Making**: Determines how traffic should be managed across the network.
- **Interface for Applications**: Allows network applications to request network services and configurations.

By centralizing control logic, SDN simplifies network management and enables more consistent policy implementation.

5 SDN Architecture

In SDN, the architecture is organized into three distinct layers, each with specific roles and responsibilities.

5.1 Infrastructure Layer

Also known as the **Data Plane**, this layer consists of physical and virtual switches that forward data packets based on flow tables provided by the SDN controller.

5.2 Control Layer

This layer houses the **SDN Controller**, which serves as the central control point for the network. It communicates with the infrastructure layer to manage traffic flow and with the application layer to receive network service requests.

5.3 Application Layer

The **Application Layer** includes network applications and services such as intrusion detection systems, firewalls, and load balancers. These applications interact with the SDN controller through APIs to define desired network behaviors.

5.4 Communication Interfaces

Communication between these layers occurs via:

- **Northbound APIs**: Interfaces between the Application Layer and the Control Layer, allowing applications to communicate network requirements to the controller.

- **Southbound APIs**: Interfaces between the Control Layer and the Infrastructure Layer, enabling the controller to instruct switches on how to handle traffic. OpenFlow is a common southbound API.

5.5 Flow Tables and Packet Processing

When a packet arrives at a switch in the infrastructure layer:

1. The switch consults its **Flow Table**, which contains entries with **match fields** and corresponding **instructions**.
2. If a match is found, the switch executes the specified instructions, such as forwarding the packet to a specific port, dropping it, or modifying headers.
3. If no match is found, the switch queries the SDN controller for guidance.
4. The controller may install a new flow entry in the switch's flow table, dictating how similar packets should be handled in the future.

This mechanism allows for dynamic and flexible network traffic management.

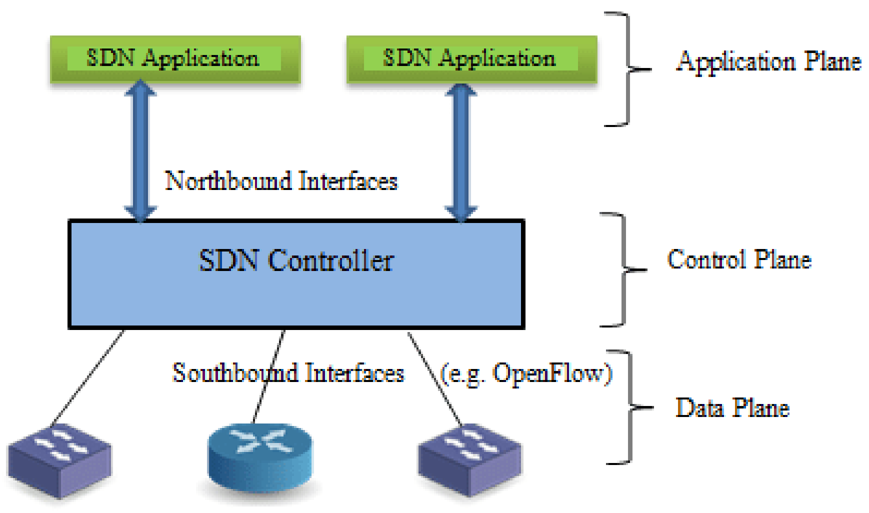

Figure 1: Software Defined Networking Architecture[1]

6 OpenFlow [2]

OpenFlow is a foundational protocol in Software Defined Networking (SDN) that facilitates seamless communication between the SDN controller and the network devices within the infrastructure layer. By standardizing the interaction between the control and data planes, OpenFlow enables dynamic and flexible network management.

6.1 Functionality of OpenFlow

OpenFlow empowers the SDN controller to:

- **Access and Modify Flow Tables**: The controller can read, add, update, or remove flow entries in the switches' flow tables.
- **Install Flow Entries**: Define how packets matching specific criteria should be handled by the switch.
- **Collect Statistics**: Gather data on network traffic and flow usage for monitoring and optimization purposes.

6.2 Flow Tables

At the heart of OpenFlow-enabled switches are **Flow Tables**, which determine how incoming packets are processed and forwarded. Each flow table entry is composed of four essential components:

1. **Rule**:
   - Defines the criteria for matching incoming packets.
   - Typically includes packet header fields such as source and destination IP addresses, port numbers, VLAN tags, and other protocol-specific information.
2. **Action**:
   - Specifies the operations to be performed on packets that match the rule.
   - Actions can include forwarding the packet to a specific port, dropping the packet, modifying packet headers, or sending the packet to the controller for further processing.
3. **Statistics**:
   - Keeps track of metrics related to the flow, such as the number of packets and bytes that have matched the rule.
   - Helps in monitoring network performance and understanding traffic patterns.
4. **Priority**:
   - Determines the order in which flow entries are evaluated.
   - Higher priority rules are matched before lower priority ones, allowing for more specific rules to take precedence over general ones.

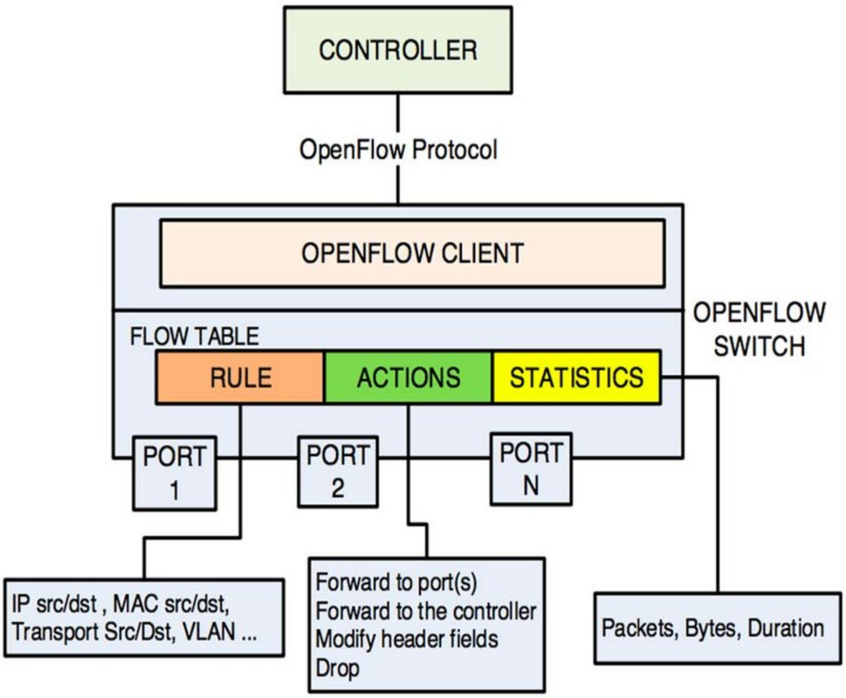

Figure 2: Communications via OpenFlow Protocol[3]

6.3 Packet Handling Process

The process of handling packets in an OpenFlow-enabled SDN environment involves the following steps:

1. **Packet Arrival**: An incoming packet reaches an OpenFlow switch.
2. **Flow Table Lookup**: The switch examines its flow tables to find a flow entry whose **Rule** matches the packet's header fields.
3. **Action Execution**:
   - If a matching flow entry is found, the switch performs the specified **Action(s)** on the packet.
   - If multiple flow entries match, the entry with the highest **Priority** is selected.
4. **Packet Forwarding or Dropping**: Based on the **Action**, the packet is either forwarded to the designated port, dropped, or sent to the SDN controller for further instructions.
5. **Controller Interaction**:
   - If no matching flow entry is found, the switch forwards the packet to the SDN controller.
   - The controller then decides how to handle the packet, which may involve installing a new flow entry into the switch's flow table.
6. **Flow Entry Installation**: The controller sends a new flow entry containing the appropriate **Rule**, **Action**, **Statistics**, and **Priority** to the switch, ensuring that future packets matching the rule are handled efficiently.

7 Advantages and Disadvantages of SDN [4]

7.1 Advantages

- **Programmability**: Networks can be easily modified and configured through software without manual intervention on individual devices.
- **Cost Reduction**: Switch hardware becomes less expensive since it only requires data plane capabilities.
- **Hardware Abstraction**: Applications can be developed independently of the underlying hardware, promoting vendor neutrality.
- **Enhanced Security**: The controller can monitor network traffic and enforce security policies, such as rerouting or dropping suspicious packets.

7.2 Disadvantages

- **Centralized Point of Failure**: Reliance on a central controller introduces a single point of failure, which can impact the entire network if compromised.
- **Scalability Challenges**: Deploying SDN on a large scale requires careful planning and is still an area of ongoing research.

8 Applications and Importance of SDN [5]

8.1 Use Cases of SDN

SDN is utilized in various contexts:

- **Enterprise Networks**: Facilitates faster application deployment and reduces deployment and operational costs.
- **Cloud Data Centers**: Enhances resource utilization and lowers capital and operational expenditures.
- **Service Providers**: Enables flexible and customizable network services for customers.

8.2 Benefits of SDN

- **Improved Connectivity**: Enhances network connectivity for communication and data sharing.
- **Accelerated Deployment**: Speeds up the deployment of new applications and services.
- **Better Security**: Offers enhanced visibility and control over network security policies.
- **Increased Control and Speed**: Enables high-speed network management through centralized, software-based controllers.

References

[1] Performance Evaluation of Openflow SDN Controllers. https://www.researchgate.net/publication/323925422_Performance_Evaluation_of_Openflow_SDN_Controllers.

[2] Paul Goransson and Chuck Black. *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann, 2014.

[3] Review on Detection Techniques against DDoS Attacks on a Software-Defined Networking Controller. https://www.researchgate.net/publication/319886437_Review_on_Detection_Techniques_against_DDoS_Attacks_on_a_Software-Defined_Networking_Controller.

[4] GeeksforGeeks. https://www.geeksforgeeks.org/software-defined-networking/.

[5] What is Software Defined Networking (SDN)? https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-ofsoftware-defined-networking-sdn/.