CSIT985
Strategic Network Design

Autumn 2024

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Lecture 12:

Advanced Networking Technologies

Presented by: Dr. Shengbing Tang

Lecturer, CCNU-UOW Joint Institute

UNIVERSITY OF WOLLONGONG AUSTRALIA

Overview

- SDN
- SDN and Big data
- NFV
- Machine Learning

Software-Defined Networking (SDN)

Before SDN – Challenges to traditional networks

- High demand on massive data transmission when the flow increases unexpectedly
- Inflexible network structure
- Issues: network sustainable development, network uncontrollability, and network security
- Implementation of new technologies leads to the complexity and cost concerns

Before SDN – Challenges to traditional networks

1. Scalability

- **Issue:** Traditional networks struggle to handle the exponential growth of connected devices and data traffic.
- **Cause:** Rigid architecture and limited automation capabilities.
- **Impact:** Performance degradation and increased costs.

2. Flexibility and Agility

- **Issue:** Traditional networks often require significant manual configuration and lack dynamic adaptability.
- **Cause:** Static hardware-centric design.
- **Impact:** Slow response to changing business requirements or network demands.

3. Performance Bottlenecks

- **Issue:** High latency and low throughput in handling real-time data.
- **Cause:** Centralized control and older switching/routing hardware.
- **Impact:** Poor user experience and inefficient operations.

Before SDN – Challenges to traditional networks

4. Security

- **Issue:** Increased vulnerability to sophisticated cyber threats.
- **Cause:** Limited ability to integrate modern security measures like zero-trust architecture.
- **Impact:** Higher risk of data breaches and attacks.

5. Integration with Emerging Technologies

- **Issue:** Difficulty in supporting AI, IoT (Internet of Things), edge computing, and 5G technologies.
- **Cause:** Lack of compatibility with advanced protocols and cloud-based architectures.
- **Impact:** Limited capability to leverage new innovations.

6. Operational Complexity

- **Issue:** High dependency on manual operations for configuration, troubleshooting, and maintenance.
- **Cause:** Lack of automation and centralized management.
- **Impact:** Increased operational costs and time inefficiencies.

Before SDN – Challenges to traditional networks

7. Cost-Efficiency

- **Issue:** Higher capital and operational expenses.
- **Cause:** Proprietary hardware, extensive cabling, and lack of virtualized resources.
- **Impact:** Limited feasibility for cost-sensitive businesses.

8. Adaptability to Dynamic Traffic Patterns

- **Issue:** Poor performance under fluctuating or unexpected traffic loads.
- **Cause:** Stiff hierarchical designs without dynamic resource allocation.
- **Impact:** Inefficient utilization and network outages.

9. Energy Consumption

- **Issue:** High energy usage and environmental impact.
- **Cause:** Inefficient hardware and lack of intelligent energy-saving features.
- **Impact:** Increased operational costs and carbon footprint.

Before SDN – Challenges to traditional networks

10. Global Reach and Mobility

- **Issue:** Challenges in providing seamless global connectivity and mobile access.
- **Cause:** Geographically fixed infrastructure and lack of seamless handover protocols.
- **Impact:** Inconsistent user experiences, especially for mobile or global users.

Before SDN – Challenges to traditional networks

1. Scalability

Example: E-commerce During Peak Times

- **Scenario:** A traditional retail website experiences a surge in traffic during Black Friday.
- **Problem:** The network infrastructure, built on fixed hardware, cannot dynamically scale to accommodate the increased load.
- **Result:** Website crashes, slow loading times, and lost sales opportunities.
- **Challenge Cause:** Static hardware configurations and lack of elastic scalability like cloud-based solutions.

Before SDN – Challenges to traditional networks

2. Flexibility and Agility
Example: Remote Work Transition During COVID-19

- **Scenario:** Companies rapidly transitioned to remote work during the pandemic.
- **Problem:** Traditional networks couldn’t quickly adapt to support a distributed workforce accessing corporate resources from home.
- **Result:** Significant delays in enabling VPN access, overloaded gateways, and poor remote user experience.
- **Challenge Cause:** Manual reconfiguration and reliance on physical appliances like firewalls.

Before SDN – Challenges to traditional networks

3. Performance Bottlenecks
Example: Online Gaming Latency
•**Scenario:** A gaming company launches a popular multi-player game but players experience lag due to high latency.
•**Problem:** Centralized data centers introduce delays for globally distributed users.
•**Result:** Frustrated users abandon the game, damaging the company’s reputation.
•**Challenge Cause:** Lack of edge computing capabilities to process data closer to the users.

Before SDN – Challenges to traditional networks

4. Security

Example: Ransomware Attack on a Hospital Network

- **Scenario:** A hospital relies on a traditional network for operations and patient data storage.
- **Problem:** The network cannot implement real-time threat detection or zero-trust policies. Attackers exploit vulnerabilities to deploy ransomware.
- **Result:** Medical operations are disrupted, sensitive data is exposed, and recovery costs are immense.
- **Challenge Cause:** Outdated firewalls and insufficient monitoring tools.

Before SDN – Challenges to traditional networks

5. Integration with Emerging Technologies
Example: IoT in Smart Cities

- **Scenario:** A city installs IoT sensors for smart traffic management.
- **Problem:** The traditional network infrastructure cannot handle the sheer volume of data generated by the sensors.
- **Result:** Delays in traffic updates and inefficient system performance.
- **Challenge Cause:** Limited capacity and inability to integrate with modern IoT protocols.

Before SDN – Challenges to traditional networks

6. Operational Complexity
Example: Manual Troubleshooting in Data Centers

- **Scenario:** A data center faces an outage caused by a misconfigured switch.
- **Problem:** The network lacks centralized monitoring tools, and IT staff spend hours manually diagnosing the issue.
- **Result:** Prolonged downtime and customer dissatisfaction.
- **Challenge Cause:** Dependence on manual intervention rather than automated tools.

Before SDN – Challenges to traditional networks

7. Cost-Efficiency
Example: Expensive Hardware Upgrades

•**Scenario:** A traditional enterprise network needs to upgrade its routing hardware to meet new demands.

•**Problem:** The upgrades require purchasing proprietary equipment and extensive installation processes.

•**Result:** High upfront costs and increased downtime during the transition.

•**Challenge Cause:** Vendor lock-in and reliance on expensive physical hardware.

Before SDN – Challenges to traditional networks

8. Adaptability to Dynamic Traffic Patterns
Example: Content Delivery During Live Streaming

- **Scenario:** A streaming platform hosts a live sports event.
- **Problem:** The traditional network experiences congestion as millions of users tune in simultaneously.
- **Result:** Buffering and disrupted service, leading to user dissatisfaction.
- **Challenge Cause:** Inability to dynamically allocate bandwidth in real time.

Before SDN – Challenges to traditional networks

9. Energy Consumption
Example: Data Centers' Carbon Footprint

•**Scenario:** A traditional data center struggles with high energy bills due to inefficient cooling systems and network equipment.
•**Problem:** Old hardware operates continuously at peak power, regardless of actual usage.
•**Result:** Increased operational costs and a large carbon footprint.
•**Challenge Cause:** Lack of intelligent energy-saving mechanisms.

Before SDN – Challenges to traditional networks

10. Global Reach and Mobility
Example: International Business Expansion

- **Scenario:** A multinational company sets up new branches in Asia but struggles to connect them to its existing network in Europe.
- **Problem:** The traditional network relies on geographically fixed data centers and lacks robust global connectivity solutions.
- **Result:** Delays in operations and increased costs for setting up international infrastructure.
- **Challenge Cause:** Limited ability to leverage cloud-native, geographically distributed networks.

Before SDN – Challenges to traditional networks

These challenges are increasingly being addressed through technologies like:

① Software-Defined Networking (SDN) for programmability and automation
② Network Function Virtualization (NFV) for flexibility and scalability
③ Cloud-Native Architectures for agility and global reach
④ AI and Machine Learning for intelligent decision-making and security
⑤ 5G and Beyond for high-speed, low-latency connectivity

What is SDN?

- **Software defined networking (SDN)** is an approach to network management that enables dynamic, programmatically efficient network configuration to improve network performance and monitoring. It is a new way of managing computer networks that makes them easier and more flexible to control.

- In traditional networks, the hardware (like routers and switches) decides how data moves through the network, but SDN changes this by moving the decision-making to a central software system. This is done by separating the **control plane** (which decides where traffic is sent) from the **data plane** (which moves packets to the selected destination).

What is SDN?

- SDN stands for Software-Defined Networking
- It is proposed in 2006 by a research group at Stanford University.
- The purpose of SDN is to redesign the Internet
- **OpenFlow protocol:** the core technology of SDN

Components of Software Defining Networking (SDN)

① **SDN Applications**: SDN Applications relay requests or networks through SDN Controller using API.

② **SDN Controller**: SDN Controller collects network information from hardware and sends this information to applications.

③ **SDN Networking Devices**: SDN Network devices help in forwarding and data processing tasks

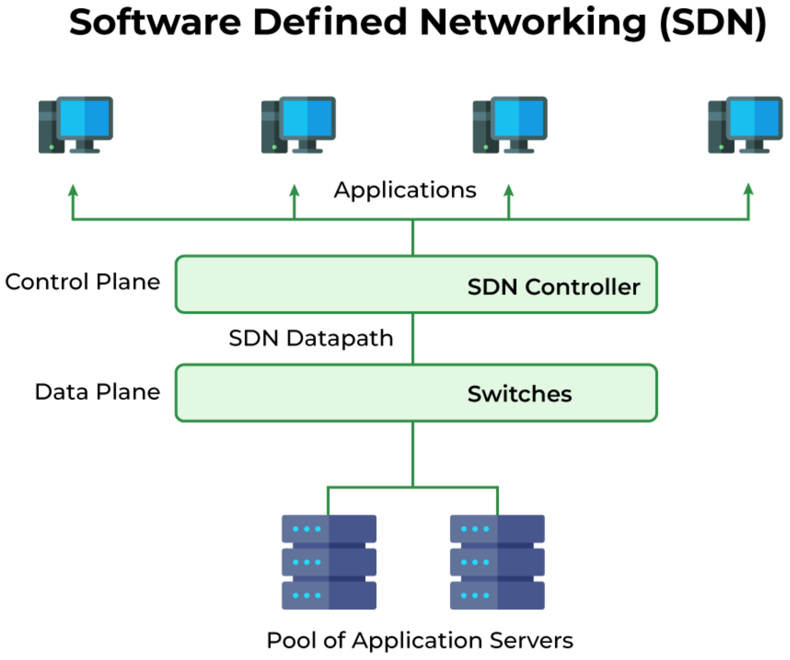
Components of Software Defining Networking (SDN)

What is a Data Plane?

- All the activities involving and resulting from data packets sent by the end-user belong to this plane.

Data Plane includes:

- Forwarding of packets.
- Segmentation and reassembly of data.
- Replication of packets for multicasting.



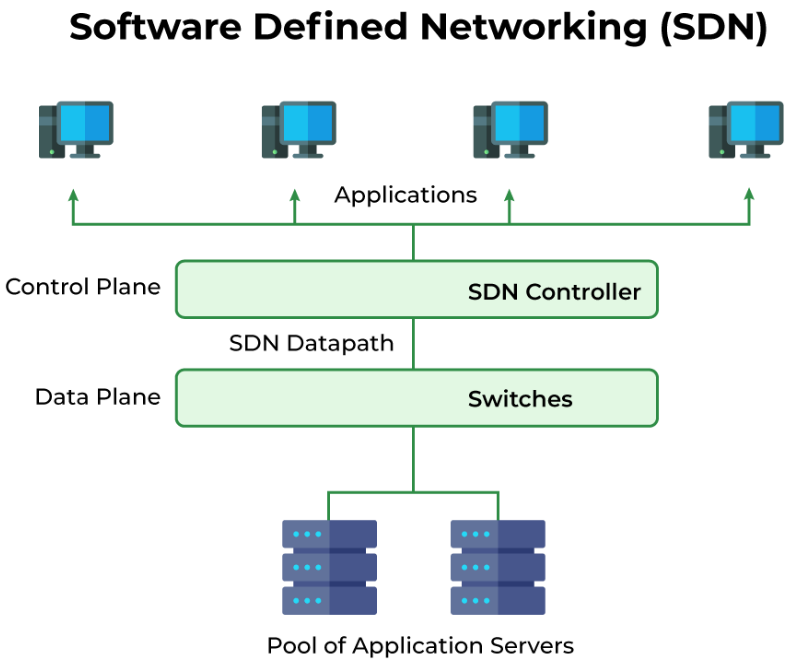
Components of Software Defining Networking (SDN)

What is a Control Plane?

- All activities necessary to perform data plane activities but do not involve end-user data packets belong to this plane. In other words, this is the **brain** of the network.

The activities of the control plane include:

- Making routing tables.
- Setting packet handling policies.



Usages of Software Defining Networking (SDN)

Where is Software Defined Networking Used?

- Enterprises use SDN, the most widely used method for application deployment, to deploy applications faster while lowering overall deployment and operating costs. SDN allows IT administrators to manage and provision network services from a single location.

- Cloud networking software-defined uses white-box systems. Cloud providers often use generic hardware so that the Cloud data center can be changed and the cost of CAPEX and OPEX saved.

Why Software Defined Networking is Important

① **Better Network Connectivity:** SDN provides very better network connectivity for sales, services, and internal communications. SDN also helps in faster data sharing.

② **Better Deployment of Applications:** Deployment of new applications, services, and many business models can be speed up using Software Defined Networking.

③ **Better Security:** Software-defined network provides better visibility throughout the network. Operators can create separate zones for devices that require different levels of security. SDN networks give more freedom to operators.

④ **Better Control With High Speed:** Software-defined networking provides better speed than other networking types by applying an open standard software-based controller.

Why SDN?

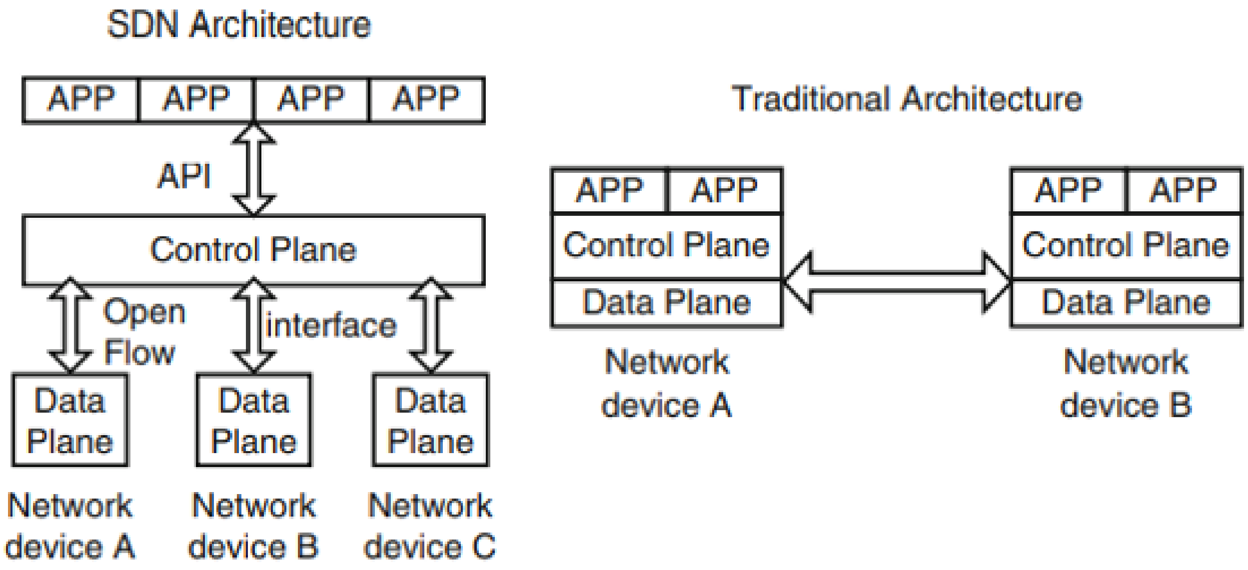


Fig. SDN vs. traditional network architecture (4)

Different Models of SDN

There are several models, which are used in SDN:

① Open SDN

② SDN via APIs

③ SDN via Hypervisor-based Overlay Network

④ Hybrid SDN

Fig. SDN vs. traditional network architecture (4)

SDN Features

- The separation of the control plane and the data plane, allowing them to evolve independently and leaving networking devices simply to forward data efficiently.
- The centralization of network control at a controller external from the network device (the SDN controller or a Network Operating System (NOS)).
- The network programmability via software applications running at the control or application planes.
- The use of flow-based forwarding rules instead of destination-based decisions.

SDN Architecture

3 main planes: application plane, control plane, data plane

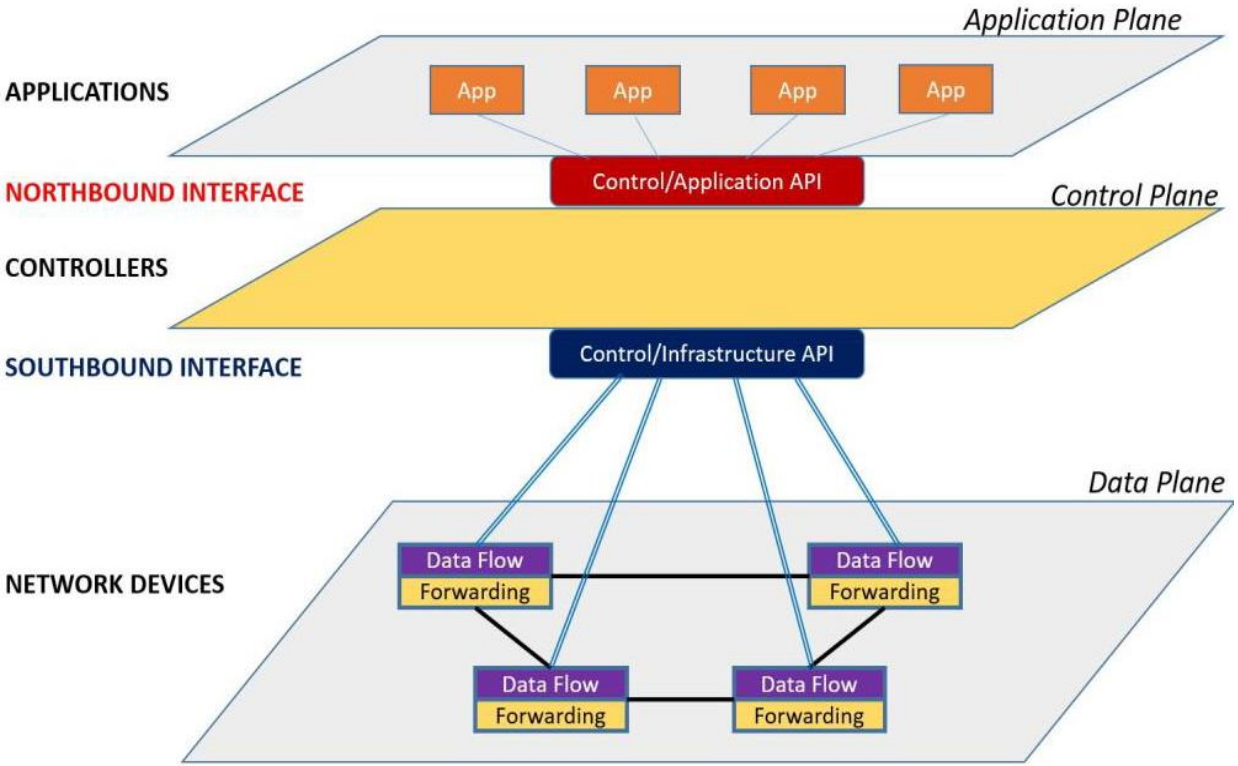


Fig. SDN Architecture [1]

SDN Architecture

3 main planes: application plane, control plane, data plane

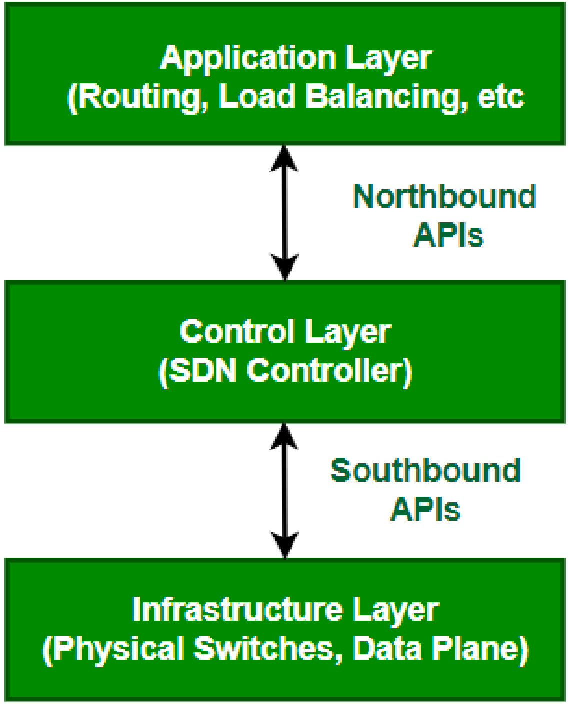


Fig. SDN Architecture [2]

SDN Principles

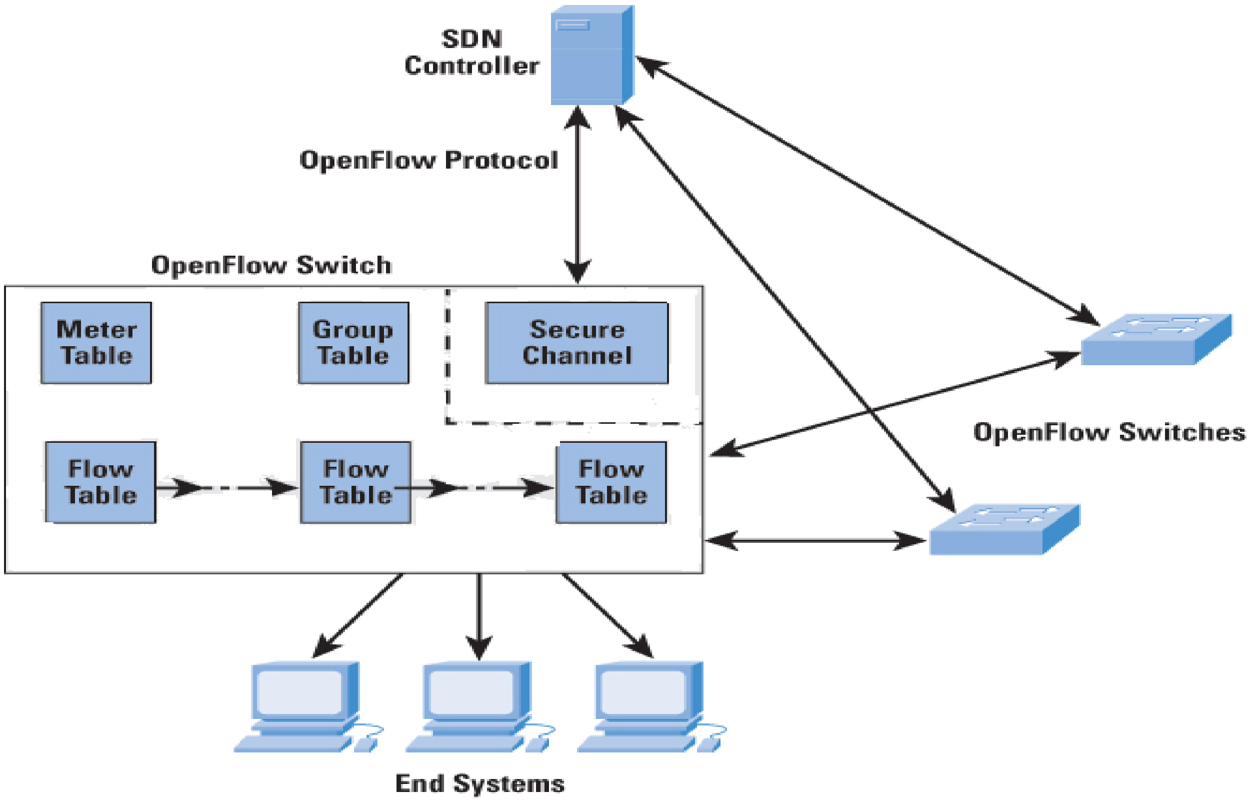


Fig. SDN Principles (1)

SDN – Application Plane

- Houses network services and applications
- Can access the global network view and use the underlying services to execute a function by using a high-level programming language in the control plane
  - Requirements for an SDN application are defined and translated into commands to program SDN switches.
- Network services are used to execute network applications and provide them with APIs to communicate with other planes.

SDN – Control Plane

- SDN vs. traditional control plane
  - programing different data plane elements with a standard programming interface, for example, OpenFlow protocol.
  - existing on a separate hardware device rather than on the forwarding devices
  - programing multiple data plane elements from a single control plane instance

SDN – Control Plane

- Two main roles:

1) managing the infrastructure layer and implementing policy decisions to the data plane via the southbound interface;

2) providing a global view of the underlying network to the application layer through the northbound interface

SDN – Control Plane

- Managed by one or multiple SDN controllers
- An SDN controller
  - pillar of the SDN architecture
  - performance and scalability of the controller are dependent on the control platform architecture
  - Key elements:
    - the applications;
    - the NOS
    - network abstraction
    - communication interfaces or APIs

SDN – Control Plane

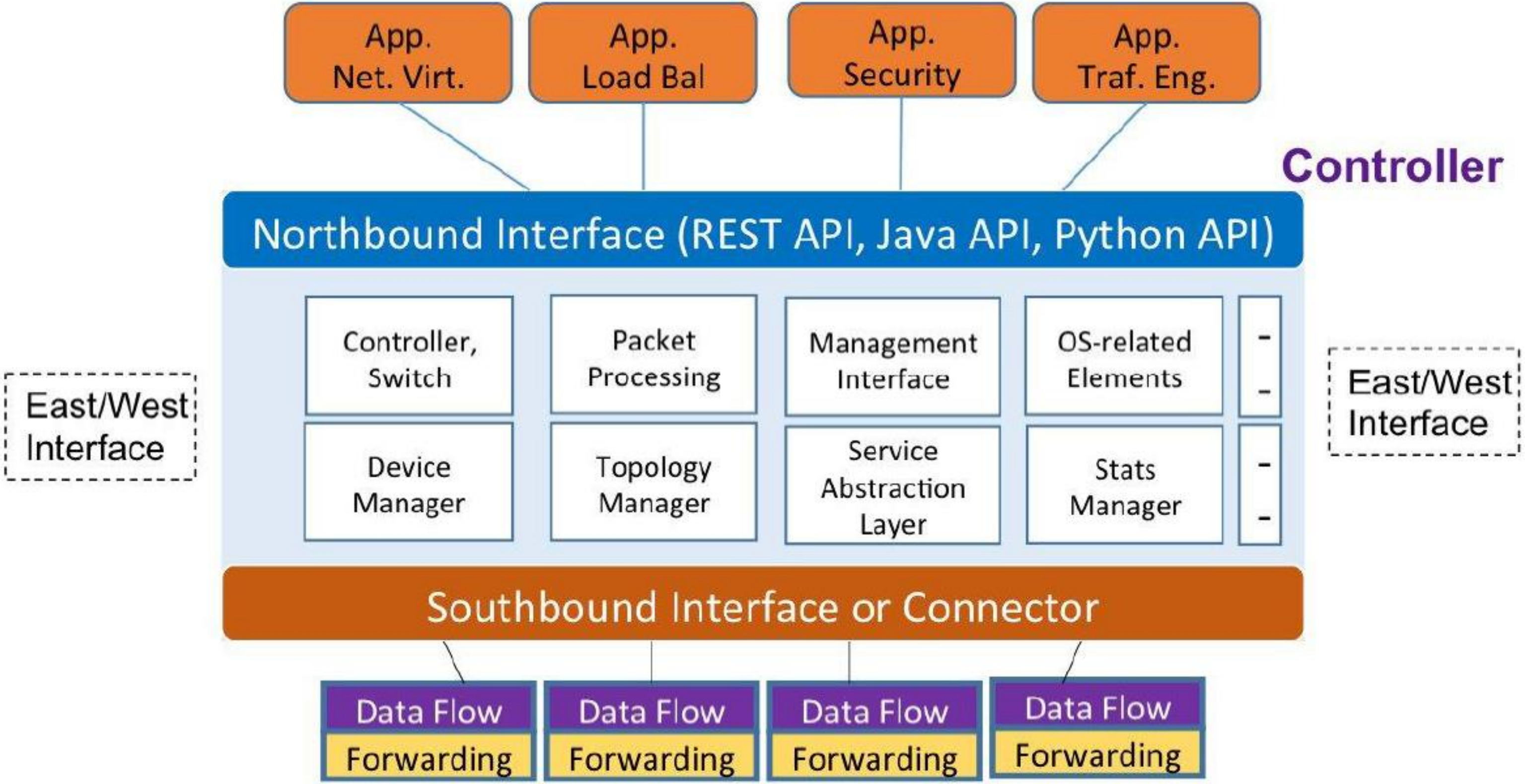


Fig. Components of an SDN controller (1)

SDN – Control Plane

Interfaces of an SDN controller

- Northbound interface or Control/Application interface
  - RESTful API, Ad Hoc APIs, etc.
- Southbound interface or Control/Infrastructure interface
  - OpenFlow, ForCES, SoftRouter
- East-West interface
  - Communication among distributed controllers

SDN – Control Plane

- SDN controllers and associated characteristic

Name	Architecture	Northbound API	Consistency	Faults	License	Prog. language	Version
Beacon [127]	centralized multi-threaded	ad-hoc API	no	no	GPLv2	Java	v1.0
DISCO [123]	distributed	REST	—	yes	—	Java	v1.1
Floodlight [128]	centralized multi-threaded	RESTful API	no	no	Apache	Java	v1.1
HP VAN SDN [122]	distributed	RESTful API	weak	yes	—	Java	v1.0
HyperFlow [133]	distributed	—	weak	yes	—	C++	v1.0
Kandoo [158]	hierarchically distributed	—	no	no	—	C, C++, Python	v1.0
Onix [7]	distributed	NVP NBAPI	weak, strong	yes	commercial	Python, C	v1.0
Maestro [126]	centralized multi-threaded	ad-hoc API	no	no	LGPLv2.1	Java	v1.0
Meridian [131]	centralized multi-threaded	extensible API layer	no	no	—	Java	v1.0
MobileFlow [154]	—	SDMN API	—	—	—	—	v1.2
MuL [159]	centralized multi-threaded	multi-level interface	no	no	GPLv2	C	v1.0
NOX [53]	centralized	ad-hoc API	no	no	GPLv3	C++	v1.0
NOX-MT [125]	centralized multi-threaded	ad-hoc API	no	no	GPLv3	C++	v1.0
NVP Controller [64]	distributed	—	—	—	commercial	—	—
OpenContrail [121]	—	REST API	no	no	Apache 2.0	Python, C++, Java	v1.0
OpenDaylight [13]	distributed	REST, RESTCONF	weak	no	EPL v1.0	Java	v1.{0,3}
ONOS [69]	distributed	RESTful API	weak, strong	yes	—	Java	v1.0
POX [160]	centralized	ad-hoc API	no	no	GPLv3	Python	v1.0
ProgrammableFlow [161]	centralized	—	—	—	—	C	v1.3
Ryu NOS [130]	centralized multi-threaded	ad-hoc API	no	no	Apache 2.0	Python	v1.{0,2,3}
SNAC [162]	centralized	ad-hoc API	no	no	GPL	C++	v1.0
Trema [129]	centralized multi-threaded	ad-hoc API	no	no	GPLv2	C, Ruby	v1.0
Unified Controller [117]	—	REST API	—	—	commercial	—	v1.0
yanc [134]	distributed	file system	—	—	—	—	—

Fig. Controller Classification (8)

SDN – Data Plane

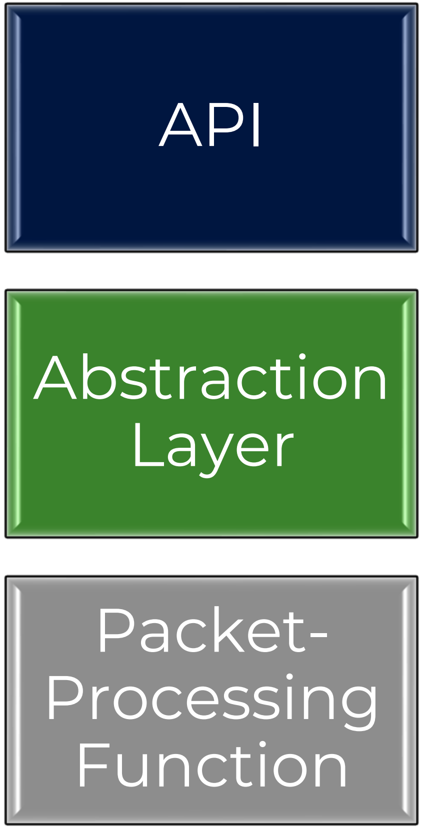
- The data plane comprises both virtual and physical SDN devices, usually known as SDN switches.

- Two main functions:
  - handling and forwarding traffic based on the rule information provided by the controller
  - gathering network state, temporally storing and transmitting them to the controller

Fig. Components of SDN switches and OpenFlow switches

SDN – Data Plane

- SDN device



SDN – OpenFlow Switch

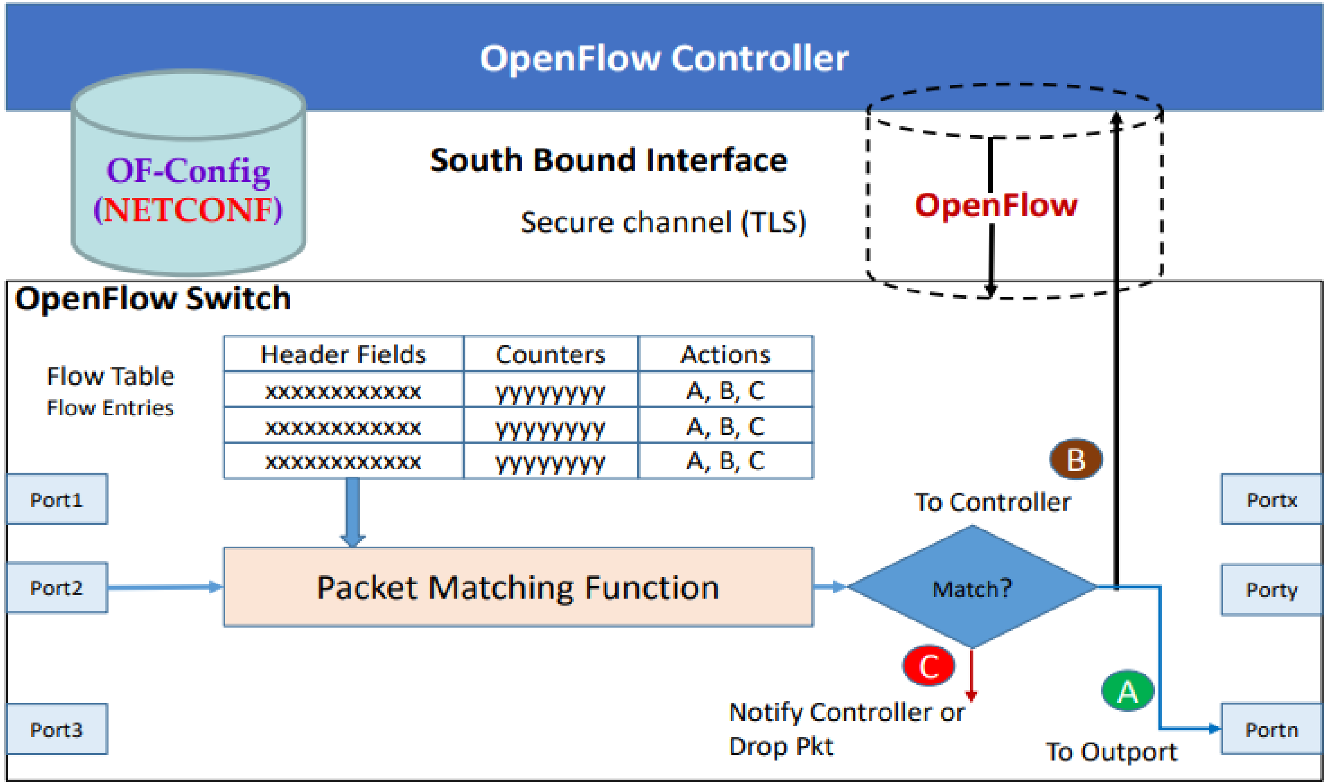
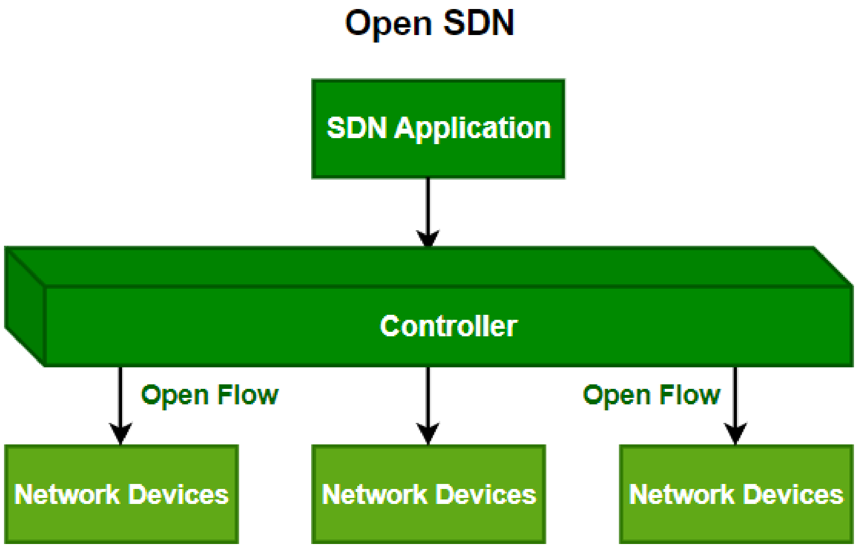


Fig. OpenFlow Switch – Basic Operation (1)

Open SDN

Open SDN is implemented using the OpenFlow switch. It is a straight forward implementation of SDN. In Open SDN, the controller communicates with the switches using south bound API with the help of OpenFlow protocol.



Difference Between SDN and Traditional Networking

Software Defined Networking	Traditional Networking
Software Defined Network is a virtual networking approach.	A traditional network is the old conventional networking approach.
Software Defined Network is centralized control.	Traditional Network is distributed control.
This network is programmable.	This network is nonprogrammable.
Software Defined Network is the open interface.	A traditional network is a closed interface.
In Software Defined Network data plane and control, the plane is decoupled by software.	In a traditional network data plane and control plane are mounted on the same plane.

SDN featured applications

- Network management
- SDN Network Virtualization for Cloud Computing
- SD-WAN (Software-defined Wide Area Network)

SDN – Challenging issues

- Non-standard NBI
- SDN security
- SDN for QoS
- SDN distributed controller
- SDN for IoT

One example of SDN

Cisco Application Centric Infrastructure (Cisco ACI)

- ACI is a software-defined networking (SDN) solution designed for modern data centers and cloud networks.
- ACI integrates hardware and software to automate network provisioning, optimize application performance, and enforce security policies.
- ACI provides centralized control, programmability, and policy-driven automation for managing complex networks.

SDN and Big Data Applications

Big data

- Big data is associated with storing and processing data sets and is defined with:
  - Five features: volume, variety, velocity, value, and veracity

SDN features for big data applications

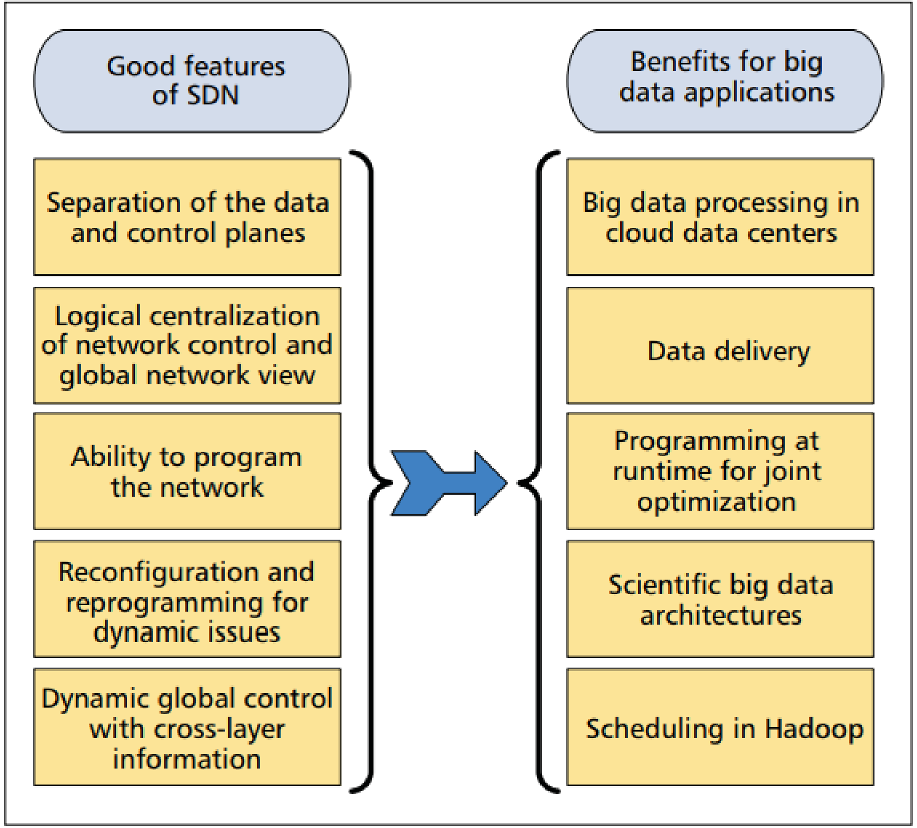


Fig. Good features of SDN that can benefit big data applications 7)

Big data can benefit SDN

- Traffic engineering in SDN
- Cross-layer design in SDN
- Defeating security attacks
- SDN-based intra- and inter-data-center networks with big data

Open issues

- Scalable controller management
- Intelligent flow table/rule management
- High flexible language abstraction
- Wireless mobile big data

Network Function Virtualization (NFV)

What is Network Function Virtualization (NFV)?

- NFV is a network architecture concept in which the network is implemented through software, virtualizing classes of network node functions.

- Under the NFV concept, virtualization technologies are used to implement network node functions on industry-standard commodity hardware, including servers, switches, and storage devices that can be moved to or instantiated in, various locations in the network as required, without the need for installation of new equipment.

What is Network Function Virtualization (NFV)?

What is Network Function Virtualization (NFV)?

- NFV refers to the use of virtual machines in place of physical network appliances. There is a requirement for a hypervisor to operate networking software and procedures like load balancing and routing by virtual computers.

- A network functions virtualization standard was first proposed at the OpenFlow World Congress in 2012 by the European Telecommunications Standards Institute (ETSI), a group of service providers that includes AT&T, China Mobile, BT Group, Deutsche Telekom, and many more.

NFV Features

- Allowing a pool of physical devices to be virtualized and chained into virtual networking functions that are provisioned as networking services.
- Separating the network functions from physical networking devices.
  - The network function being virtualized is termed a virtual network function (VNF).
- Virtualizing networking functions using virtualization technique
- Decoupling network functions from the network equipment
- Enabling network functions to be executed as software instance on demand

NFV Architecture

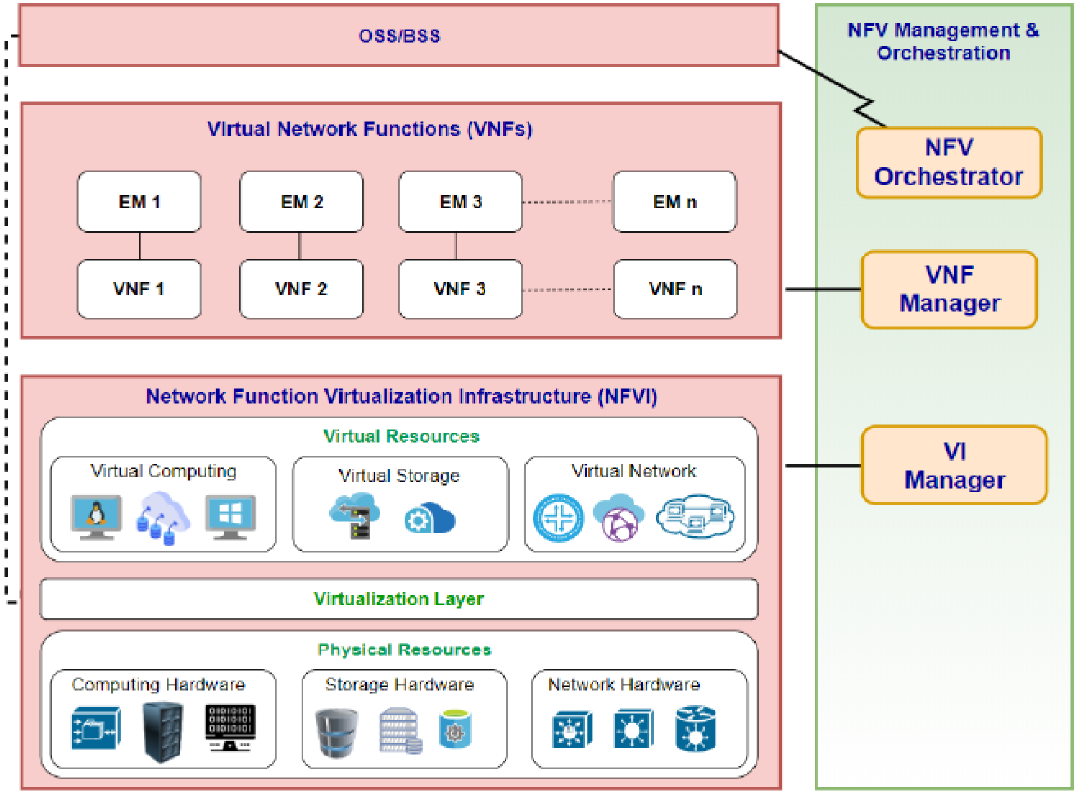


Fig. NFV Architecture (2)

- 3 main elements
  - NFV infrastructure
  - VNFs
  - NFV Management and orchestration (MANO)

Components of NFV

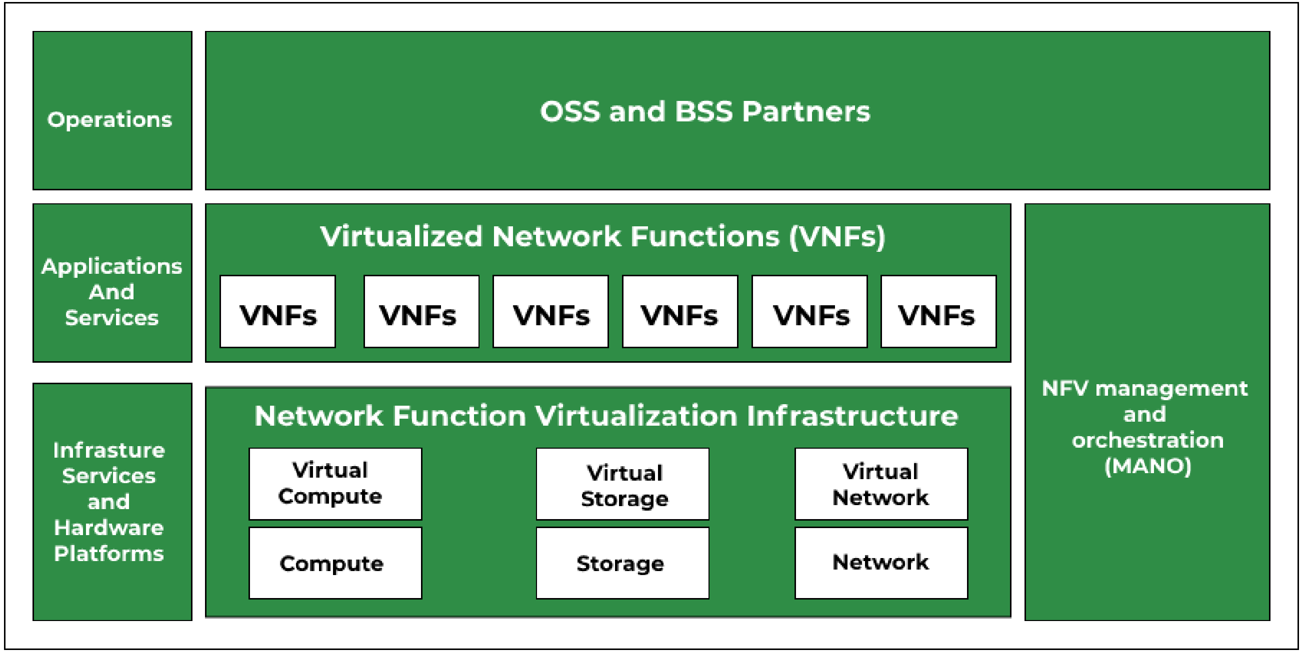
① **Centralized virtual network infrastructure:** The foundation of an NFV infrastructure can be either a platform for managing containers or a hypervisor that abstracts the resources for computation, storage, and networking.

② **Applications:** Software delivers many forms of network functionality by substituting for the hardware elements of a conventional network design (virtualized network functions).

③ **Framework:** To manage the infrastructure and provide network functionality, a framework is required (commonly abbreviated as MANO, meaning Management, Automation, and Network Orchestration).

How NFV works?

- Usage of software by virtual machines enables to carry out the same networking tasks as conventional hardware. The software handles the task of load balancing, routing, and firewall security.
- Network engineers can automate the provisioning of the virtual network and program all of its various components using a hypervisor or software-defined networking controller.



Relation of SDN and NFV

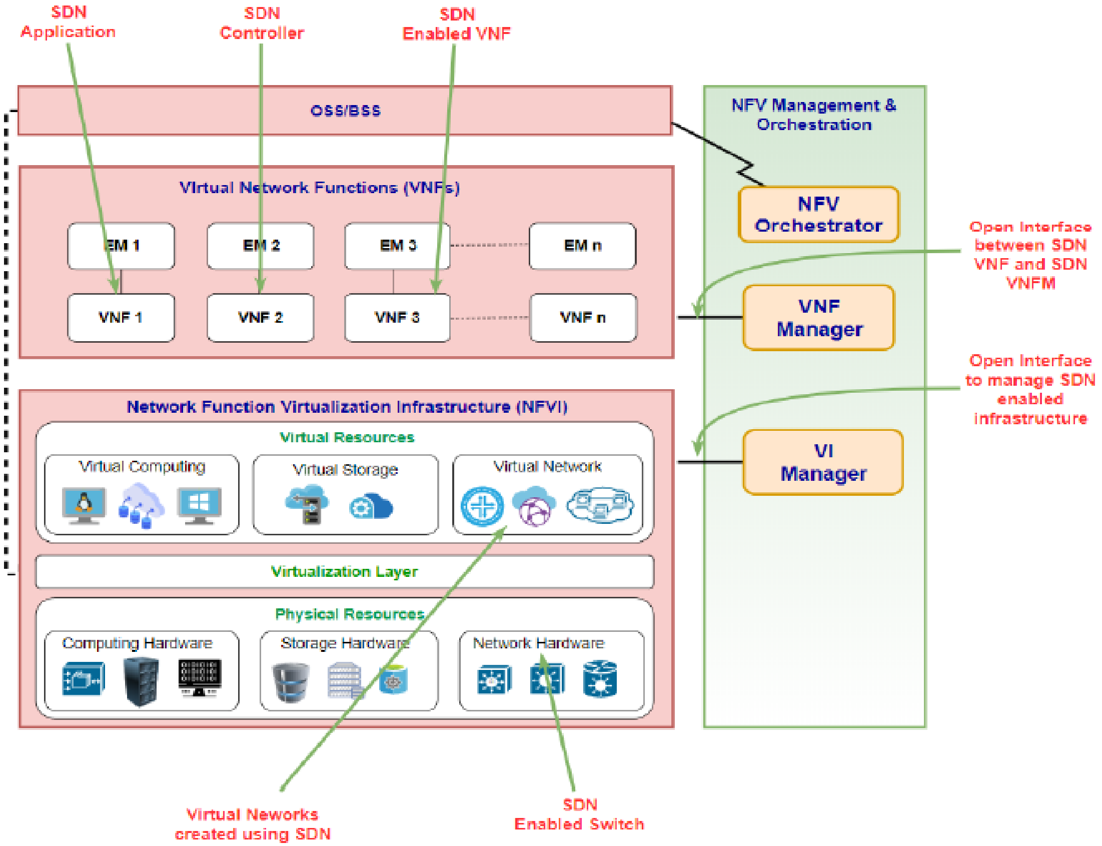


Fig. Mapping SDN Components to NFV Architecture [3]

Relation of SDN and NFV

Key Takeaways:

- **SDN and NFV** are distinct but related technologies aimed at making networks more agile and flexible.
- **SDN** separates the control plane from the data plane, allowing centralized network management and more efficient traffic routing.
- **NFV** virtualizes network functions, enabling them to run on standard servers for increased agility and cost savings.
- **SDN is mainly used** in data center or campus networks for centralized control, while **NFV is often used** in WANs to reduce physical device needs.
- **While both SDN and NFV** aim to improve network efficiency and reduce costs, they differ in architecture, deployment scenarios, and management requirements.
- **SDN offers** centralized management, improved network performance, and reduced costs but may present security risks and high deployment costs.
- **NFV offers** agile and flexible networks with reduced costs but may face challenges in management, orchestration, and certain deployment environments.

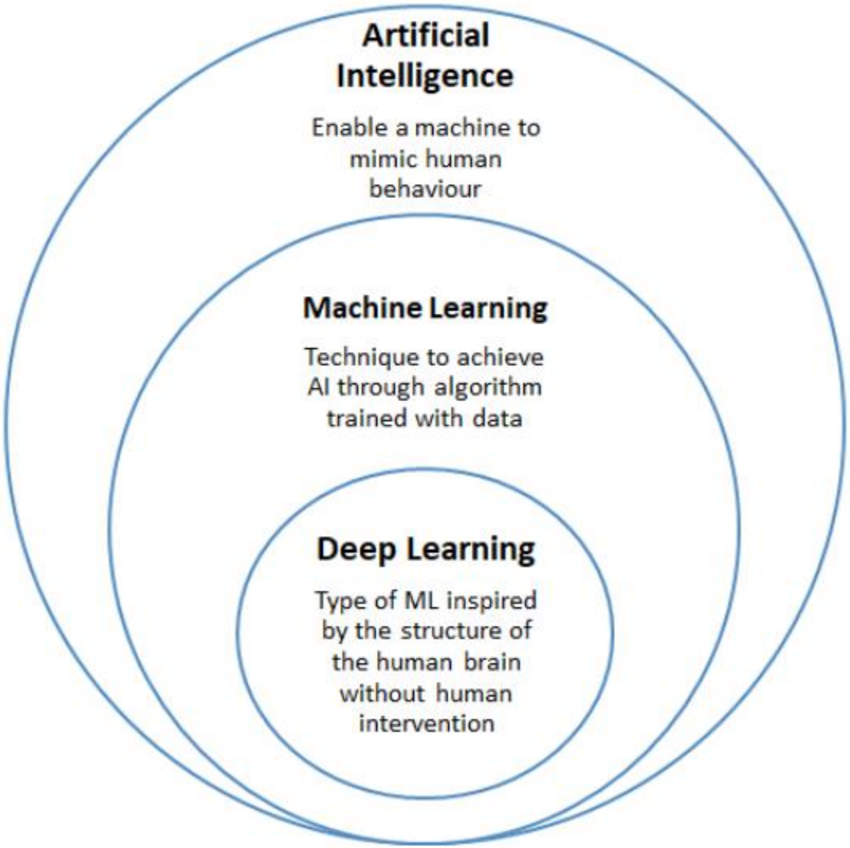
SDN	NFV
SDN architecture mainly focuses on data centers.	NFV is targeted at service providers or operators.
SDN separates control plane and data forwarding plane by centralizing control and programmability of network.	NFV helps service providers or operators to virtualize functions like load balancing, routing, and policy management by transferring network functions from dedicated appliances to virtual servers.
SDN uses OpenFlow as a communication protocol.	There is no protocol determined yet for NFV.
SDN supports Open Networking Foundation.	NFV is driven by ETSI NFV Working group.
Various enterprise networking software and hardware vendors are initiative supporters of SDN.	Telecom service providers or operators are prime initiative supporters of NFV.
Corporate IT act as a Business initiator for SDN.	Service providers or operators act as a Business initiator for NFV.
SDN applications run on industry-standard servers or switches.	NFV applications run on industry-standard servers.
SDN reduces cost of network because now there is no need of expensive switches & routers.	NFV increases scalability and agility as well as speed up time-to-market as it dynamically allot hardware a level of capacity to network functions needed at a particular time.
Application of SDN: • Networking • Cloud orchestration	Application of NFV: • Routers, firewalls, gateways • WAN accelerators • SLA assurance • Video Servers • Content Delivery Networks (CDN)

Machine Learning in Networking

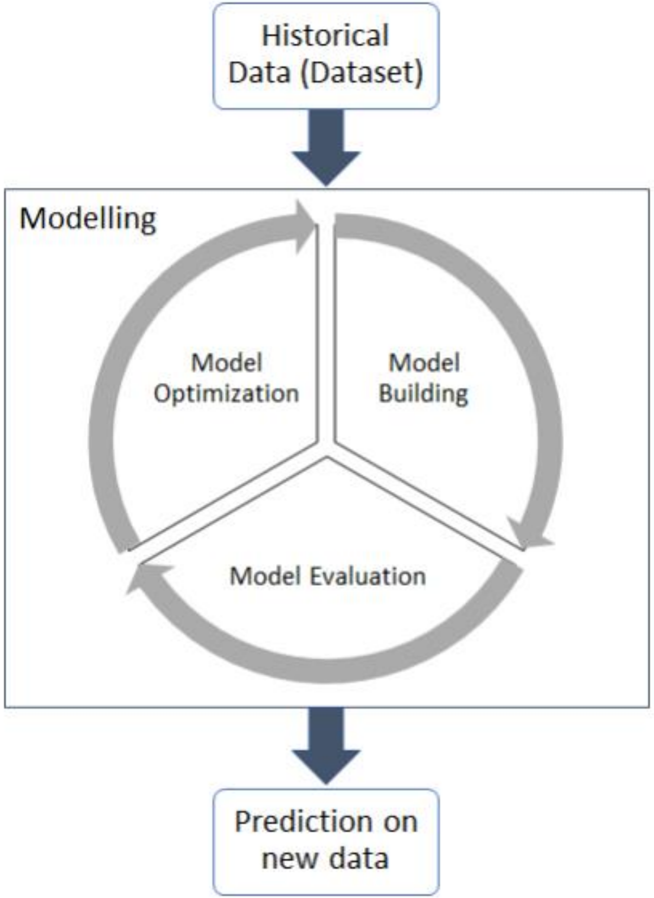
What is ML?

- "Machine learning (ML) is a subset of Artificial Intelligence (AI) application that allows systems to learn automatically and provides predictions or solutions based on experience"
- ML based applications
  - Image and speech recognition
  - Guiding systems
  - Communication networks

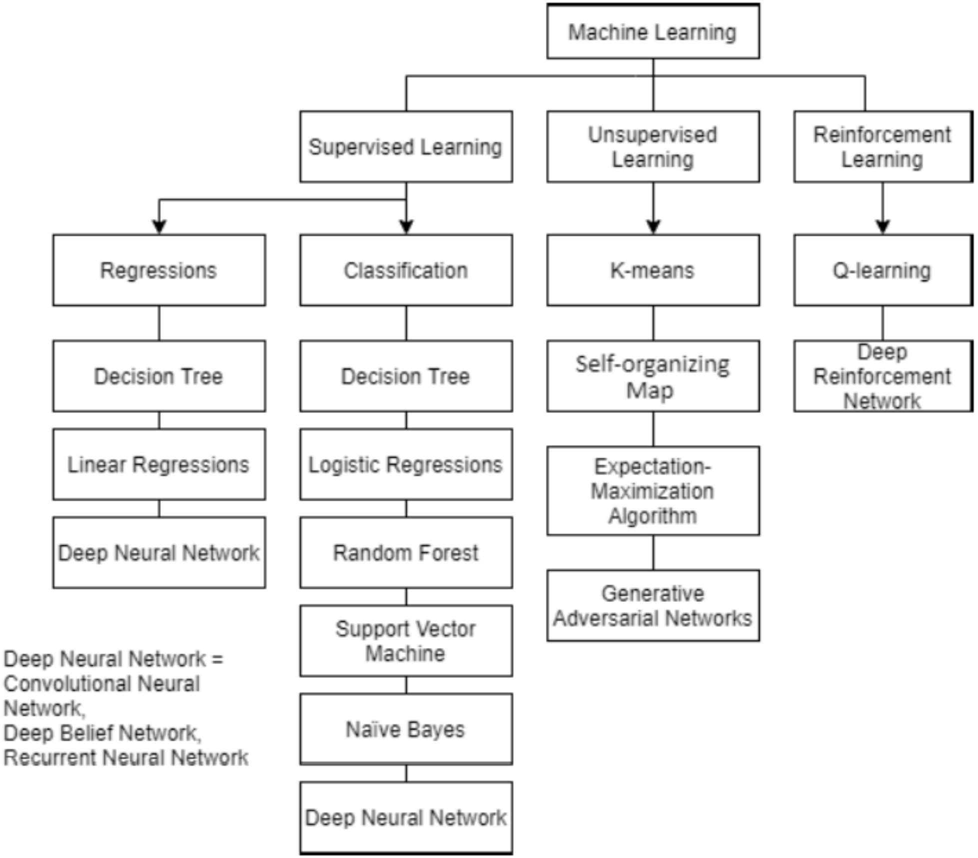
Difference between AI, ML, and DL [6]



ML – Basic workflow of real-world ML system [6]



ML model categories [6]



ML – based Application[6]

- A predictive model in communication network
  - Predicting the networks optical signal to noise ratio
  - Predicting the next node for the traffic forwarding
  - Predicting the link quality in the network
  - Predicting the traffic volume in the network
  - Predicting revenue in the 5G infrastructure

ML – based Application[6]

- ML-based IDS
  - Providing a general representation of known attacks from historical data
  - Categorized into host- and network-based IDS

Intrusion	Descriptions
Denial-of-Service (DoS)	The administrative server will be flooded with many service request and will fail to provide services to other legitimate users
Scanning attack (Probe)	Attacker attempt to locate the target systems in the network and subsequently exploit known vulnerabilities
Remote-to-local (R2L)	Attacker use to gain local access to the vulnerable machine provided the attacker can send packets to the victim machine over a network
User-to-root (U2R)	Exploits are used to gain the root access to a machine by an unprivileged local user

Fig. Common Attacks in the network (6)

ML – based Application[6]

- ML for improving routing decision in communication networks
  - Support choosing the best path for packet transmission
  - DL-based routing algorithm

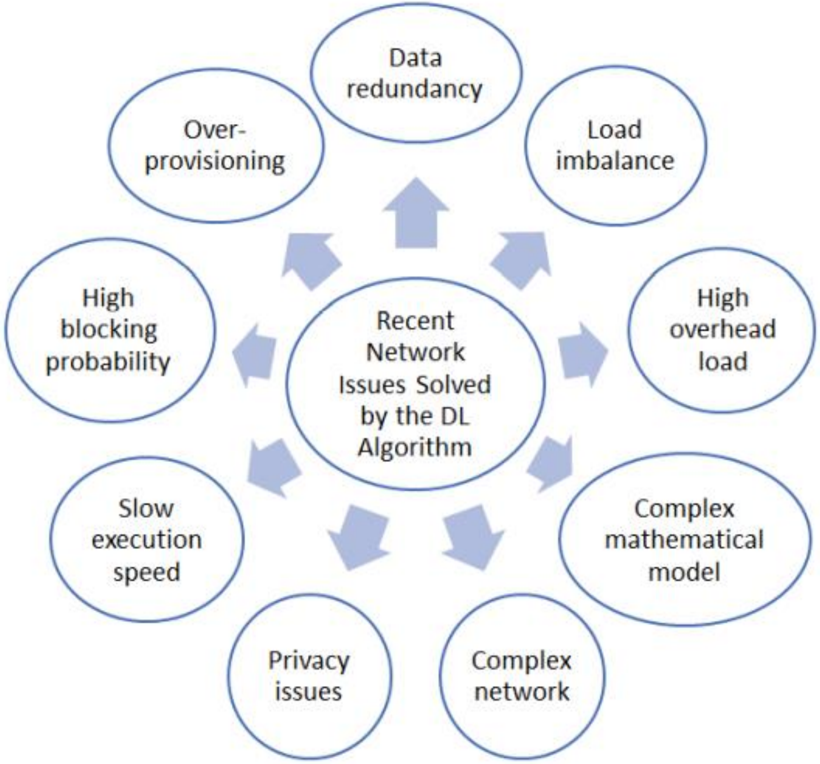


Fig. Recent network issues solved by DL-based routing algorithm[6]

ML – based Application[6]

- ML algorithm for improving QoS in a network communication system
  - Throughput maximization
  - Reducing network delay
  - Quality of experience (QoE) improvement

ML – based Application[6]

- ML algorithm for network resource management
  - Process of managing and allocating the available network resources for the network process
  - Network resources including switches, routers, bandwidth communication network, and spectrums
  - Traditional resource management are static-based approaches

Challenges and future research trends [6]

- High computational load and trade-off with ML accuracies
- Data availability and privacy issues
- Imbalanced dataset
- Tested for a real-world ML implementation study
- Hybrid ML algorithms for different network applications
- Advancement of 5G and future 6G

One example: using ML for anomaly detection

Step 1: Data Collection

Network traffic data is collected from sources such as:

- Routers, switches, and firewalls (NetFlow, sFlow data)
- Intrusion detection systems (IDS) logs
- Application logs and endpoint activity

This data typically includes features like:

- Packet size and type
- Source and destination IP addresses
- Protocol types (TCP, UDP, etc.)
- Port numbers
- Session durations

Step 2: Data Preprocessing

- **Feature Extraction**: Extract meaningful features like traffic volume, connection duration, and protocol usage.
- **Normalization**: Normalize data to ensure all features contribute equally to the ML model.
- **Data Cleaning**: Remove incomplete or redundant entries.

One example: using ML for anomaly detection

Step 3: Choosing an ML Model

Depending on the type of anomaly detection, different ML models can be used:

① **Unsupervised Learning** (for unknown attacks):

- Algorithms like K-Means Clustering or Isolation Forest are used to identify patterns that deviate from normal behavior.
- **Example**: Isolation Forest flags anomalous network traffic that doesn't fit with the majority of "normal" data points.

② **Supervised Learning** (for known attacks):

- Classification algorithms like Support Vector Machines (SVM), Decision Trees, or Neural Networks are trained on labeled data (normal vs. malicious traffic).
- **Example**: A dataset with labeled DDoS attacks trains a classifier to predict whether new traffic is malicious.

③ **Deep Learning** (for complex, high-dimensional data):

- Recurrent Neural Networks (RNNs) or Transformers are used to analyze time-series data, identifying subtle anomalies over time.
- **Example**: An RNN identifies a gradual increase in abnormal packet rates indicating a stealthy attack.

One example: using ML for anomaly detection

Step 4: Training the Model

The ML model is trained using historical network data, where "normal" and "anomalous" behaviors are labeled or inferred. The training process involves:

- Splitting the data into training and testing sets.
- Fine-tuning model parameters using techniques like cross-validation.

Step 5: Deployment and Real-Time Monitoring

- Deploy the trained ML model in a real-time monitoring system.
- Continuously analyze incoming network traffic.

References and Reading

[1] Hoang, D., *Software defined networking–shaping up for the next disruptive step?* Australian Journal of Telecommunications and the Digital Economy, 2015. **3**(4).

[2] D. B. Hoang and S. Farahmandian, "Security of Software-Defined Infrastructures with SDN, NFV, and Cloud Computing Technologies," in *Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications*, S. Y. Zhu, S. Scott-Hayward, L. Jacquin, and R. Hill, Eds., ed Cham: Springer International Publishing, 2017, pp. 3-32.

[3] ETSI, Network Functions Virtualisation (NFV) - Virtual Network Functions Architecture. 2014.

[4] Zeng, D, Gu, L, Pan, S & Guo, S 2019, Software Defined Systems, Springer International Publishing, Cham.

[5] Open Data Center Alliance Master Usage Model: Software-Defined Networking Rev. 2.0, 2013-2014.

[6] M. A. Ridwan, N. A. M. Radzi, F. Abdullah and Y. E. Jalil, "Applications of Machine Learning in Networking: A Survey of Current Issues and Future Challenges," in *IEEE Access*, vol. 9, pp. 52523-52556, 2021, doi: 10.1109/ACCESS.2021.3069210.

[7] L. Cui, F. R. Yu and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," in *IEEE Network*, vol. 30, no. 1, pp. 58-65, January-February 2016, doi: 10.1109/MNET.2016.7389832.

[8] Kreutz, Diego & Ramos, Fernando & Veríssimo, Paulo & Esteve Rothenberg, Christian & Azodolmolky, Siamak & Uhlig, Steve. (2014). Software-Defined Networking: A Comprehensive Survey. ArXiv e-prints. 103. 10.1109/JPROC.2014.2371999.

Thank you
Q&A ?

UNIVERSITY
OF WOLLONGONG
AUSTRALIA