Introduction

Software-defined networking (SDN), as the name suggests is a networking platform defined by software as per the vendor requirements. It therefore is an approach to networking which decouples the control from the hardware and gives it to the software application called the controller.

The traditional networks are controlled by switches and routers. Whenever the packet arrives at them, the switch looks into its inbuilt rules to forward the packet to its destination. Every packet destined for the same destination address is routed along the same path. In enterprise network, smart switches designed with application specific integrated circuits (ASICs) are used to identify different packets and route them differently. However, as they compute a smart way of routing packets, these switches are quite expensive.

SDN allows network administrators called the hypervisors to shape the network and control the logic. A program is fed to the controllers rendering the switch dumb. Hence, every task is performed under the supervision of the controller. The administrator can change the rules in the controller whenever necessary, prioritizing, de-prioritizing and even dropping the packets based on certain criterion. SDN have found its wide application in cloud where the administrator manages the load. In this way the network administrator can use less expensive commodity switches, taking direct control over the traffic.

SDN uses the OpenFlow protocol which provides the open standard API to help the network administrators to remotely control the routing tables.

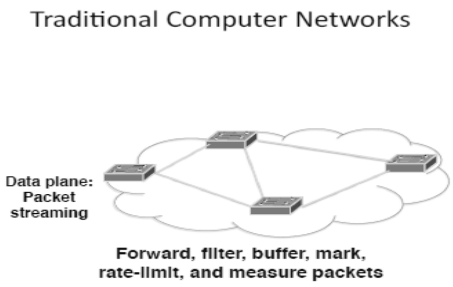
Traditional Computer Network and their Limitations

In a traditional network, data flow is controlled by switches and routers.Basic elements of switches and routers are:

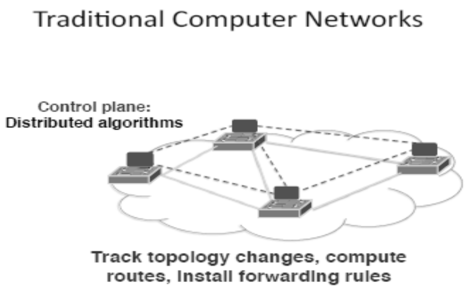
Data Plane: Physically carries data from one port to another.

Control Plane: Contains logic the device use to program data plane.

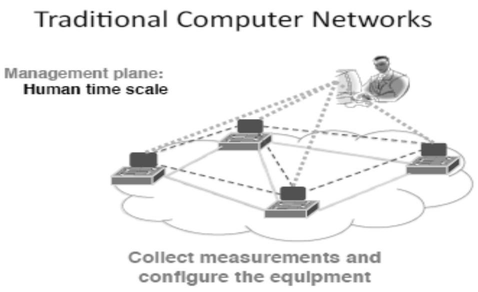
Management Plane: Allows administrator to configure the network.


Traditional Computer Networks
Data plane: Packet streaming
Forward, filter, buffer, mark, rate-limit, and measure packets

Data Plane


Traditional Computer Networks
Control plane: Distributed algorithms
Track topology changes, compute routes, install forwarding rules

Control Plane


Traditional Computer Networks
Management plane: Human time scale
Collect measurements and configure the equipment

Management Plane

The traditional network has certain limitations in terms of its fixed nature that adds to the infrastructural costs, inconsistent policies, and absence of scalability, vendor dependence and complexity.

Picture courtesy: http://www.datacenterknowledge.com/archives/2012/08/30/virtualization-of-datacenters-new-options-in-the-control-and-data-planes/

Present market requirements are hard to be accomplished using the traditional network architectures. Many IT companies are trying to use various device level management tools to face the challenges as today's user demand are based on mobility and efficient bandwidth requirement. Following are the limitations of the current networks, which include:

Network complexity: The fixed nature of the network requires a well maintained infrastructure to support multiple communications. This increases the demand for manpower and different protocols for different services need to be defined. Moreover mobility is not supported. Updating the network (adding or removing devices) requires to update the network devices individually hence leads to latency decreasing performance and reliability.

Inconsistent policies: Policies define a set of rules to be followed by each component in a network to ensure its successful working. These policies are to be configured in each and every device of the network. This type of configuration in the current networks takes time i.e. from few hours to days e.g. reconfiguring the ACLs across the entire network, applying set of access, security, QoS etc.

Inability to scale: Data centers are the repository of data serving the rapidly growing demands of users that directly tend to rapid growth of network. Hence, the network becomes vastly more complex with the addition of hundreds and thousands of network devices that must be managed and configured regularly.

Vendor Dependence: Changing business needs and user demands to deploy new services and capabilities in current networks cannot be achieved timely and hence, hinders innovation.

Frequent changing traffic patterns, consumerization of IT, rise of cloud services and requirements for more bandwidth are some of the key computing trends driving the trend for a new network paradigm.

Motivation of SDN – ETHANE

ETHANE-an initial OpenFlow based network is a network architecture which provides a simple management model with strong security. Defined over human friendly names, ETHANE allows network manager to define a single network wide, fine grain policy and then enforce it at each switch. To make network more manageable and secure, it uses two approaches. One approach introduces the middleboxes at the network choke points to exert efficient control. Another approach is to add functionality to existing networks by providing tools, access control lists, filters, routing algorithms etc. to support better connectivity.

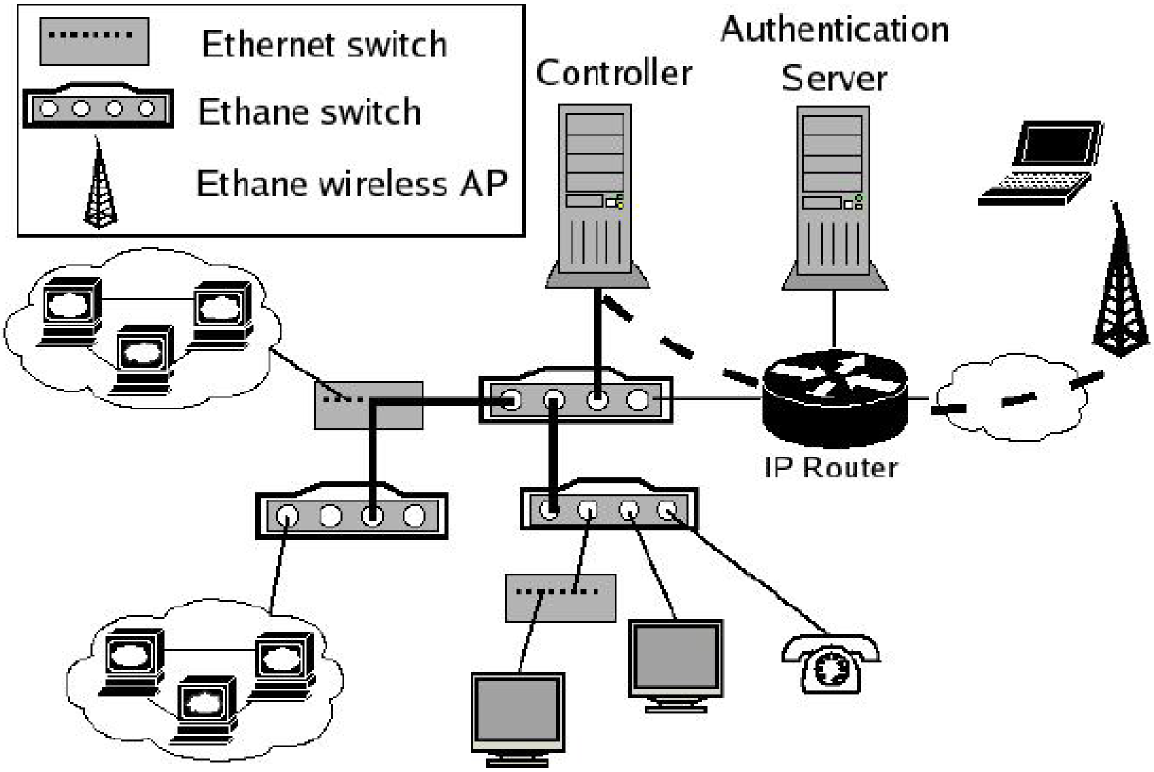
ETHANE is built around three fundamental principles:

- The networks should be governed by policies declared over high-level names rather than the low-level names. This means that the network should be managed in terms of the entities (users, hosts, access points) we desire to control.
- The packet flow should be governed by policies that determine the path to be followed i.e. firstly; the policy may require the packet to be passed through the intermediate middlebox. Secondly, if the path is controlled the packet flow will be serviced in an appropriate manner.
- The network should have a strong binding between a packet and its origin.

ETHANE network design

ETHANE prevents communication between end-hosts without explicit permissions thus, controlling the network. It contains the central controller and ETHANE switches. Controller is fed with the global network policy that determines the behavior of packets. When the packet arrives at the controller, it decides whether the flow should be allowed or not. Moreover, it performs route computation for permitted flows.

ETHANE switches are simple and dumb. These consist of a simple flow table and secure channel to the controller. Switches simply act on the direction of the controller to forward packets. When a packet arrives that doesn’t match the entry of the flow table, it is directed towards the controller along with the information about the port on which it arrived. If the packet arrived matches the entry of the flow table, it is forwarded as per the controllers directions.

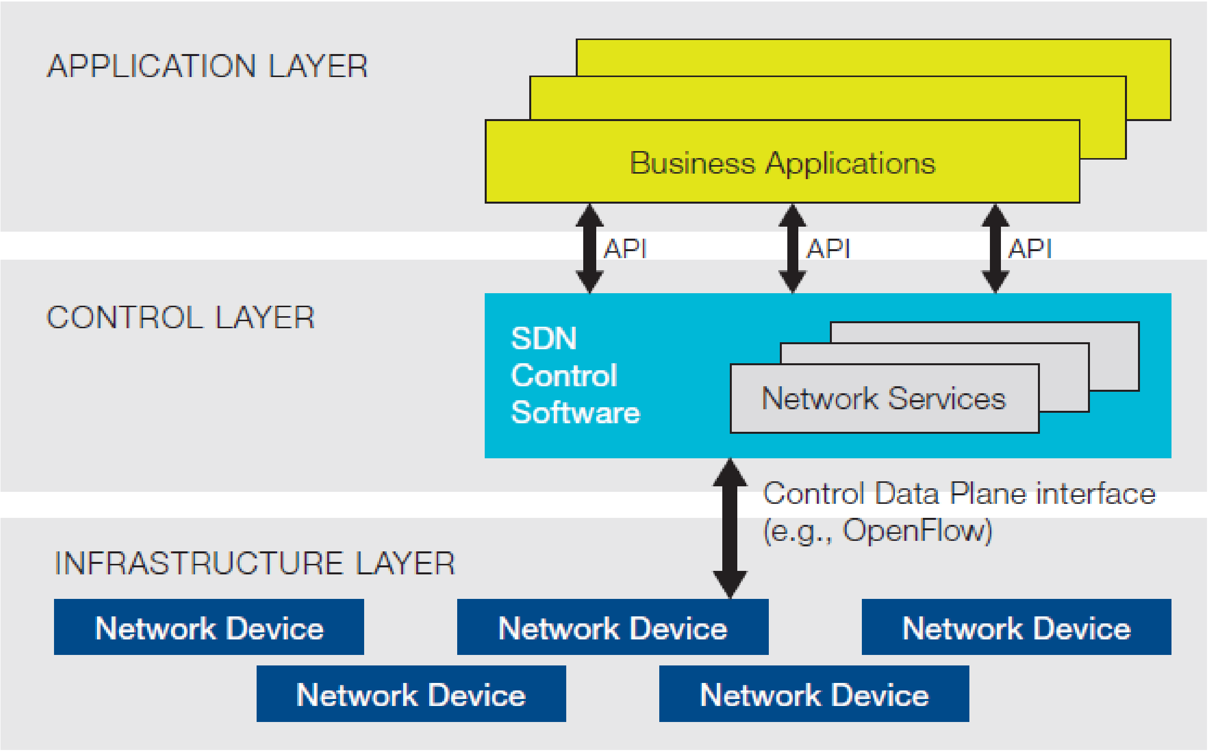


An example Ethane Deployment

Introducing Software Defined Networking

Software Defined Networking (SDN) is a new emerging network architecture which is entirely programmable and has a centralized controller that manages the forwarding of the packets. Thus, SDN is dynamic, cost effective, adaptable and manageable making it ideal for high bandwidth applications. Therefore, it is a new approach for networks providing them with real time intelligence, application integration and high levels of automation to prepare the network to meet the rigorous demands of cloud era.

Figure 1 depicts a logical view of SDN architecture. Network intelligence is centralized in the SDN controllers which act as network administrators known as hypervisors. Thus, the networks appears as a single, logical switch to the applications and policy engines. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.



Picture courtesy: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

The OpenFlow Protocol is the fundamental element for building SDN. SDN architecture is:

Directly programmable: Controller is programmed directly as it is decoupled from forwarding functions.

Agile: Administrators dynamically adjust network traffic flow according to changing needs by abstracting control from forwarding.

Programmatic configuration: SDN lets the network manager to configure, manage and optimize network resources using automated SDN programs which they can write themselves.

Open standards- based and vendor neutral: SDN uses open standard API known as OpenFlow. It simplifies network design and operation as the instructions are provided by SDN controllers rather than multiple vendor specific devices and protocols.

Applications of SDN

SDN and associated standards effectively address the needs of network operators in each segment of marketplace including:

The Enterprise

Cloud: Whether to support a private or hybrid cloud environment, SDN allows network resources to be allocated in an efficient way, enabling rapid provisioning of services.

Campus: SDN’s automated control supports the convergence of data voice and video anytime, anywhere by enforcing policies across both wired and wireless networks. Moreover, SDN extends its support to automated provisioning and management of network resources, determined by individual users and application requirements.

Data Center: SDN architecture facilitates network virtualization enabling hyper scalability in data center, automated VM migration and integration with storage, better server utilization, lower energy use, and bandwidth optimization.

The Carriers and Service providers: SDN’s automated, centralized control and provisioning model makes it easy to support multi-tenancy ensuring the network resources are deployed optimally.

Challenges of SDN:

- The big challenge currently is the lack of open standards and the fact there is no clear definitions as to how SDN is implemented.
- The number of rules for handling the packets is quite large.
- Controllers in SDN are not as fast as switches. This results in delay and overheads while processing the packets.
- APIs for controllers are till low-level and thus requires introduction of new programming languages.
- Security issues are the major challenges of SDN as if the controller is compromised, spoofed; the entire network can be reprogrammed. SDN is more prone to DOS attacks.

OpenFlow

Openflow is a protocol that allows communication between control plane and forwarding plane of SDN architecture. In other words, it enables extraction of the logic of switches and routers and handling over the control to logically centralized control software known as controller. Thus, it allows manipulation of forwarding plane as well.

Traditional network do not allow much of innovation as they are vendor-dependent and do not provide much flexibility in the network. Moreover, to support innovation it is required that the system provides high performance at low-cost implementations and also to isolate experimental traffic from the existing production traffic.

To solve this problem, OpenFlow switches have been introduced. These switches allow the flow tables to be programmed by the controller. Network administrator can now partition the traffic thus allowing the traffic of researchers, whose flow is maintained by the researcher, and also the present production traffic.

Requirement of Openflow

In traditional networks, network designers design different protocols based on different domains or to provide solution for a particular application. For example, they design one protocol for business related applications, one for data-intensive queries, and another for parallel processing and so on. To manage their network, network managers have to make desired configuration on switch-by-switch basis. They also invest in tools for managing the performance of the applications, tracking them and fixing them.

These all activities result in underutilization of network resources as the resources allocated to an application can’t be utilized even when it is not being used by the particular application. As already discussed, it incurs more cost in network components as in enterprise. The resources required are more to handle elephant flows as well as network valleys. The network becomes more complex due to these resources and will thus incur more labour costs. Moreover, the network becomes less manageable. The OpenFlow provides solution to tackle these problems.

How OpenFlow work?

An OpenFlow switch has three parts:

Flow table: It contains the entry to tell the switch how to process the packet flow

Secure channel: It facilitates connection of switch to a remote controller

OpenFlow protocol: It provides a standard way for the communication between switch and controller.

Flow table

Flow table has three basic fields:
Packet header: This defines the flow of data such as source address, destination address, source port etc.
Action: This section defines how to process the packet i.e. what to be done with the packet
Statistics: This section of flow table maintains track of number of packets and bytes of data that has been processed after the entry for that type of packet is entered in the flow table.

Actions:

Different actions can be taken to process the packets. The basic actions are:

1. Drop the packet. This step may be taken when security threat is detected.
2. Forward the packet to the controller. This action is taken when the packet doesn't match any of the flow table entry rule. This is the case when packet arrives for the first time.
3. Forward the packet to a specific port. This action is taken when switch already has an entry to handle that packet. It is the general routing of the packets.

Source address	Destination address	.....	Action	Statistics
*	10.0.0.43		Drop	43
192.168.1.4	*		Port 2	235
*	*		Controller	15

A sample open flow table

Working of OpenFlow network:

Whenever a packet arrives at a switch, the switch tries to match it with the pre-defined entry in the flow table. If the packet matches any entry, the corresponding action is taken and the statistics are updated. If it doesn't match, then the packet is encapsulated and forwarded to the controller via a secure channel. Controller processes and identifies the action to be taken based upon the programming and sends it back to the switch. It also adds a corresponding entry in the flow table to handle such type of packets. The packets are then processed by the switch as per the instructions of the controller. Whenever, further such packets arrive at the switch, the processing is done directly by the switch and the packet is not sent to the controller again. Each entry in flow table has and associated expiry time after which the entry is deleted from the table. Controller can also send instructions to remove the entry from the flow table.

Vendors marketing OpenFlow solutions:

Many vendors have recognized the future and importance of OpenFlow solutions and are working on it. Some of them are: BigSwitch, Nicra and NEC, Brocade Communications, Juniper Networks, Cisco, Extreme Networks, IBM, Hewlett-Packard, Dell.

Benefits of OpenFlow based SDN

- **Network Aware Intelligence:** OpenFlow controllers can understand the entire topology and can be programmed to handle workflows accordingly.
- **Centralized control:** As the whole control of network is in hand of controller, it eliminates the need of configuring each and every switch to adapt a change in network. And hence it enables faster deployment, configuration and updating entire network.
- **Reduced complexity:** As explained in above sections, traditional network require more labour as it requires configuration on per-switch basis. This manual task for network management increases complexity. OpenFlow provides automation by enabling dynamic control of network by controller.
- **Higher rate of innovation:** Traditional networks were vendor dependent and hence to implement any changes, researchers have to wait for the vendor's production cycle. OpenFlow has virtualized the network and has abstracted the individual services and thus providing the ability to introduce new services and changes in the network at a better speed. Now, researchers have to no longer wait for the vendor.
- **Better user experience:** OpenFlow based SDN provides dynamic adoption. For example, earlier user had to set the resolution setting for a video service which would sometimes result in delays and interruptions. OpenFlow based SDN provides a facility to automatically detect the settings and adjust accordingly based on the bandwidth available.

Due to changing user needs, dynamic networks have become a necessity. Traditional networks are complex and difficult to manage incurring more labour cost and maintenance cost. These networks are not as much secure as required in enterprise. Software defined networks provide a solutions to these problem by providing centralized and easily manageable control. Openflow is an open standard that supports SDN by providing an interface for interaction between forwarding plane and control plane. This makes the network programmable by the network managers and eliminated the requirement to depend upon the vendor for even minor changes. This enables network virtualization and thus allows research flow and existing production flow simultaneously, thus making the network more reliable. It allows to program different application as per the need of the network like load balancing, traffic engineering, power saving etc. This enables faster innovation, better user experience and reduced complexity.