OpenFlow

Architecture & Usage

ICD Course, Dr. Foshati, Fall 2024

Topics covered

- Introduction
- Basics & Architecture
- Implementation & Communication with Controller
- Usages, Advantages & Challenges
- Conclusion & Future of OpenFlow

Introduction to OpenFlow

- Traditional networks
  - **static configurations** & **decentralized control**
  - rigid and difficult to manage
- **Software-Defined Networking (SDN)**
  - separating the control plane from the data plane
- **OpenFlow**
  - first and most widely adopted SDN protocol
  - direct control of network switches via a centralized controller
  - **flexibility, programmability, and dynamic traffic management**

Basic concept & Architecture

Challenges of Traditional Networks

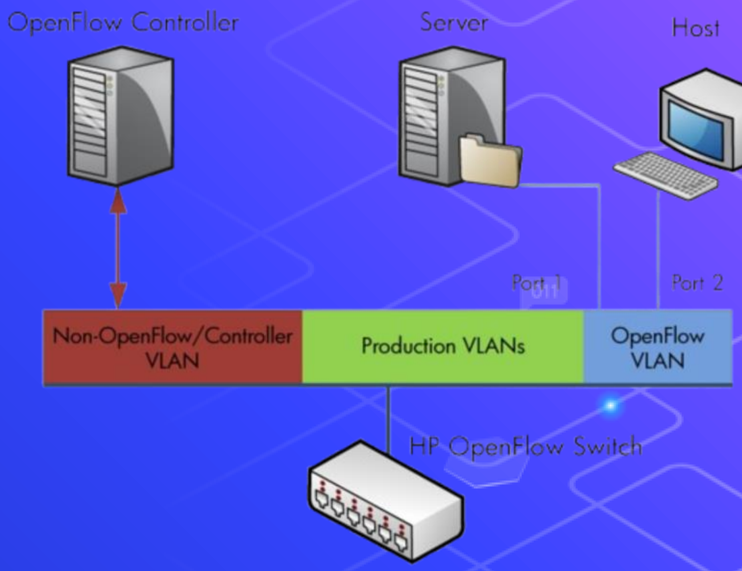
- **Decentralized Control:** independent switch/routers leading to complexity.
- **Static Configuration:** manually defined rules & limiting flexibility
- **Difficult Traffic Management:** complex configurations for Optimizing network performance
- **Slow Adaptation to Change:** Modifying network policies is time-consuming and error-prone

What is OpenFlow?

- direct communication between **network devices** and a **centralized SDN controller**
- introduced by **Stanford University (2008)**
- standardized by **Open Networking Foundation (ONF)**
- dynamic **control and modification of network traffic flows** without changing physical hardware
- **datacenters, enterprise networks, cloud providers, and research labs**

OpenFlow Architecture

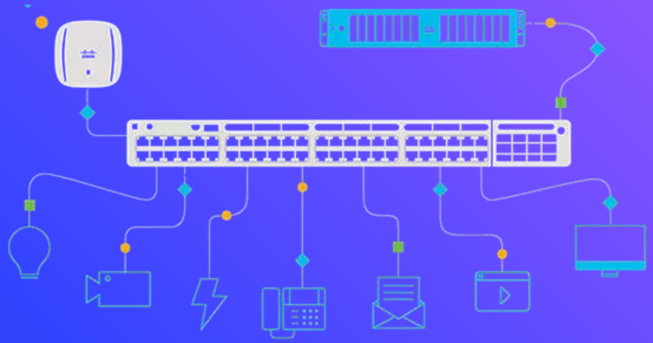
- **SDN Controller:**
  - Centralized intelligence
  - decides how packets are forwarded.
  - Examples: ONOS, Ryu, Floodlight.
- **OpenFlow Switch:**
  - Implements the rules provided by the controller
  - Contains **Flow Tables** to manage traffic forwarding.
- **OpenFlow Protocol:**
  - Standardized communication channel between the controller and switches.



OpenFlow Packet Processing

1. **A new packet arrives at the switch.**
2. **Switch checks its Flow Table:**
   - If a matching rule exists → **Apply the action (forward, drop, modify, etc.).**
   - If no match → **Send the packet to the controller (Packet-In message).**
3. **Controller makes a decision:**
   - Installs a new rule in the switch (Flow-Mod message).
   - Sends a direct action (Packet-Out message).
4. **Switch follows the new instructions and processes future packets accordingly.**

Key Components of Switch



1. **Flow Table**
   - Stores rules for packet forwarding.
   - Matches fields like IP, MAC, VLAN, and TCP/UDP ports.
2. **Pipeline Processing**
   - Multiple flow tables can process packets in stages.
3. **Secure Channel**
   - Establishes a **TLS or TCP connection** between the switch and the controller.
4. **OpenFlow Protocol**
   - Defines the messages used for communication.

Flow Tables in OpenFlow

- **Match Fields:** Identifies which packets the rule applies to (IP, MAC, Port, VLAN)
- **Priority:** rule precedence if multiple matches exist
- **Actions:** what to do with the matched packets (Forward, Drop, Modify, etc.)
- **Counters:** Tracks statistics (e.g., number of packets matched)
- **Timeouts:** Rules can be removed after a certain time

Controller’s Decision Process

- **A. Rule Matching**
  - **policy or rule** for this type of traffic
  - matches **packet headers** (e.g., MAC, IP, TCP/UDP ports) with **predefined flow rules**.
- **B. Dynamic Traffic Analysis**
  - **network monitoring** to check traffic patterns, congestion, or security threats
  - if a link is overloaded, the controller might reroute traffic
- **C. Custom Applications & Algorithms**
  - **custom SDN applications** for load balancing, security, or quality of service (QoS)

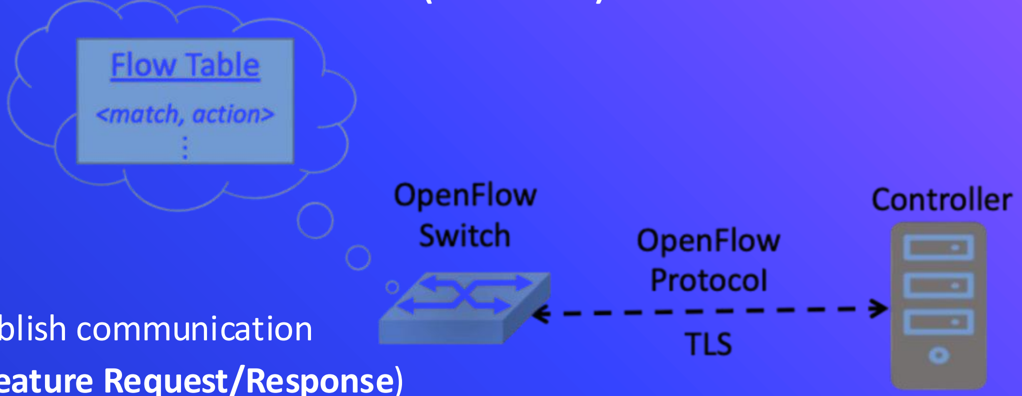
implementation & communication with the controller

Types of OpenFlow Messages

- **Controller-to-Switch Messages**
  - **Feature Request/Response** – Controller queries switch capabilities.
  - **Flow-Mod** – Controller modifies switch Flow Table.
  - **Packet-Out** – Controller instructs switch to forward a packet.
- **Asynchronous Messages**
  - **Packet-In** – Switch sends a new (unknown) packet to the controller.
  - **Flow-Removed** – Informs controller when a flow entry expires.
  - **Port Status** – Notifies controller of port changes (up/down).
- **Symmetric Messages**
  - **Hello** – Establishes connection between switch and controller.
  - **Echo Request/Reply** – Checks connectivity and latency.

OpenFlow Control Communication Flow

- OpenFlow switches connect to an **SDN Controller** via a **Secure Channel (TLS or TCP)**
- The controller
  - monitors the network
  - modifies traffic rules
  - dynamically installs new policies.
- The communication follows these steps:
  - switch **sends a "Hello" message** to establish communication
  - controller queries switch capabilities (**Feature Request/Response**)
  - Switch forwards unknown packets to the controller (**Packet-In messages**)
  - The controller **installs new rules** and updates the switch (**Flow-Mod messages**)
  - The switch follows these rules to process future packets



OpenFlow vs. Traditional

Feature	Traditional Networks	OpenFlow (SDN)
Control Plane	Distributed	Centralized
Traffic Management	Static, manually configured	Dynamic, programmable
Adaptability	Limited (hardware-dependent)	High (software-based control)
Complexity	High (manual updates, multiple protocols)	Lower (centralized updates, unified protocol)
Network Efficiency	Lower (fixed routing, inefficient use of resources)	Higher (optimized, software-defined traffic control)

Applications, Benefits, Limitations, Use cases

Benefits of OpenFlow

- **Centralized Network Control**
  - network visibility
  - simplify management
- **Dynamic Traffic Management**
  - real-time network adjustments and optimizations
- **Reduced Hardware Dependency**
  - Works with commodity switches, lowering costs
- **Faster Network Innovation**
  - rapid deployment of new networking protocols and services
- **Enhanced Security & Automation**
  - programmable security policies
  - automatic responses to threats

Challenges & Limitations

- **Scalability Issues**
  - performance bottlenecks in Large-scale networks
- **Complex Deployment**
  - Transitioning from traditional networking requires new tools and expertise
- **Security Concerns**
  - The **controller could becomes a single point of failure**
- **Limited Hardware Support**
  - Not all switches support OpenFlow

Challenges

Applications of OpenFlow

- **Data Centers**
  - Automates network management and optimizes resource allocation.
  - Enables dynamic traffic engineering for cloud providers.
- **Enterprise Networks**
  - Simplifies network policy enforcement and security.
  - Reduces hardware dependency and operational costs.
- **Telecom & 5G Networks**
  - Enables software-defined **5G network slicing** and traffic optimization.
  - Supports dynamic Quality of Service (QoS) adjustments.
- **Research & Academia**
  - Used in network simulations and SDN testbeds.
  - Powers experimental network architectures.

Real-World Deployments

- **Google B4 Network**
  - Google's **global SDN backbone** for inter-data center traffic
  - **efficient bandwidth utilization** and **traffic engineering**
- **Facebook's Data Centers**
  - OpenFlow-based SDN to optimize large-scale cloud infrastructure
- **AT&T SDN Strategy**
  - OpenFlow for **dynamic traffic routing** and **5G network management**
- **GENI & Internet2 Research Networks**
  - OpenFlow-based experimental platforms for next-gen networking research

Conclusion & Future OF OpenFlow

The Future of OpenFlow & SDN

- **Integration with AI & Machine Learning**
  - AI-driven network automation for predictive traffic management and security
- **Enhanced Security Mechanisms**
  - More robust encryption and anomaly detection for protecting the SDN controller.
- **Support for 5G & Edge Computing**
  - OpenFlow helps optimize **low-latency networks** for **IoT, 5G, and cloud-edge architectures**.
- **Shift Towards P4 & Intent-Based Networking**
  - OpenFlow is evolving, and new technologies like **P4 (Programmable Data Plane)** are emerging.

Conclusion

- **OpenFlow revolutionized networking** by enabling SDN and centralized traffic control.
- It provides **flexibility, automation, and cost efficiency**, making networks smarter.
- OpenFlow is used in **data centers, 5G, cloud computing, and research**.
- **Challenges remain** in scalability, security, and hardware adoption.
- The future of OpenFlow includes **AI-driven networking, P4, etc**.

Questions

References

- [*Open Networking Foundation (ONF)*]
- [*Google's B4 SDN Paper*]
- [*OpenFlow Specification (ONF Docs)*]
- [*WIKIPEDIA*]
- *Software Defined Networking: Design and Deployment* – Jim Doherty
- *SDN: Software Defined Networks* – Thomas D. Nadeau & Ken Gray

Thanks for Attention

Yousef Sadidi
401170597