**18ECE320T - Software Defined Networks**

**UNIT 4**

What is a Data Center?

A facility composed of networked computers, storage systems and computing infrastructure that organizations use to assemble, process, store and disseminate large amounts of data.

How do data centers work?

A data center facility, which enables an organization to collect its resources and infrastructure for data processing, storage and communications, includes the following:

* systems for storing, sharing, accessing and processing data across the organization;
* physical infrastructure for supporting data processing and data communications; and
* utilities such as cooling, electricity, network security access and uninterruptible power supplies (UPSes).

Why are data centers important?

Data centers enable organizations to concentrate on the following:

* IT and data processing personnel;
* computing and network connectivity infrastructure;
* protect proprietary systems and data;
* centralize IT and data processing employees, contractors and vendors;
* apply information security controls to proprietary systems and data;
* realize economies of scale by consolidating sensitive systems in one place.

What are the core components of data centers?

Facility. This includes the physical location with security access controls and sufficient square footage to house the data center's infrastructure and equipment.

Enterprise data storage. A modern data center houses an organization's data systems in a well-protected physical and storage infrastructure along with servers, storage subsystems, networking switches, routers, firewalls, cabling and physical racks.

Support infrastructure. This equipment provides the highest available sustainability related to uptime. Components of the support infrastructure include power distribution and supplemental power subsystems;electrical switching;UPSes;backup generators;ventilation and data center cooling systems, such as in-row cooling configurations and computer room air conditioners; and adequate provisioning for network carrier, or telecom, connectivity.

Operational staff. These employees are required to maintain and monitor IT and infrastructure equipment around the clock.

What are the types of data centers?

Enterprise data centers. These proprietary data centers are built and owned by organizations for their internal end users. They support the IT operations and critical applications of a single organization and can be located both on-site and off-site.

Managed services data centers. Managed by third parties, these data centers provide all aspects of data storage and computing services. Companies lease, instead of buy, the infrastructure and services.

Cloud-based data centers. These off-site distributed data centers are managed by third-party or public cloud providers, such as Amazon Web Services, Microsoft Azure or Google Cloud. Based on an infrastructure-as-a-service model, the leased infrastructure enables customers to provision a virtual data center within minutes.

Colocation data centers. These rental spaces inside colocation facilities are owned by third parties. The renting organization provides the hardware, and the data center provides and manages the infrastructure, including physical space, bandwidth, cooling and security systems. Colocation is appealing to organizations that want to avoid the large capital expenditures associated with building and maintaining their own data centers.

Edge data centers. These are smaller facilities that solve the latency problem by being geographically closer to the edge of the network and data sources.

Hyperscale data centers. Synonymous with large-scale providers, such as Amazon, Meta and Google, these hyperscale computing infrastructures maximize hardware density, while minimizing the cost of cooling and administrative overhead.

**SDN in the Data Center**

Data centers hold thousands, even tens of thousands, of physical servers. These data centers can be segregated into the following three categories:

• Private single-tenant. Individual organizations that maintain their own data centers belong in this category. The data center is for the private use of the organization, and there is only the one organization or tenant using the data center.

• Private multitenant. Organizations that provide specialized data center services on behalf of other client organizations belong in this category. IBM and EDS (now HP) are examples of companies that host such data centers. These centers are built and maintained by the organization providing the service, and multiple clients store data there, suggesting the term multitenant. These data centers are private because they offer their services contractually to specific clients

Public multitenant. Organizations that provide generalized data center services to any individual or organization belong in this category. Examples of companies that provide these services include Google and Amazon. These data centers offer their services to the public. Anybody, whether individuals or organizations, who wants to use these services may access them via the web.

Cloud Data centres: Data centers accessible through the Internet. These types of data centers are often referred to as residing in the cloud.

Public cloud: A service provider makes services available to the general public over the Internet. Examples of public cloud offerings include Microsoft Azure Services Platform and Amazon Elastic Compute Cloud.

Private cloud: A set of server and network resources is assigned to one tenant exclusively and protected behind a firewall specific to that tenant. The physical resources of the cloud are owned and maintained by the cloud provider, which may be a distinct entity from the tenant. The physical infrastructure may be managed using a product such as VMware’s vCloud. Amazon Web Services is an example of the way a private cloud may also be hosted by a third party (i.e., Amazon).

Hybrid Cloud: Part of the cloud runs on resources dedicated to a single tenant, but other parts reside on resources that are shared with other tenants. The shared resources may be acquired and released dynamically as demand grows and declines. Example is Verizon’s cloud bursting

**Data Center Demands**

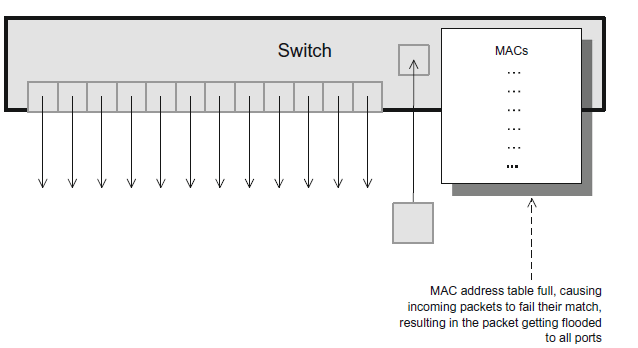
**1.Overcoming Current Network Limitations**

The dynamic nature and large number of VMs in the data center have placed demands on the capacity of network components. In particular, these areas include

* MAC address table size explosion
* number of VLANs,
* spanning tree.

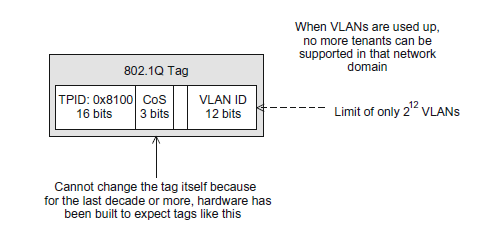
**a. MAC Address Explosion:**

In switches and routers, the device uses a MAC address table to quickly determine the port or interface out of which the device should forward the packet. For speed, this table is implemented in hardware. As such, it has a physical limit to its size. Networks had manageable limits on the maximum number of MAC addresses that would need to be in the MAC address table at any given time. Layer two switches are designed to handle the case of a MAC table miss by flooding the frame out all ports except the one on which it arrived, as shown in Figure. From the response from the destination, the switch is able to learn the port on which that MAC address is located and populates its MAC table accordingly. This scheme works well unless the MAC table is full, in which case it cannot learn the address.



**b. Number of VLANs**

When the IEEE 802.1 working group created the 802.1Q extension to the definition of local area networks, they did not anticipate that there would be a need for more than 12 bits to hold potential VLAN IDs. The IEEE 802.1Q tag for VLANs is shown in Figure . The tag depicted in Figure supports 212 (4096) VLANs.When data centers continued to expand, however, especially with multi-tenancy and server virtualization,the number of VLANs required could easily exceed 4096.An upshot of the limit of 4096 VLANs has been an increase in the use of MPLS since it does not have limitation on number of MPLS tags. It is likely,, that MPLS will see more use in data centers



**c. Spanning Tree**

Bridges were built as transparent devices capable of forwarding packets from one segment to another without explicit configuration of forwarding tables by an operator. The bridges learned forwarding tables by observing the traffic being forwarded through them. They create a spanning tree, which enforces a loop-free hierarchical structure on the network in situations where physical loops do exist. This spanning tree was then calculated using the Spanning Tree Protocol (STP), The process of determining this spanning tree is called convergence.The fluidity of data center virtualization has increased the frequency of changes and disruptions, thus requiring reconvergence to occur more often, adding to the inefficiency of STP in the data center. Data centers need more cross-sectional bandwidth i.e using the most efficient path between any two nodes without imposing an artificial hierarchy in the traffic patterns.

**2.Adding, Moving, and Deleting Resources**

Networks need to adapt in order to keep pace with the virtualization capabilities of servers and storage. Speed and automation are of critical importance when it comes to handling the rapid changes demanded by virtualized servers and storage. These changes need to have the ability to be automated so that changes that must happen immediately can take place without human intervention. With SDN one can use the foreknowledge that a new service is about to be initiated and proactively allocate the network capacity it will require.

**3.Failure Recovery**

The size and scale of data centers today make recovery from failure a complex task, and the ramifications of poor recovery decisions are only magnified as scale grows. Determinism, predictability, and optimal re-configuration are among the most important recovery-related considerations.

It is desirable that the network moves to a known state, given a particular failure. Distributed protocols make this difficult to do. A complete view of the network as is provided by SDN is required to make the recovery process yield the best result.

**4. Multitenancy**

Data center consolidation has resulted in more and more clients occupying the same set of servers, storage, and network. The challenge is to keep those individual clients separated and insulated from each other and to utilize network bandwidth efficiently.

In a large multi tenant environment, it is necessary to keep separate the resources belonging to each client. For servers this could mean not mixing clients’ virtual machines on the same physical server. For networks it could mean segregation of traffic using a technology that ensures that packets from two distinct tenants remain insulated from one another. This is needed not only for the obvious security reasons but also for QoS and other service guarantees.

**5. Traffic Engineering and Path Efficiency**

It is imperative to optimally utilize the capacity of the network. To understand traffic loads and take the appropriate action, the traffic data must be monitored and measured.

One of the reasons for the increasing attention on traffic engineering and path efficiency in the data center has been the rise of East-West traffic relative to North-South traffic.

Example: Bringing up newsfeed on Facebook page: Here East-West traffic is as large as the North-South traffic.

Traffic types as follows:

East-West traffic is composed of packets sent by one host in a data center to another host in that same data center.

North-South traffic is traffic entering (leaving) the data center from (to) the outside world.

**Tunneling Technologies for the Data Center Network**Tunneling protocols are based on the notion of encapsulating an entire layer two - MAC frame inside an IP packet. This is known as MAC-in-IP tunneling.The hosts involved in communicating with each other are unaware that there is anything other than a traditional physical network between them. The hosts construct and send packets in exactly the same manner as they would had there been no network virtualization involved. In this way, network virtualization resembles server virtualization, where hosts are unaware that they are actually running in a virtual machine environment. Hypervisor-based tunneling technologies are employed to achieve this virtualization of the network

**MAC-in-IP tunneling concept**

Encapsulation used in Tunnelling makes virtualized networks possible.

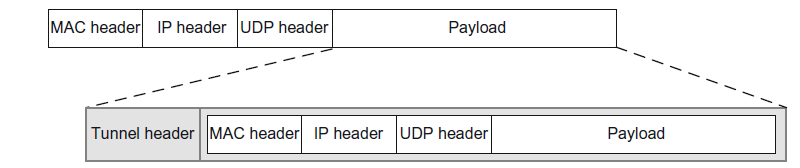
When a packet enters the edge of the virtual network at the source (virtual tunnel endpoint (VTEP)), the networking device (usually the hypervisor) will take the packet in its entirety and encapsulate it within another frame.

The hypervisor then takes this encapsulated packet and, based on information programmed by the controller, sends it to the destination’s VTEP. This VTEP decapsulates the packet and forwards it to the destination host.

As the encapsulated packet is sent across the physical infrastructure, it is being sent from the source’s VTEP to the destination’s VTEP. Consequently, the IP addresses are those of the source and destination VTEP.

This tunneling mechanism is referred to as MAC-in-IP tunneling

Proprietary : Cisco offers VXLAN , Microsoft uses NVGRE , and Nicira’s is called STT



**Tunneling methods**

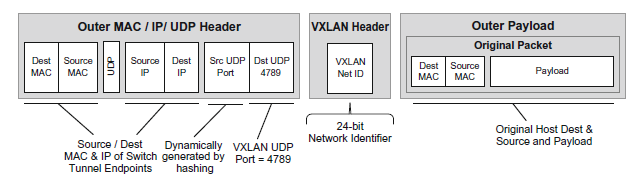
* Virtual eXtensible Local Area Network (VXLAN)
* Network Virtualization using Generic Routing Encapsulation (NVGRE)
* Stateless Transport Tunneling (STT)

**Virtual eXtensible Local Area Network**

Developed primarily by VMware and Cisco

Main characteristics :

VXLAN utilizes MAC-in-IP tunneling. Each virtual network or overlay is called a VXLAN segment. VXLAN segments are identified by a 24-bit segment ID, allowing for up to 224 (approximately 16 million) segments. VXLAN tunnels are stateless. VXLAN segment endpoints are the switches that perform the encapsulation and are called virtual tunnel endpoints (VTEPs). VXLAN packets are unicast between the two VTEPs and use UDP-over-IP packet formats. It is UDP based. The UDP port number for VXLAN is 4789.



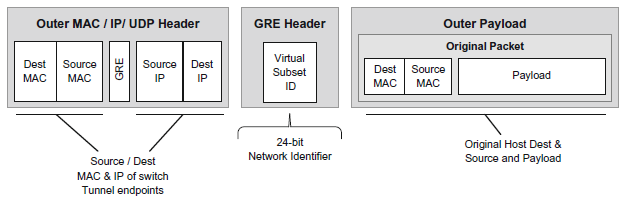
* Figure illustrates the format of a VXLAN packet.
* The outer header contains the MAC and IP addresses appropriate for sending a unicast packet to the destination switch, acting as a virtual tunnel endpoint.
* The VXLAN header follows the outer header and contains a VXLAN network identifier of 24 bits in length, sufficient for about 16 million networks.
* Advantage of VXLAN: Assists load balancing within the network

**Network Virtualization using Generic Routing Encapsulation(NVGRE)**

Developed primarily by Microsoft with contributions from HP, Dell, Intel

Main characteristics of NVGRE are:

NVGRE utilizes MAC-in-IP tunneling.Each virtual network or overlay is called a virtual layer two network. NVGRE virtual networks are identified by a 24-bit virtual subnet identifier, allowing for up to 224 (16 million) networks.NVGRE tunnels, like GRE tunnels, are stateless.NVGRE packets are unicast between the two NVGRE end points, each running on a switch.NVGRE utilizes the header format specified by the GRE standard.



* Figure shows the format of an NVGRE packet.
* The outer header contains the MAC and IP addresses appropriate for sending a unicast packet to the destination switch, acting as a virtual tunnel endpoint, just like VXLAN.
* Recall that for VXLAN the IP protocol value was UDP. For NVGRE the IP protocol value is 0x2F, which means GRE.
* GRE is a separate and independent IP protocol in the same class as TCP or UDP. Consequently, as you can see in the diagram, there are no source and destination TCP or UDP ports.
* The NVGRE header follows the outer header and contains a NVGRE subnet identifier of 24 bits in length, sufficient for about 16 million networks.

**Stateless Transport Tunneling**

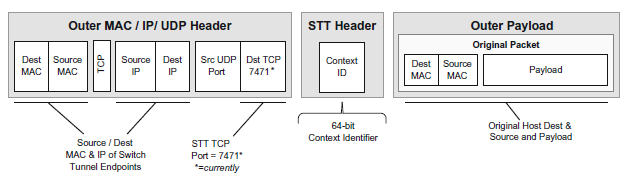
Stateless Transport Tunneling (STT) - major sponsor was originally Nicira.

Some of the main characteristics of STT are:

STT utilizes MAC-in-IP tunneling. The general idea of a virtual network exists in STT but is enclosed in a more general identifier called a context ID. STT context IDs are 64 bits, allowing for a much larger number of virtual networks and a broader range of service models.

STT attempts to achieve performance gains over NVGRE and VXLAN by leveraging the TCP Segmentation Offload (TSO) found in the network interface cards (NICs) of many servers. TSO is a mechanism implemented on server NICs that allows large packets of data to be sent from the server to the NIC in a single send request, thus reducing the overhead associated with multiple smaller requests.

STT, as the name implies, is stateless. STT packets are unicast between tunnel end points, utilizing TCP in the stateless manner associated with TSO. This means that it does not use the typical TCP windowing scheme, which requires state for TCP synchronization and flow control.



* Figure shows the format of an STT packet.
* The outer header contains the MAC and IP addresses appropriate for sending a unicast packet to the destination switch, acting as a VTEP.
* For VXLAN, the IP protocol value was UDP, and for NVGRE the IP protocol value was GRE. For STT, the IP protocol is TCP.
* The TCP port for STT is 7471.
* The STT header follows the outer header and contains an STT context identifier of 64 bits in length, which can be subdivided and used for multiple purposes;
* However that is done, there is ample space for as many virtual networks as required.

**Path Technologies in the Data Center**

With the size and demands of data centers, it is imperative that all physical network links forming the data center’s network infrastructure be utilized to their full capacity. Layer three networks require intelligent routing of packets as they traverse the physical network. Path-related technologies provide some of the intelligence required to make the most efficient use of the network and its interconnecting links

**1. General Multipath Routing Issues:**

There will be multiple routers connected in some manners to provide redundant links for failover. Multipath routing makes use of multiple routes in order to balance traffic across a number of potential paths.

Issues that must be considered in any multipath scheme are:

* The potential for out-of-order delivery (OOOD) of packets that take different paths and must be reordered by the recipient,
* The potential for maximum packet size differences on links within the different paths, causing issues for certain protocols such as TCP and its path MTU discovery.

**2. Multiple Spanning Tree Protocol**

The Multiple Spanning Tree Protocol (MSTP) was introduced to achieve better network link utilization with spanning tree technology when there are multiple VLANs present.

Each VLAN would operate under its own spanning tree. The improved use of the links was to have one VLAN’s spanning tree use unused links from another VLAN, when reasonable to do so. MSTP was originally introduced as IEEE 802.1s.

But in MSTP, it was necessary to have a large number of VLANs in order to achieve a well-distributed utilization level across the network links.

**3. Shortest Path Bridging**

IEEE 802.1aq is the Shortest Path Bridging (SPB) standard, and its goal is to enable the use of multiple paths within a layer two network. Thus, SPB allows all links in the layer two domain to be active.

SPB is a link state protocol, which means that devices have awareness of the topology around them and are able to make forwarding decisions by calculating the best path to the destination.

It uses the Intermediate System to Intermediate System (IS-IS) routing protocol to discover and advertise network topology and to determine the paths to all other bridges in its domain.

SPB accomplishes its goals using encapsulation at the edge of the network. This encapsulation can be either MAC-in-MAC (IEEE 802.1ah) or Q-in-Q (IEEE 802.1ad).

**4.Equal-Cost Multipath**

In large networks optimal path computation is very important. In the data center, there is a general routing strategy called equal-cost multipath (ECMP) that is applicable. Multipath routing is a feature that is explicitly allowed in both OSPF and IS-IS. OSPF and IS-IS are modern link-state protocols used for calculating optimal routes.The notion is that when more than one equal-cost path exists to a destination, these multiple paths can be computed by a shortest-path algorithm and exposed to the packet-forwarding logic. At that point some load-balancing scheme must be used to choose between the multiple available paths. Because several routing protocols can derive the multiple paths and there are many ways in which to load-balance across the multiple paths, ECMP is more of a routing strategy than a specific technology.

**5. SDN and Shortest-Path Complexity**

The advantages of SDN in path computation:

* It has a more stable, global view of the network,
* It can take more factors into consideration, including current bandwidth loads
* The computation can be performed on the higher-capacity memory and processor of a server.

Inspite of these advantages, Shortest-path remains a fundamentally difficult problem that grows harder very fast as the number of switches and links scales. The well-known Dijkstra’s algorithm for shortest-path remains in wide use (it is used in both OSPF and IS-IS), and SDN is unlikely to alter that.

**SDN Use Cases in the Data Center**

There are various type of SDN

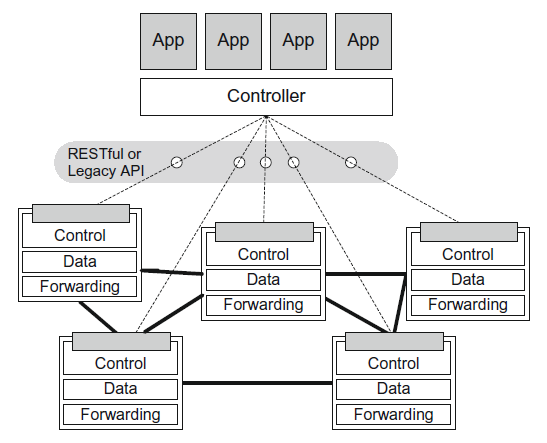
1. The original SDN, which we call Open SDN;

2.SDN via APIs;

3.SDN via hypervisor-based overlays

**SDN via Existing APIs**

* Command Line Interface (CLI) and Simple Network Management Protocol (SNMP): Traditional methods to set configuration parameters on devices
* Fig shows a controller communicating with devices via a proprietary API.
* New method: RESTful API. REST (*Representational State Transfer)*  has become the dominant method of making API calls across networks.
* REST uses *HyperText Transfer Protocol* (HTTP), the protocol commonly used to pass web traffic.
* RESTful APIs are simple and extensible and have the advantage of using a standard TCP port and thus require no special firewall configuration to permit the API calls to pass through firewalls

****

Advantage of SDN via APIs

* Since this uses legacy management interfaces, it therefore works with legacy switches. Thus, this solution does not require upgrading to OpenFlow-enabled switches.
* It allows for some improvement in agility and automation. These APIs also make it easier to write software such as orchestration tools that can respond quickly and automatically to changes in the network (e.g., the movement of a virtual machine in a data center).
* It allows for some amount of centralized control of the devices in the network. Therefore, it is possible to build an SDN solution using the provided APIs on the distributed network devices.
* Potential for increased openness: Although the individual interfaces may be proprietary to individual vendors, when they are exposed to the applications, they are made open for exploitation by applications. The degree of openness will vary from one NEM to another.

Disadvantages of SDN via APIs

* The network programmer needs to interact directly with each switch. It does not provide an abstract, network-wide view to the programmer.
* since there is still a control plane operating on each switch, the controller and, more important, the programmer developing applications on top of that controller must synchronize with what the distributed control plane is doing.
* The SDN via existing APIs approach relies on the same complicated, expensive switches as before.
* SDN-like software applications using this type of API-based approach will onlyworkwith devices from that specific vendor or a small group of compatible vendors.

**SDN via hypervisor-based overlays**

Well suited to environments such as data centers already running compute and storage virtualization software for their servers. The virtual network traffic runs above the physical network infrastructure. The hypervisors inject traffic into the virtual network and receive traffic from it.

The traffic of the virtual networks is passed through those physical devices, but the endpoints are unaware of the details of the physical topology, the way routing occurs, or other basic network functions. Since these virtual networks exist above the physical infrastructure, they can be controlled entirely by the devices at the very edge of the network.

In data centers, these would typically be the hypervisorsof the VMs that are running on each server.DN via hypervisor-based overlay networks

**Comparison of alternatives in addressing Daa center needs**

**1.Overcoming Current Network Limitations**

***SDN via Overlays***

* *SDN via overlays,* host **MAC addresses** are hidden within the encapsulated frame. The only MAC addresses visible through the physical network are the MAC addresses of the tunnel endpoints, which are at the hypervisors.
  + e.g. For eight VMs per hypervisor, the total number of MAC addresses is reduced by a factor of eight.
* For the issue of **VLAN exhaustion**, *SDN via overlays is a good solution because t*he new mechanism for multitenancy is tunnels, not VLANs. All traffic is tunneled and VLANs are not required for supporting the isolation of multiple tenants. The number of tunneled networks or segments can be 16 million or greater using VXLAN,NVGRE, or STT tunneling technologies.
* It addresses **agility and automation** needs because it is implemented in software, and these virtual networks can be constructed and taken down in a fraction of the time that would be required to change the physical network infrastructure.
* For the issue of **spanning tree convergence**, SDN via overlays does not address issues related to the physical infrastructure.
* They fail to address traffic prioritization and efficiency in the physical infrastructure,

***Open SDN***

* The SDN controller can create tunnels as required at what will become the tunnel endpoints, and then Open Flow rules are used to push traffic from hosts into the appropriate tunnel. SDN devices can be built that derive these benefits from tunneling but with the performance gain of hardware.

***SDN via APIs***

* Adding SDN APIs to networking devices does not directly address network limitations.

**2. Adding, Moving, and Changing Resources**

***SDN via Overlays***

* Overlays are the simplest way to provide the automation and agility required to support frequent adds, moves, deletes, and changes.
* SDN does not deal with the physical infrastructure at all. The networking devices that it manipulates are most often the virtual switches that run in the hypervisors.
* Furthermore, the network changes required to accomplish the task are simple and confined to the construction and deletion of virtual networks, which are carried within tunnels that are created expressly for that purpose.
* These virtual networks are easily manipulated via software

***Open SDN***

* If OpenSDN is being used to create tunnels and virtual networks, the task is to create the overlay tunnels as required and to use Open Flow rules to push packets into the appropriate tunnels.
* In addition to the advantages of virtual networks via tunnels, Open SDN offers the ability to change the configuration and operation of the physical network below— referred to as the *underlay*.

***SDN via APIs***

* APIs provide a programmatic framework for automating tasks that would otherwise require manual intervention. The ability to have a controller that is aware of server virtualization changes and can make changes to the network in response is a definite advantage.

**3. Failure Recovery**

***SDN via Overlays***

* Overlay technology does not deal with the physical network below it. So it does not improve the failure recovery methods in the data center. If there are failures in the physical infrastructure, those must be dealt with via the mechanisms already in place, apart from overlays.

***Open SDN***

* One of the benefits of OpenSDN is that with a centralized controller the whole network topology is known and routing (or, in this case, rerouting) decisions can be made that are consistent and predictable.

***SDN via APIs***

* The presence of improved APIs on network devices and a controller to use those APIs to automatically update the device provides some improvement in failure recovery.
* If the APIs are giving only a slightly improved access to traditional configuration parameters, clearly SDN via APIs provides little value in this area. However, if those APIs are accompanied by the devices ceding their path decision making functionality to the SDN controller, the APIs can furnish more value.

**4.Multitenancy**

* The traditional way of achieving the separation required by this sharing of network bandwidth has been through the use of VLANs.
* ***SDN via Overlays***
* Overlay technology resolves the multitenancy issue by its very nature through the creation of virtual networks that run on top of the physical network. These virtual networks substitute for VLANs as the means of providing traffic separation and isolation. In overlay technologies, VLANs are only relevant within a single tenant. For each tenant, there is still the 4096 VLAN limit, but that seems to currently suffice for a single tenant’s traffic.
* ***Open SDN***
* Open SDN can implement network virtualization using layer three tunnel-based. Other types of encapsulation (e.g., MAC-in-MAC, Q-in-Q) can also be employed to provide layer two tunneling, which can provide multiplicative increases in the number of tenants.
* ***SDN via APIs***
* Without some complementary changes to devices, such as providing virtual networks, SDN via APIs does not address the issue of multitenancy.

**5.Traffic Engineering and Path Efficiency**

**SDN via Overlays**

In a similar manner to the issue of failure recovery, SDN via overlays does not have much to contribute in this area due to the fact that it does not attempt to affect the physical network infrastructure.

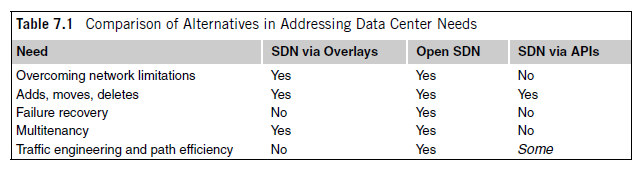
Traffic loads as they pass from link to link across the network are not a part of the discussion concerning traffic that is tunneled across the top of the physical devices and interfaces. Thus, SDN via overlays is dependent on existing underlying network technology to address these types of issues.

**Open SDN**

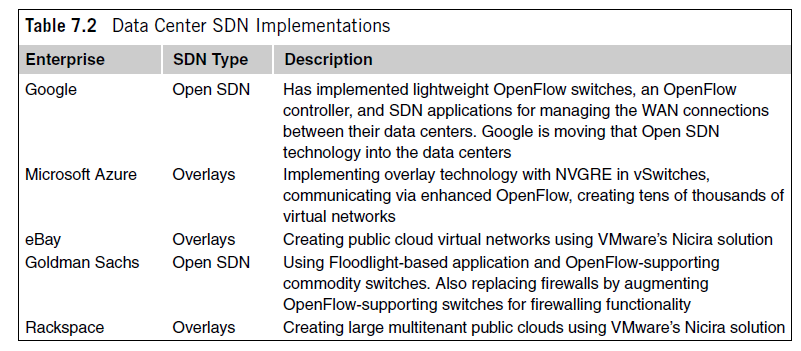
Open SDN has major advantages here in the areas of centralized control and having complete control of the network devices. Open SDN has centralized control with a view of the full network and can make decisions predictably and optimally. It also controls the individual forwarding tables in the devices and, as such, maintains direct control over routing and forwarding decisions

**SDN via APIs**

It is worth noting that policy-based routing (PBR) has the ability to direct packets across paths at a fairly granular level. In theory one could combine current traffic-monitoring tools with PBR and use current SNMP or CLI APIs to accomplish the sort of traffic engineering we discuss here. RSVP and MPLS-TE are examples of traffic engineering protocols that may be configured via API calls to the device’s control plane.



**Real-World Data Center Implementations**

****

**SDN Application and Use Case**

**SDN in Service Provider (SP) and Carrier Networks**

Service provider (SP) and carrier networks are wide area in nature and are sometimes referred to as backhaul networks, since they aggregate edge networks, carrying huge amounts of data and voice traffic across geographies, often on behalf of telecommunications and Internet service providers (ISPs).

Examples of service providers and carriers are Verizon, AT&T, Sprint, Vodafone, and China Mobile.

SPs are responsible for taking network traffic from one source, passing it throughout the SP’s network, and forwarding it out the remote network edge to the destination.

Thus, the packets themselves must cross at least two boundaries.

When more than one SP must be traversed, the number of boundaries crossed increases.

Traffic coming into the service provider’s network is typically marked with specific tags for VLAN and priority.

The boundaries that traffic must cross are often referred to as customer edge (CE) and provider edge (PE).

The network to-network interface (NNI) is the boundary between two SPs.

Since the NNI is an important point for policy enforcement, it is important that policy be easily configurable at these boundaries.

(See Figure) The original packet as it emanates from the source device is unencapsulated.

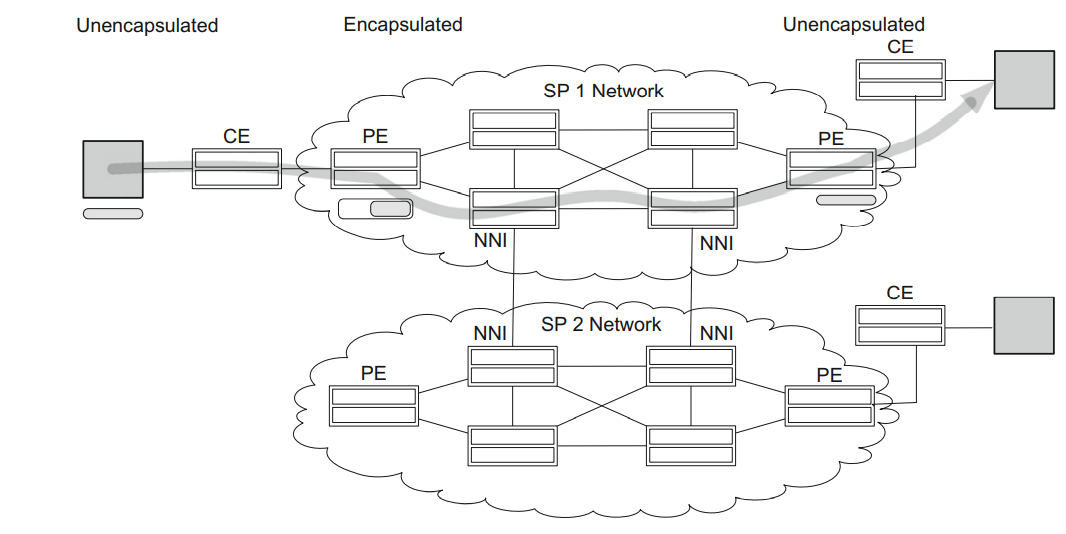
The packet may acquire a VLAN tag as it passes through the CE (probably provided by the ISP).

When the packet passes through the PE, it may be encapsulated using a technology such as PBB, or it may be tagged with another VLAN or MPLS tag.

When the packet exits the provider network and passes through the other PE on the right, it is correspondingly either decapsulated or the tag is popped and it is then passed into the destination CE network.

If the customer packets were directed to a destination on the SP2 network, they would traverse such an NNI boundary.

Policies related to business agreements between the SP1 and SP2 service providers would be enforced at that interface



**SDN Applied to SP and Carrier Networks**

Monetization. This refers to the ability to make or save money by using specific techniques and tools.

SDN is promoted as a way for providers and carriers to monetize their investments in networking equipment by increasing efficiency, reducing the overhead of management, and rapidly adapting to changes in business policy and relationships.

Ways in which SDN can help improve monetization for providers and carriers are

(1) bandwidth management,

(2) CAPEX and OPEX savings, and

(3) policy enforcement at the PE and NNI boundaries.

SDN exhibits great agility and the ability to maximize the use of existing links using traffic engineering and centralized, network-wide awareness of state, allows changes to be made easily and with improved ability to do so with minimal service interruption.

This facilitates the profitable use of bandwidth as well as the ability to adapt the network to changing requirements related to customer needs and service-level agreements (SLAs).

**Ways in which SDN can Reduce costs**

* First, there are **CAPEX savings**. The cost of white-box SDN devices is appreciably lower than the cost of comparable non-SDN equipment.
* This may be due to bill-of-materials (BOM) reduction as well as the simple fact that the white-box vendors are accustomed to a lower-margin business model.
* The BOM reductions derive from savings such as reduced memory and CPU costs due to the removal from the device of so much CPU- and memory-intensive control software.
* Second, there are reduced **OPEX costs**, which come in the form of reduced administrative loads related to the management and configuration of the devices.

**SDN in Campus Networks**

* Campus networks are a collection of LANs in a concentrated geographical area.
* Usually the networking equipment and communications links belong to the owner of the campus.
* This may be a university, a private enterprise, or a government office, among other entities.
* Campus end users can connect through wireless access points (APs) or through wired links.
* They can connect using desktop computers, laptop computers, shared computers, or mobile devices such as tablets and smartphones.
* The devices with which they connect to the network may be owned by their organization or by individuals.
* Furthermore, those individually owned devices may be running some form of access software from the IT department, or the devices may be completely independent.

**Networking Requirements pertaining to Campus Networks**

**(1) Differentiated levels of access**

Various types of users in the campus will require different levels of access.

* Day guests should have access to a limited set of services, such as the Internet.
* More permanent guests may obtain access to more services.
* Employees should receive access based on the category into which they fall, such as executives or IT staff.

These differentiated levels of access can be in the form of access control as well as their quality of service, such as traffic prioritization and bandwidth limits.

(2) Bring your own device (BYOD),

(3) access control and security,

(4) service discovery,

(5) end-user firewalls.

**Dangers of campus networks -** Possibility of infected devices introducing unwanted threats to the network and is magnified by the presence of BYOD devices on the network. These may take the form of malicious applications such as port scanners, which probe the network looking for vulnerable devices. End-user firewalls are needed to protect against such threats.

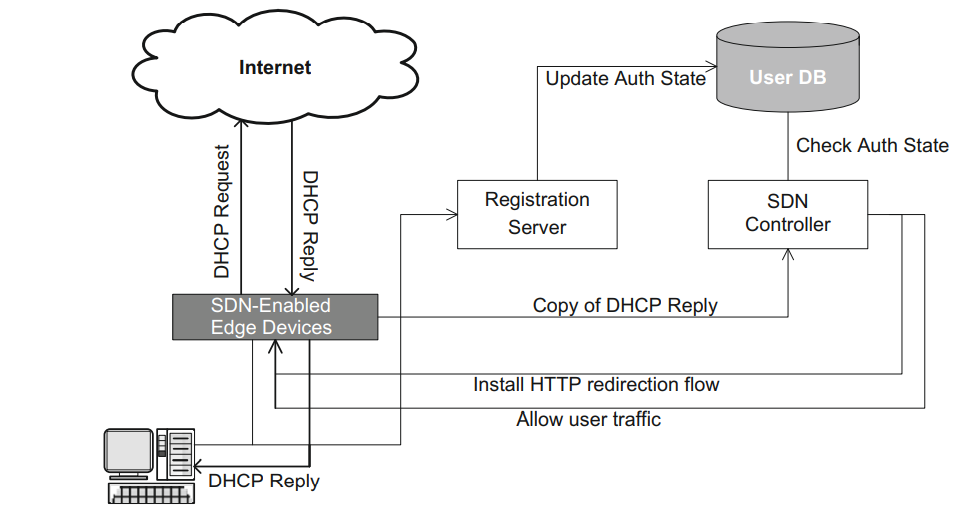
**SDN on Campus: Device and User Security**

Technological trends such as BYOD, access control, and security can also be addressed by SDN technology. For example, one of the requirements for registering users’ BYOD systems and guests involves the use of a **captive portal.**

This is the mechanism by which a user’s browser request gets redirected to another destination website. This other website can be for the purpose of device registration or guest access. Captive portals are traditionally implemented by encoding the redirection logic into the switch firmware or by in-line appliances. Configuring which users need to be authenticated via the captive portal would entail configuring all these devices where a user could enter the network.

**SDN solution for a captive portal: SDN-based captive portal-based application**

* The user connects to the network and attempts to establish network connectivity.
* No access rules are in place for this user, so the SDN controller is notified.
* The SDN controller programs flows in the edge device, which will cause the user’s HTTP traffic to be redirected to a captive portal.
* The user is redirected to the captive portal and engages in the appropriate exchange to gain access to the network.
* Once the captive portal exchange is complete, the SDN controller is notified to set up the user’s access rules appropriately.
* The user and/or BYOD device now has the appropriate level of access

****

The network edge device is initially programmed to route ARP, DNS, and DHCP requests to the appropriate server. In the figure, the end user connects to the network and makes a DHCP request to obtain an IP address. When the DHCP reply is returned to the user, a copy is sent to the SDN controller. Using the end-user MAC address as a lookup key, the controller consults the database of users.

If the user device is currently registered, it is allowed into the network. If it is not, then OpenFlow rules are programmed to forward that user’s HTTP traffic to the controller. When the unauthenticated user’s HTTP traffic is received at the controller, that web session is redirected to the captive portal web server. After completing the user authentication or device registration, the controller updates the user’s flow(s) so that the packets will be allowed into the network.

**SDN in Mobile Networks**

Mobile networking vendors, such as AT&T, Verizon, and Sprint, compete for customers to attach to their networks.

When mobile customers use traditional WiFi hotspots to connect to the Internet, those mobile vendors effectively lose control of their customers. This is because the users’ traffic enters the Internet directly from the hotspot. Since this completely circumvents the mobile vendor’s network, the vendor is not even aware of the volume of traffic that the user sends and receives and certainly cannot enforce any policy on that connection.

When customer traffic circumvents the mobile provider’s network, the provider loses a revenue-generating opportunity. Thus, the mobile provider is interested in a solution that allows its customers to access its networks via public WiFi hotspots without the provider losing control of and visibility to its customers’ traffic. The owner of such hotspots may want for multiple vendors to share the WiFi resource offered by the hotspot.

Mobile vendors interested in gaining access to users who are attaching to the Internet via WiFi hotspots require a mechanism to control their users’ traffic. Control in this context may simply mean being able to measure how much traffic that user generates. It may mean the application of some policy regarding QoS. It may mean diverting the user traffic before it enters the public Internet and redirecting that traffic through its own network.

**SDN Applied to Mobile Networks**

SDN technology can play a role in such a scheme in the following ways:

• Captive portals - It requires allowing users to register for access based on their mobile credentials. Once valid credentials are processed, the user is granted appropriate levels of access.

• Tunneling back to the mobile network - establishment of tunnels from the user’s location back to the mobile vendor’s network. By programming SDN flows appropriately, that user’s traffic would be forwarded into a tunnel and diverted to the mobile vendor’s network.

• Application of policy - Usage charges could be applied by the mobile provider. In addition to charging for this traffic, other users pecific policies could be enforced. Such policies could be applied at WiFi hotspots where the user attaches to the network. SDN-enabled access points can receive policy, either from the controller of the mobile vendor or from a controller on the premises.

**SDN and Open Flow in Mobile networks**

In Figure, the customers on the left want to access the Internet, and each set has a different carrier through which they gain network access.

Connecting through an OpenFlow-enabled wireless hotspot, they are directed through a broker that acts as an OpenFlow controller.

Based on their carrier, they are directed to the Internet in various ways, depending on the level of service and the mechanism set up by the carrier.

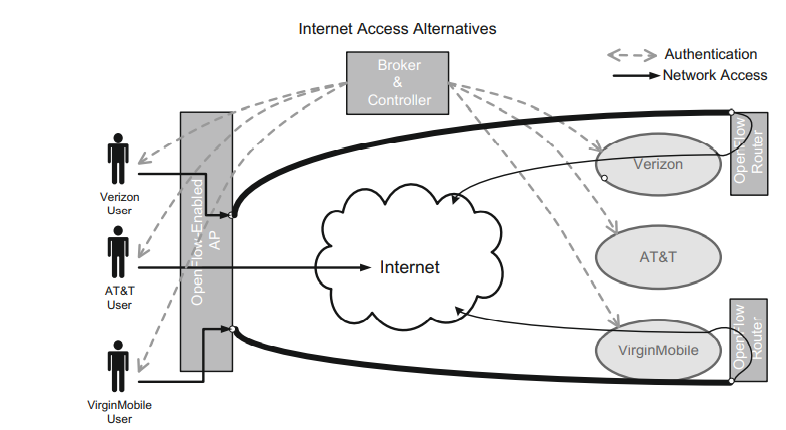
In the example, AT&T users gain access directly to the Internet, whereas Verizon and Virgin Mobile users access the Internet through being directed by OpenFlow through a tunnel to the carrier’s network.

Both tunnels start in the OpenFlow-enabled AP. One tunnel ends in an OpenFlow-enabled router belonging to Verizon, the other in an OpenFlow-enabled router belonging to Virgin Mobile.

These two routers then redirect their respective customers’ traffic back into the public Internet. In so doing, the customers gain the Internet access they desire, and two of the mobile providers achieve the WiFi offload they need while maintaining visibility and control over their users’ traffic.

To facilitate such a system, there is presumably a business relationship between the providers and the hotspot owner whereby the hotspot owner is compensated for allowing the three providers’ users to access the hotspot.

There would likely be a higher fee charged Verizon and Virgin Mobile for the service that allows them to retain control of their customers’ traffic



**Reactive SDN Applications**

The communication between the switch and the controller will scale with the number of new flows injected into the network. The switches may often have relatively few flow entries in their flow tables, since the flow timers are set to match the expected duration of each flow in order to keep the flow table size small. This results in the switch frequently receiving packets that match no rules.

Those packets are encapsulated in PACKET\_IN messages and forwarded to the controller and thence to an application. The application inspects the packet and determines its disposition. The outcome is usually to program a new flow entry in the switch(es) so that the next time this type of packet arrives, it can be handled locally by the switch itself. The application will often program multiple switches at the same time so that each switch along the path of the flow will have a consistent set of flow entries for that flow.The original packet will often be returned to the switch so that the switch can handle it via the newly installed flow entry. The kind of applications that naturally lend themselves to the reactive model are those that need to see and respond to new flows being created.

Examples of such applications include per-user firewalling and security applications that need to identify and process new flows.Reactive programming may be more vulnerable to service disruption if connectivity to the controller is lost.

**Reactive Applications – Listener Approach**

With the Java APIs one can register a listener and then receive callbacks from the controller when packets arrive.

These callbacks come with passed arguments including the packet and associated metadata, such as which switch forwarded the packet and the port on which the packet arrived.

Consequently, reactive applications tend to be written in the native language of the controller, which, is Java, C or Ruby.

Reactive applications have the ability to register listeners, which are able to receive notifications from the controller when certain events occur.

**Reactive Applications – Listeners**

Some important listeners available in the popular Floodlight and Beacon controller packages are

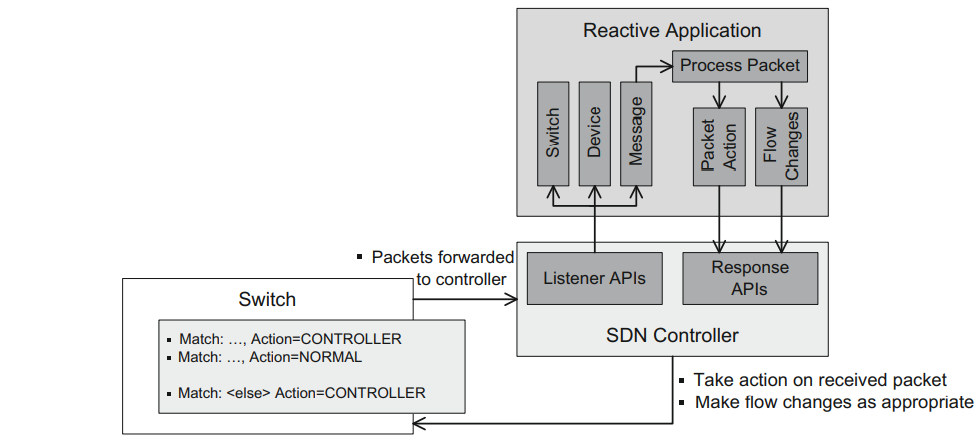
Switch Listener. Switch listeners receive notifications whenever a switch is added or removed or has a change in port status.

Device Listener. Device listeners are notified whenever a device (an end-user node) has been added, removed, or moved (attached to another switch) or has changed its IP address or VLAN membership.

Message Listener. Message listeners get notifications whenever a packet has been received by the controller. The application then has a chance to examine it and take appropriate action.

These listeners allow the SDN application to react to events that occur in the network and to take action based on those events.

**Reactive Applications – Actions**

* When a reactive SDN application is informed of an event, such as a packet that has been forwarded to the controller, a change of port state, or the entrance of a new switch or device into the network, the application has a chance to take some type of action.
* The most frequent event coming into the application would normally be a packet arriving at the controller from a switch, resulting in an action.
* Such actions include:
* **Packet-specific actions**. The controller can tell the switch to drop the packet, to flood the packet, to send the packet out a specific port, or to forward the packet through the NORMAL non-OpenFlow packet-processing pipeline.
* **Flow-specific actions.** The controller can program new flow entries on the switch, intended to allow the switch to handle certain future packets locally without requiring intervention by the controller.
* 
* Controller has a listener interface that allows the application to provide listeners for switch, device, and message (incoming packet) events.
* Typically, a reactive application will have a module to handle packets incoming to the controller that have been forwarded through the message listener.
* This packet processing can then act on the packet.
* Typical actions include returning the request to the switch, telling it what to do with the packet (e.g., forward out a specific port, forward NORMAL or drop the packet).
* Other actions taken by the application can involve setting flows on the switch in response to the received packet, which will inform the switch what to do the next time it sees a packet of this nature.
* For reactive applications, the last flow entry will normally be programmed to match any packet and to direct the switch to forward that otherwise unmatched packet to the controller.
* This methodology is precisely what makes the application reactive.
* When a packet not matching any existing rule is encountered, it is forwarded to the controller so that the controller can react to it via some appropriate action.
* A packet may also be forwarded to the controller in the event that it matches a flow entry and the associated action stipulates that the packet be passed to the controller.
* In the reactive model, the flow tables tend to continually evolve based on the packets being processed by the switch and by flows aging out.

**Proactive SDN Applications**

Less communication emanating from the switch to the controller.

The proactive SDN application sets up the switches in the network with the flow entries that are appropriate to deal with incoming traffic before it arrives at the switch.

Events that trigger changes to the flow table configurations of switches may come from mechanisms that are outside the scope of the switch-to-controller secure channel.

For example, some external traffic-monitoring module will generate an event which is received by the SDN application, which will then adjust the flows on the switches appropriately.

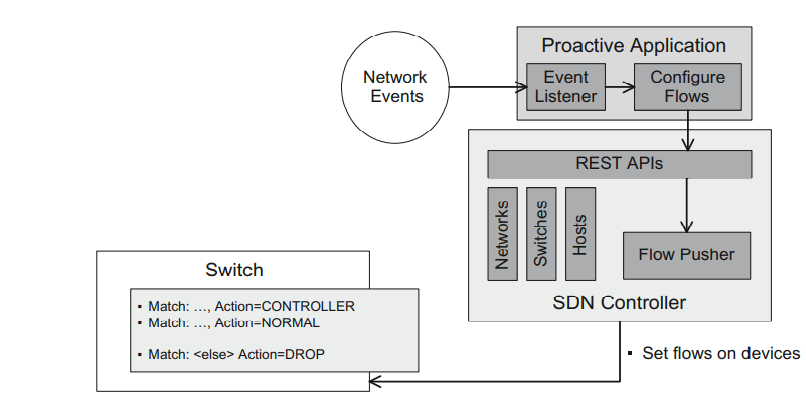
Applications that naturally lend themselves to the proactive model usually need to manage or control the topology of the network.

Examples of such applications include new alternatives to spanning tree and multipath forwarding.

Loss of the controller will have less impact in the proactive model if the failure mode specifies that operation should continue.

With the proactive model, there is no additional latency for flow setup, because they are prepopulated.

A drawback of the proactive model is that most flows will be wildcard-style, implying aggregated flows and thus less fine granularity of control



* In such an application no listeners are receiving from the controller events about switches, devices, and messages.
* Such listeners are not a natural fit with a one-way RESTful API, wherein the application makes periodic calls to the API and the API has no means of initiating communication back to the application.
* As shown in Figure, proactive applications rely on stimulus from external network events.
* Such stimuli may originate from a number of sources, such as traffic or switch monitors using SNMP or SNORT, or external applications like a server virtualization service, which notifies the application of the movement of a virtual machine from one physical server to another.
* The RESTful APIs are still able to retrieve data about the network, such as domains, subnets, switches, and hosts.
* RESTful APIs generally also provide an interface, often referred to as a **flow pusher**, which allows the application to set flows on switches.
* Thus the proactive application has the same ability to program flows as the reactive application but will tend to use wildcard matches more frequently.
* The last flow entry will typically be to DROP unmatched packets.
* This is because proactive applications attempt to anticipate all traffic and program flow entries accordingly.
* As a consequence, packets that do not match the configured set of rules are discarded.
* The match criteria for flow entries can be programmed such that most arriving packet types are expected and match some entry before this final DROP entry.
* If this were not the case, the purely proactive model could become an expensive exercise in dropping packets!