Software Defined Networking for IoT

Dr. Bibhas Ghoshal

IIIT Allahabad

The Internet's Landscape

Innovation in Applications

Innovation in Technologies

Stagnant Internet Protocols **TCP/IP, BGP, DNS, OSPF**

Where is the Problem ?

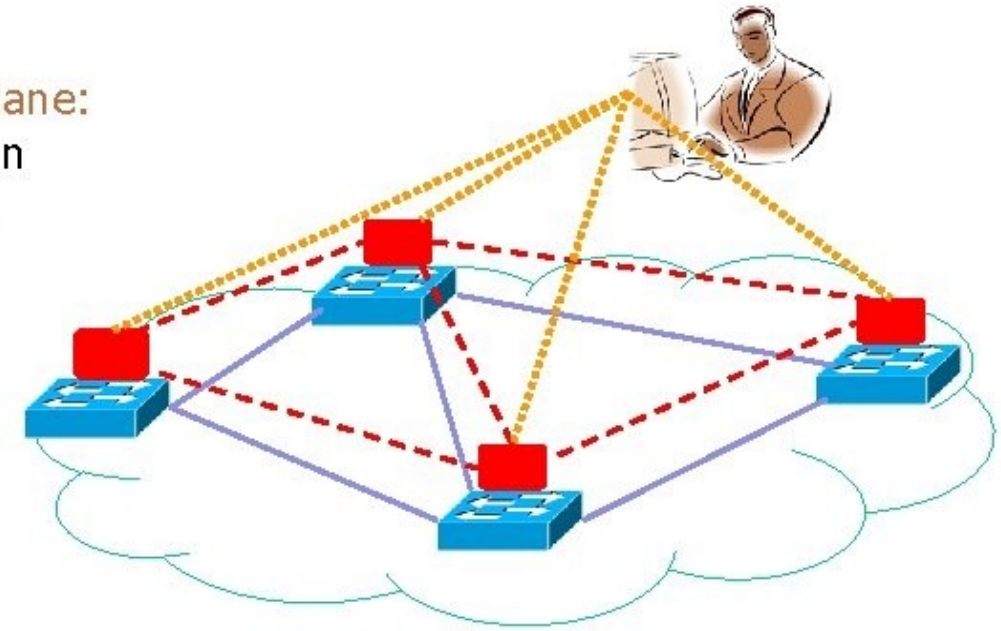
- **Closed equipment**
  - Software bundled with hardware
  - Vendor specific interfaces
- **Over Specified**
- **Few people who innovate**
- **Operating a network is expensive**
- **Buggy software in the equipment**

Traditional Computer Networks

Management plane:
Configuration

Control plane:
Distributed algorithms

Data plane:
Packet streaming



Proprietary software and vendor-specific configuration interfaces

A plane is an abstract conception of where certain processes take place. The term is used in the sense of "plane of existence."

No standardization

Control Plane

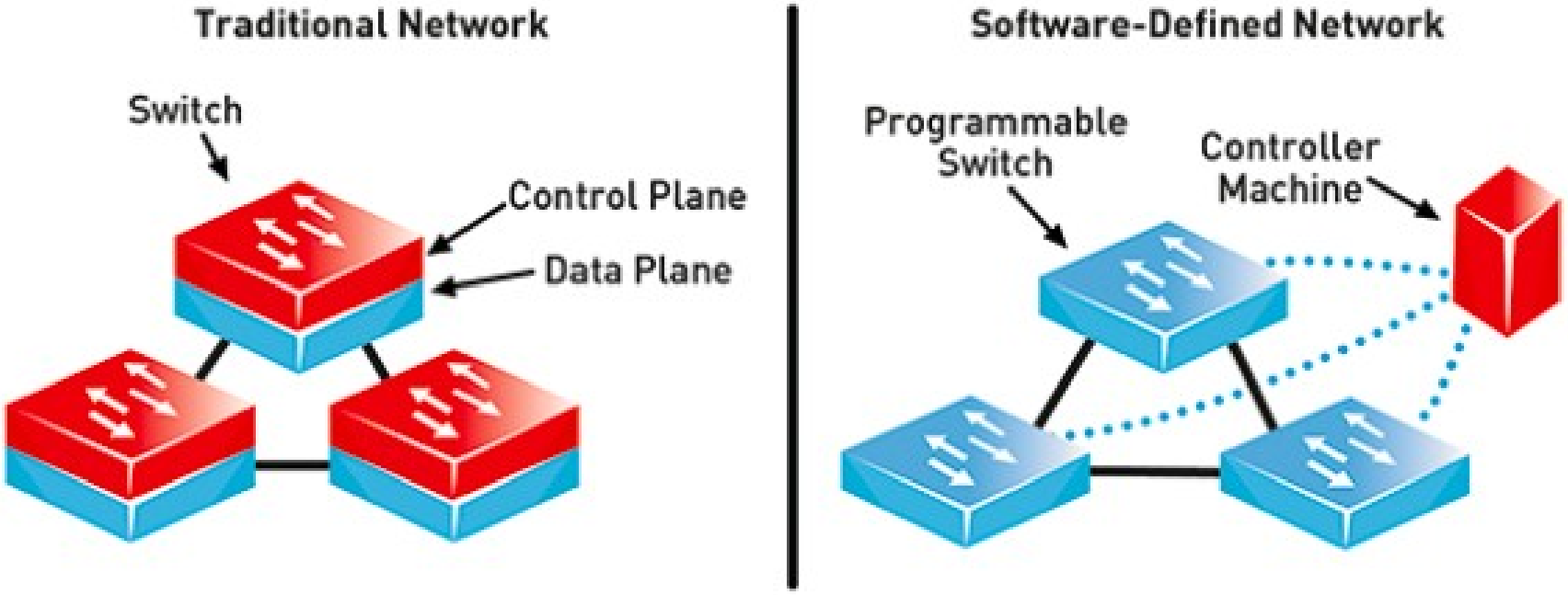
- **Part of a network that controls how data packets are forwarded — how data is sent from one place to another.**
- **The process of creating a routing table, for example, is considered part of the control plane.**
- **Routers use various protocols to identify network paths, and they store these paths in routing tables.**
- **Protocols routers use to create their routing tables :**
  - Border Gateway Protocol (BGP)
  - Open Shortest Path First (OSPF)
  - Enhanced Interior Gateway Routing Protocol (EIGRP)
  - Intermediate System to Intermediate System (IS-IS)
- **Track topology changes, compute routes, install forwarding routes**

Data Plane (Forwarding Plane) and Management Plane

- **Data Plane**
- **Determines how packets should be forwarded**
- **Matching on some bits, taking simple actions**
- **Functionality**
  - Forwarding
  - Access control
  - Mapping header files
  - Traffic monitoring
  - Buffering and marking
  - Shaping and scheduling
- **Management Plane**
  - Collect measurements and Configure equipments

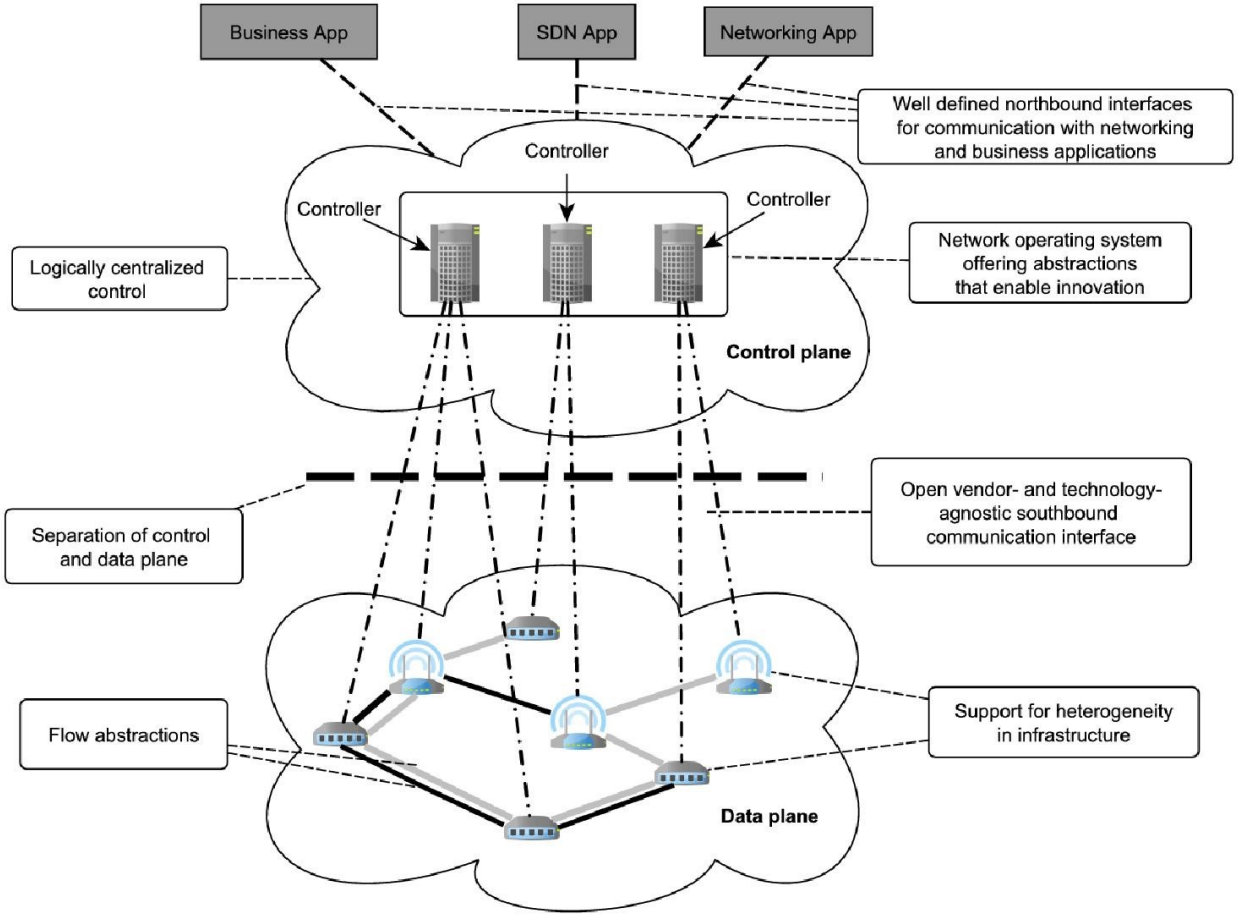
Separating the Control and Data Plane

Vendors provide the hardware (data plane) and we decide control plane by writing custom logic - software



Software Defined Networking (SDN) : Concept

According to Cisco : Software-defined networking (SDN) is an architecture designed to make a network more flexible and easier to manage. SDN centralizes management by abstracting the control plane from the data forwarding function in the discrete networking devices.



Why SDN?

- **Virtualization: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.**
  - Abstraction ⇒ Virtualization.
- **Orchestration: Should be able to control and manage thousands of devices with one command.**
- **Programmable: Should be able to change behaviour on the fly**
- **Dynamic Scaling: Should be able to change size, quantity**
  - **Virtualization ⇒ Scaling**
- **Automation: To minimize manual involvement**

Troubleshooting, Reduce downtime, Policy enforcement, Provisioning/Re-provisioning/Segmentation of resources, Add new workloads, sites, devices, and resources

http://www.cse.wustl.edu/~jain/cse570-18/

Why SDN?

- **Visibility: Monitor resources, connectivity**
- **Performance: Optimize network device utilization, Traffic engineering/Bandwidth management, Capacity optimization, Load balancing, High utilization, Fast failure handling**
- **Multi-tenancy: Tenants need complete control over their addresses, topology, and routing, security**
- **Service Integration: Load balancers, firewalls, Intrusion Detection Systems (IDS), provisioned on demand and placed appropriately on the traffic path**

Openness: Full choice of "How" mechanisms

⇒ Modular plug-ins

⇒ Abstraction:

Abstract = Summary = Essence = General Idea ⇒ Hide the details

Define tasks by APIs and not by how it should be done. E.g., send from A to B.

http://www.cse.wustl.edu/~jain/cse570-18/

How SDN works ?

Devices in SDN : Controller and Switches

Switches have no built-in features and need to be instructed by the controller

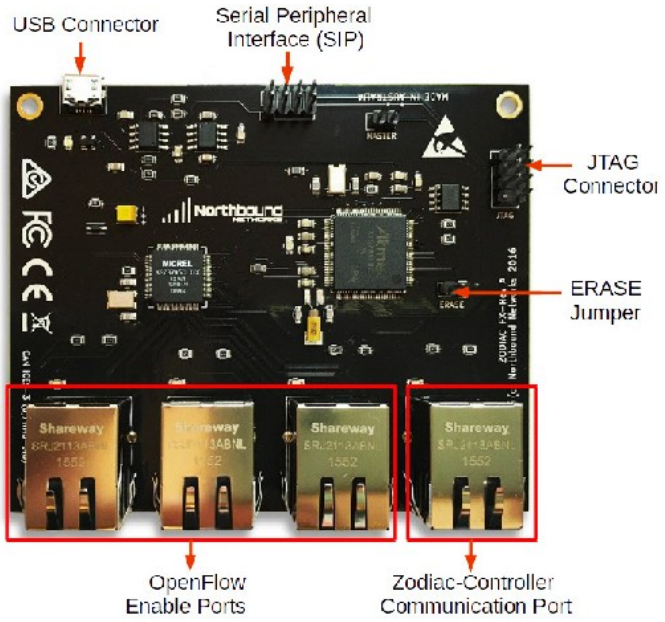


Fig. 1: Communication peripherals of Zodiac-FX

Zodiac FX : 4-port 10/100M Ethernet switch that was driven by the need for a low-cost option for SDN experimentation and home use

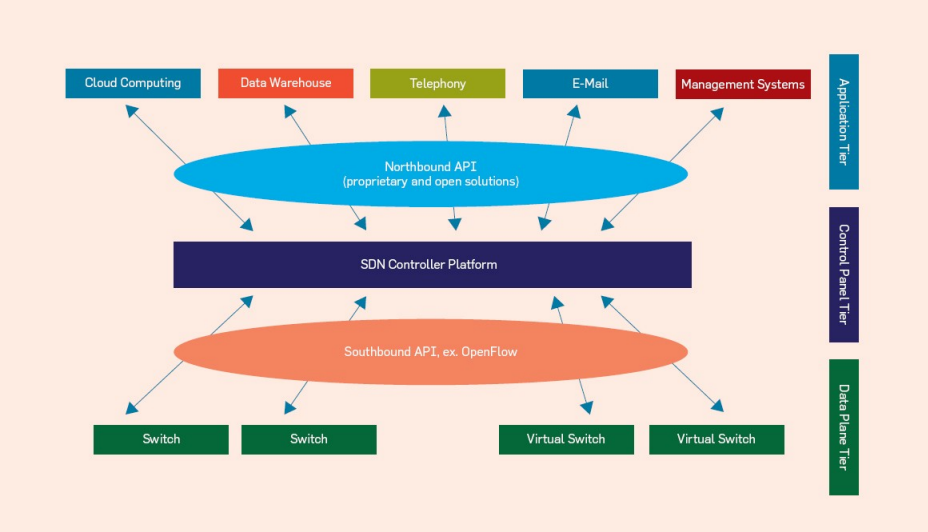
How SDN Works ?

API used :

Northbound API - interface application layer with control layer; provides abstract view of network to application layer

South bound API – interface controller and infrastructure layer; controllers can deploy different rules on routers and switches and they can communicate with controller in real time

East and West bound API – interfacing multiple controllers so that they can coordinate decisions

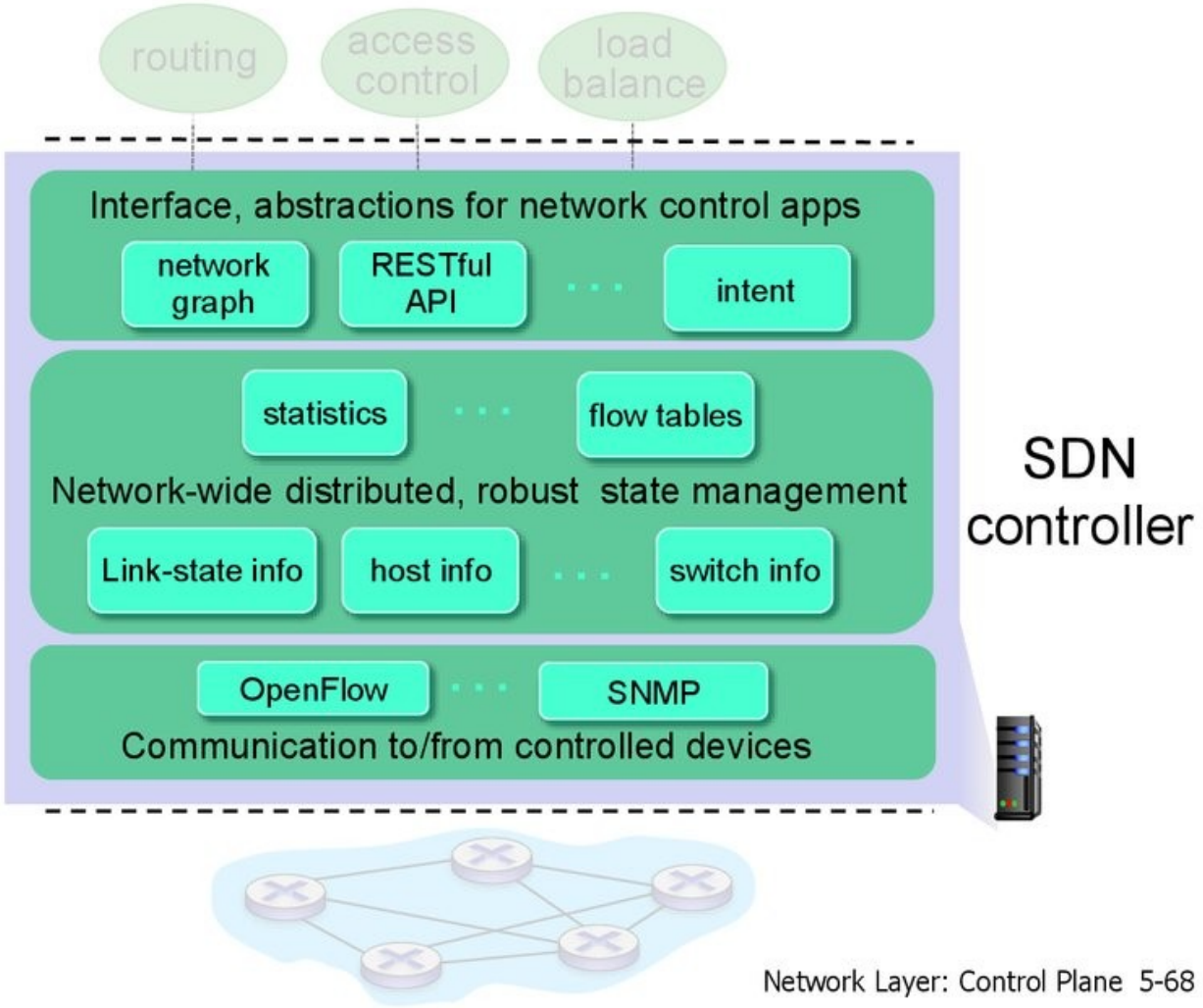


Components of SDN controller

Interface layer to network control apps: abstractions API

Network-wide state management layer: state of networks links, switches, services: a *distributed database*

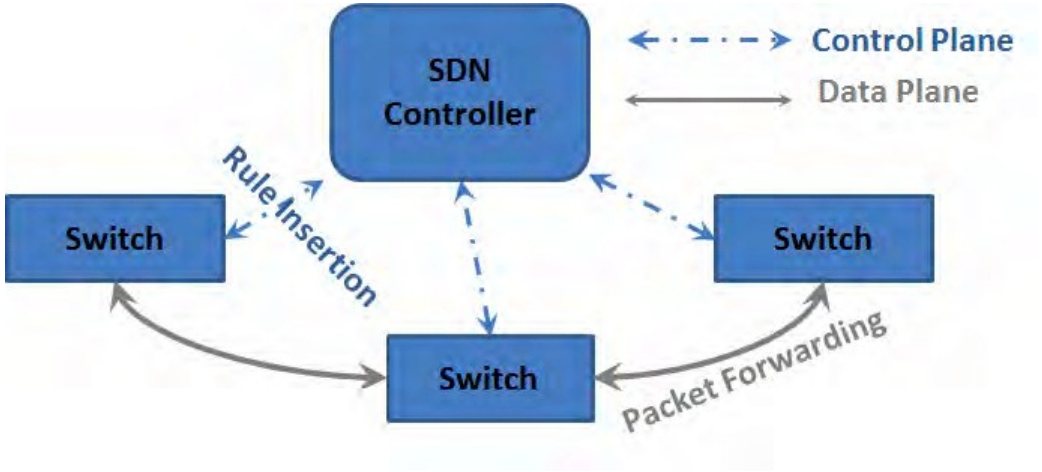
communication layer: communicate between SDN controller and controlled switches



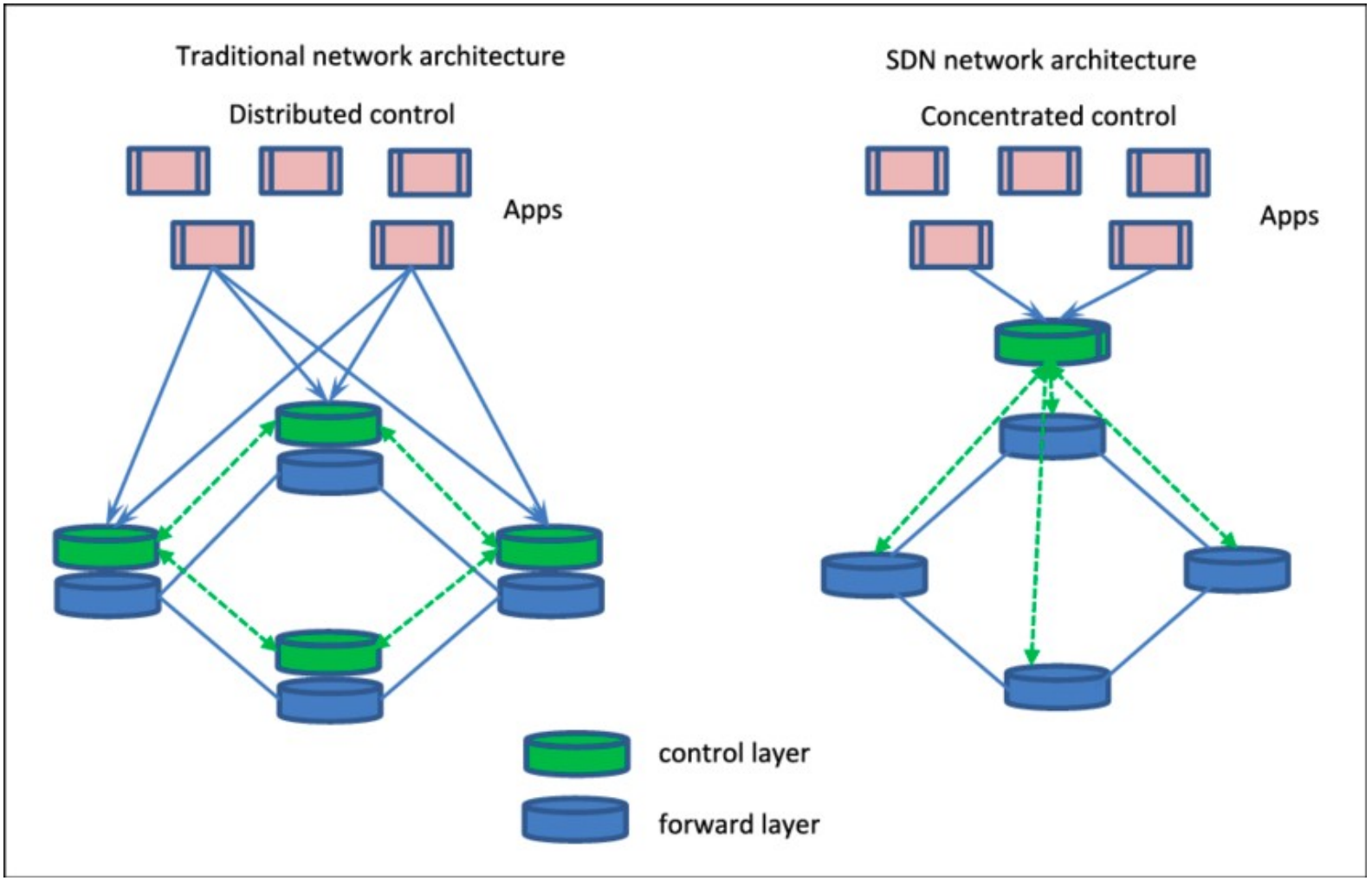
SDN Controller

An application that manages flow control to enable intelligent networking

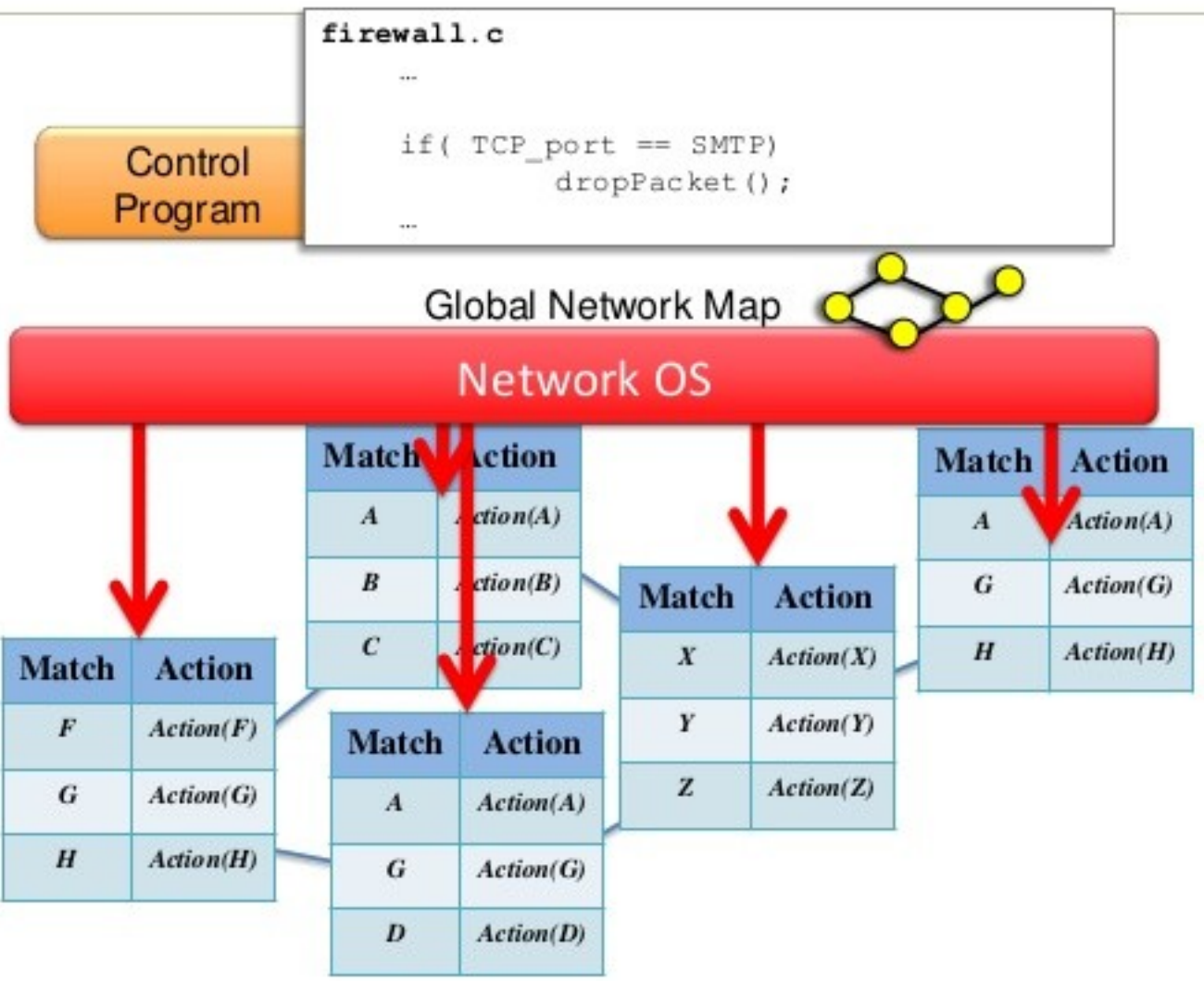
They are based on protocols such as Open flow that allow servers to tell switches where to send packets



SDN Architecture

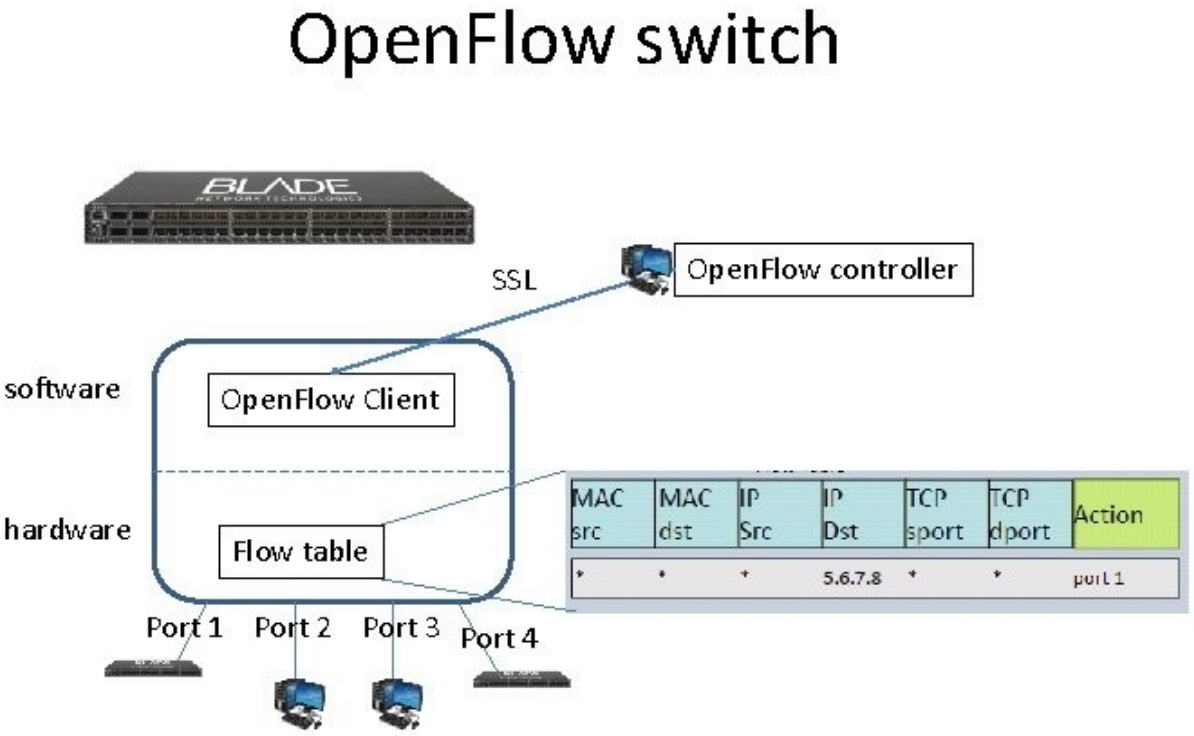


Software Defined Network (SDN)

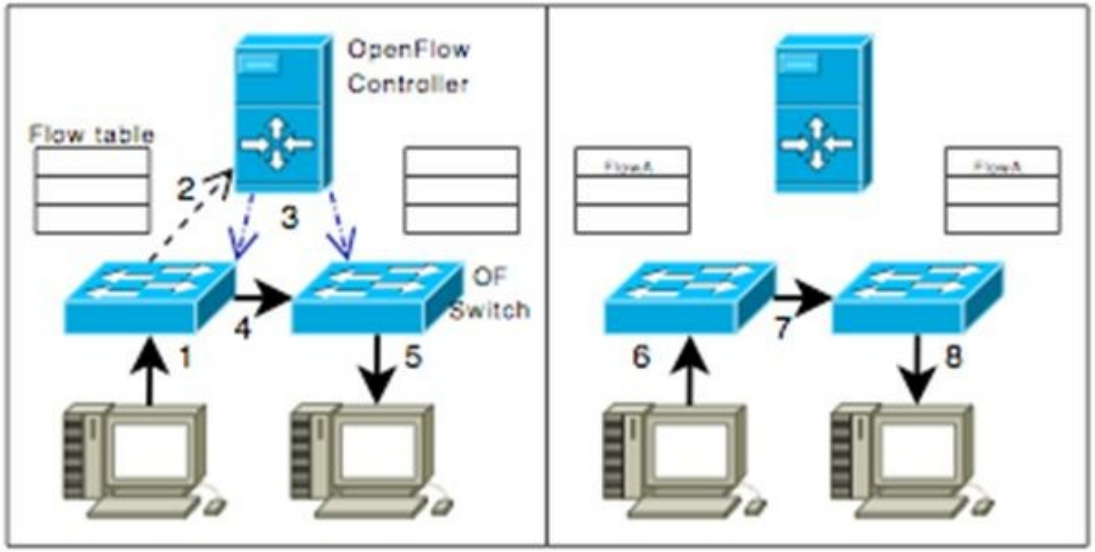


Open Flow

- **Protocol for controlling the forwarding behaviour of ethernet switches in a SDN**
- **Specifications maintained by Open Networking Forum**



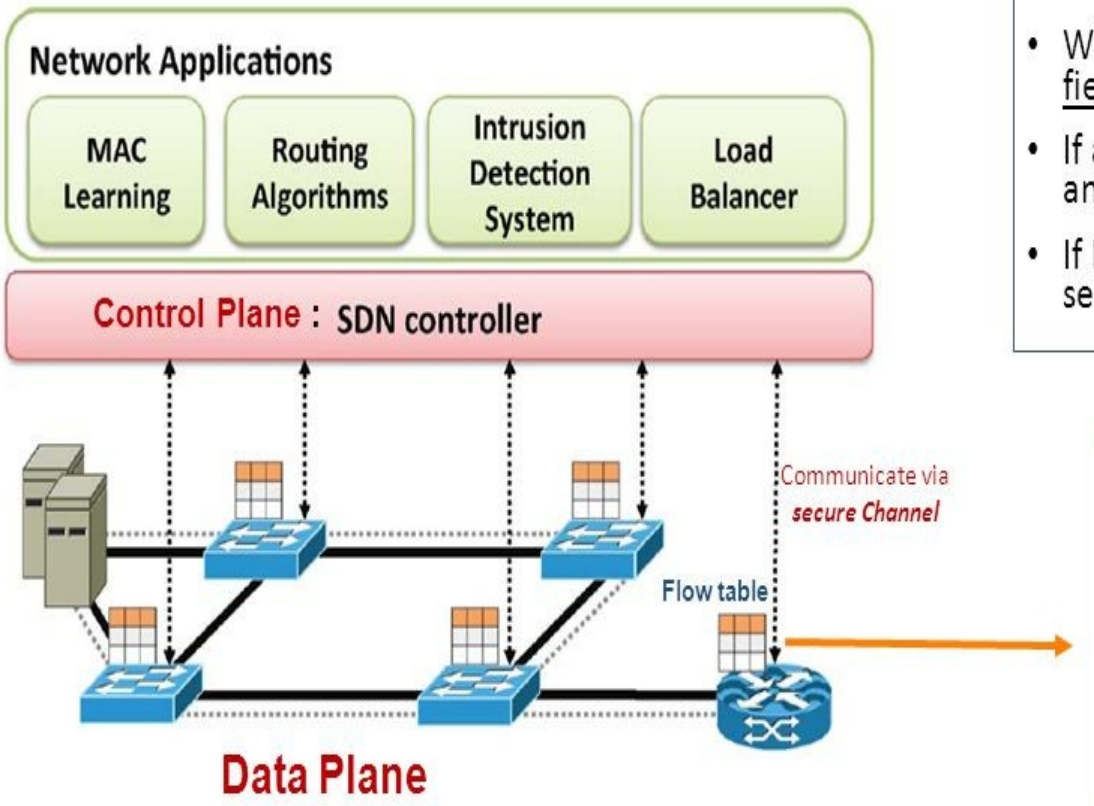
How OpenFlow Switch works?

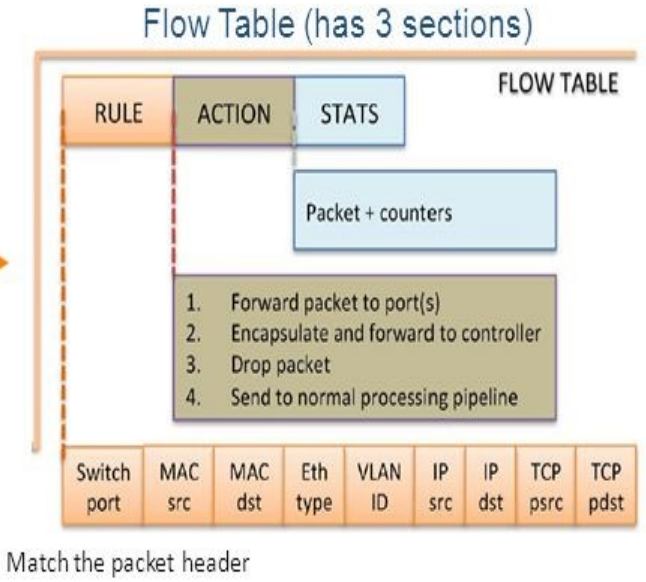


- When an OpenFlow Switch receives a packet it has never seen before, for which it has no matching flow entries, it sends this packet to the controller.
- The controller then makes a decision on how to handle this packet. It can drop the packet, or it can add a flow entry directing the switch on how to forward similar packets in the future.

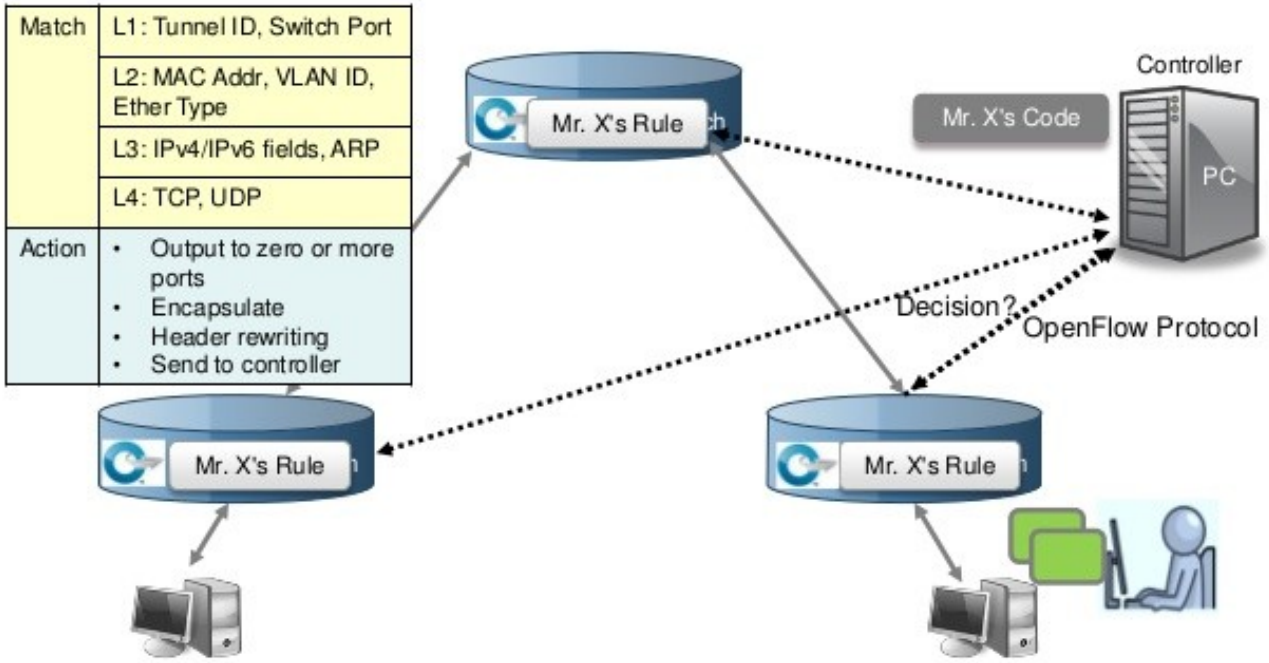
Basic OpenFlow: How Does it Work?

- Controller manages the traffic (network flows) by manipulating the flow table at switches.
  - Instructions are stored in flow tables.
- When packet arrives at switch, match the header fields with flow entries in a flow table.
- If any entry matches, performs indicated actions and update the counters.
- If Does not match, Switch asks controller by sending a message with the packet header.





How OpenFlow Works?



OpenFlow offloads control intelligence to a remote software!