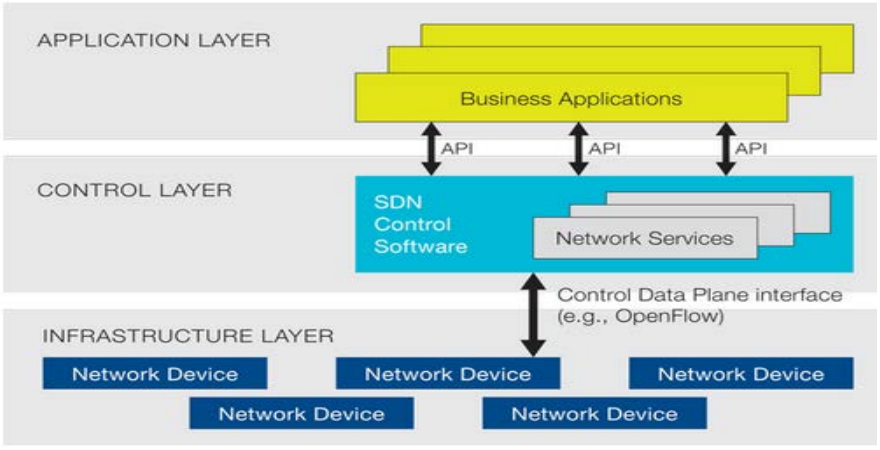
An Overview of OpenFlow

By Jim Metzler, Ashton Metzler & Associates
Distinguished Research Fellow and Co-Founder,
Webtorials Editorial/Analyst Division

The OpenFlow Protocol

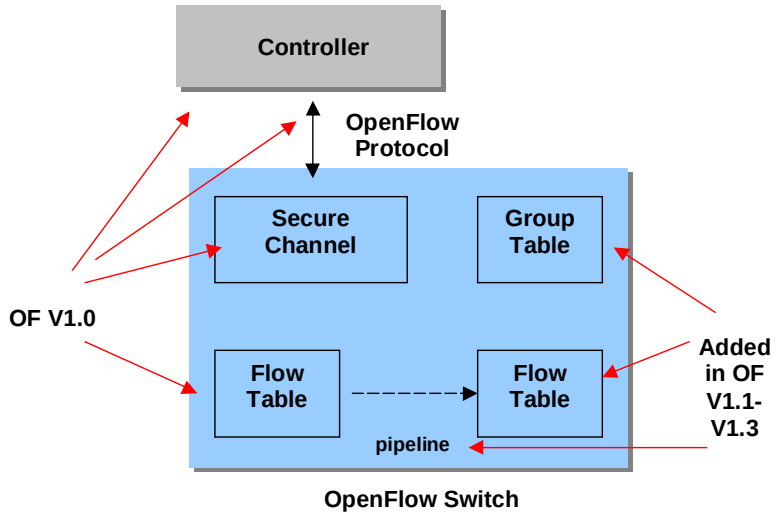
Figure 1 depicts the Open Networking Foundation's (ONF's) SDN architecture. Within that architecture OpenFlow is a standards-based protocol that enables a SDN controller to program the behavior of an OpenFlow-enabled switch. OpenFlow V1.0 was developed by Stanford University and was published in December 2009. The basic elements of an OpenFlow V1.0 network are shown on the left hand side of **Figure 2**. The central controller communicates with the switch's OpenFlow agent over a secure TLS (Transport Layer Security) channel. This channel could be either in-band or out-of-band. The OpenFlow agent on the switch populates the flow table as directed by the controller. Note that within **Figure 2**, OpenFlow is referred to as OF.

Figure 1: ONF's SDN Architecture



Subsequent to the publication of OpenFlow V1.0, the development of OpenFlow became the responsibility of the ONF. This OpenFlow specification has been enhanced three times. Version 1.1 was published in February 2011; V1.2 was published in December of 2011 and V1.3 was published in June of 2012. While few vendors adopted v1.1 or v1.2 of OpenFlow, many vendors have either already adopted v1.3 or have indicated that they will.

Figure 2: The Major Components of an OpenFlow Switch V1.0-V1.3



The data path of an OpenFlow V1.0 switch is comprised of a single Flow Table that includes the rules for matching flows to table entries, an action associated with each flow entry, and counters recording the number of packets and bytes received per flow and other port and table statistics, as shown in **Figure 3**.

Figure 3: The OpenFlow V1.0 Flow Table Fields

Header Fields	Counters	Actions

Figure 4 shows the 12-tuple of header fields that are used to match flows in the flow table,

Figure 4: The OpenFlow V1.0 Header Fields

Ingress Port	Ether Source	Ether Dest	Ether Type	VLAN ID	VLAN Prior	IP Source	IP Dest	IP Proto	IP TOS	Source Port	Dest Port

OpenFlow V1.0 switches are required to support two basic types of actions: Forward and Drop.

Forwarding is either directed to a physical port or to one of the following virtual ports:

- ALL: Send the packet out all interfaces, not including the incoming interface.
- CONTROLLER: Encapsulate and send the packet to the controller.
- LOCAL: Send the packet to the switch's local networking stack.
- TABLE: Perform actions in the flow table. Applies for only packet-out messages.
- IN PORT: Send the packet out the input port.

For OpenFlow V1.0 there are also a number of optional/recommended actions:

- NORMAL: Process the packet using the traditional forwarding path supported by the switch (for OpenFlow-hybrid switches)
- FLOOD: Flood the packet along the spanning tree
- ENQUEUE: Forward a packet through a specific port queue to provide QoS
- MODIFY FIELD: Change the content of header fields, including set VLAN ID and priority, strip VLAN, modify Ethernet or IPV4 source and destination addresses, modify IPV4 TOS, modify transport source and destination ports

When a packet arrives at the OpenFlow V1.0 switch, the header fields are compared to flow table entries. If a match is found, the packet is either forwarded to specified port(s) or dropped depending on the action stored in the flow table. When an OpenFlow Switch receives a packet that does not match the flow table entries, it encapsulates the packet and sends it to the controller. The controller then decides how the packet should be handled and notifies the switch to either drop the packet or make a new entry in the flow table to support the new flow.

Over the last few years extensive enhancements have been made to the OpenFlow specification under of the auspices of the ONF. A complete listing of the enhancements included in OpenFlow V1.1-V1.3 is well beyond the scope of this document. However, some of the major changes include:

- Additional components of a flow entry in the flow table as shown below. In addition to the match and counter fields, the following fields are included in the entry:

    - Instructions to execute actions or to modify the action set or pipeline processing
    - Priority: matching precedence of the flow entry
    - Timeouts: maximum amount of time or idle time before flow expiration
    - Cookie: opaque data value chosen and used by the controller to process flows

Match Fields	Counters	Instructions/Actions	Priority

- Flexible pipeline processing through multiple flow tables. As a packet is processed through the pipeline, it is associated with a set of accumulating actions and metadata. The action set is resolved and applied at the end of the pipeline. The metadata allows a limited amount of state to be passed down the pipeline.

- The new group table abstraction and group action enable OpenFlow to represent a set of ports as a single entity for forwarding packets. Different types of groups are provided, to represent different forwarding abstractions, such as multicasting or multi-pathing.

- Support for virtual ports, which can represent complex forwarding abstractions such as Link Aggregation Groups (LAGs) or tunnels. Encapsulation/Decapsulation of packets supports Network Virtualization tunnels, including PBB, QinQ VLAN stacking, and Push/Pop/Rewrite of MPLS headers.

- OpenFlow Extensible Match (OXM) uses a TLV structure to give a unique type to each header field increasing the flexibility of the match process.

- Basic support for IPv6 match and header rewrite has been added via OXM.

- Routing emulation; i.e., TTL decrement.

- Per flow meters measure and control the rate of packet forwarding—including rate limiting packets sent to controller.

- Support for multiple controllers to improve reliability.

With V1.4, OpenFlow will provide enhanced extensibility of the OpenFlow wire protocol and a new set of port properties to provide support for optical ports. This will allow Ethernet optical ports or optical ports on circuit switches to be configured and monitored.

Potential Use Cases and Benefits of OpenFlow

There are a number of possible ways for the control centralization, programmability, and flow forwarding characteristics of OpenFlow to be exploited by innovative users and vendors of network devices and software. This includes the following examples.

- **Centralized FIB/Traffic Engineering**
  One of the primary benefits of OpenFlow is the centralized nature of the Forwarding Information Base (FIB). Centralization allows optimum routes to be calculated deterministically for each flow by leveraging a complete model of the end-to-end topology of the network. This model can be build using a discovery protocol, such as the Link Layer Discovery Protocol (LLDP). Based on an understanding of the service levels required for each type of flow, the centralized OpenFlow controller can apply traffic engineering principles to ensure each flow is properly serviced. Bandwidth allocations can be controlled dynamically to provide bandwidth on demand with changing traffic patterns. The result can be much better utilization of the network without sacrificing service quality. Centralized route processing also allows the pre-computation of a set of fail-over routes for each possible link or node failure. Centralized processing also can take advantage of virtually unlimited processing power of multi-core processors and cluster computing for calculating routes and processing new flows.

  The Google G-Scale WAN backbone links Google's various global data centers. G-Scale is a prime example of a production OpenFlow Layer 3 network that is realizing the benefits of FIB centralization. The G-Scale control plane is based on BGP and IS-to-IS and the OpenFlow-only switches are very simple 128 port 10 GbE switches built by Google using merchant silicon (when Google built these switches, 128 port 10 GbE switches had not yet been introduced in the commercial market). Google has identified a number of benefits that are associated with its G-Scale WAN backbone including that Google can run the network at utilization levels up to 95%.

- **Other WAN Optimizations**
  WAN traffic can be dynamically rerouted to reduce/control latency for VoIP and other latency sensitive applications. Traffic can also be load balanced over parallel paths of differing costs.

- **QoS Optimization**
  With OpenFlow V1.3 per flow meters can be used for rate limiting or to provide real time visibility of application performance allowing the controller to modify forwarding behavior to maximize application performance. For example, the controller can configure an OpenFlow switch to modify the QoS markings to change the priority received over the remainder of the end-to-end path.

- **OpenFlow-Based Virtual Networking**
  With OpenFlow V1.3 virtual ports, an OpenFlow switch can be programmed to perform tunnel encapsulation and de-capsulation. Therefore, an OpenFlow switch can be programmed to be a network virtualization edge point that is part of an overlay-based network virtualization solution. OpenFlow can also provide another type of network virtualization for isolating network traffic based on flows segregation or segmentation. It is possible to leverage OpenFlow in such a way that flows are separated based on a subset of the match fields listed earlier in this document.

- **OpenFlow-Based Multi-Pathing**
  Most networking vendors offer data center fabric solutions featuring some form of Layer 2 multi-pathing to improve the networks capacity to handle “east-west” traffic flow characteristic of server virtualization, converged storage networking, and cluster computing. OpenFlow offers another approach to multi-pathing that does not rely on standards such as TRILL or SPB. As noted earlier, the OpenFlow Controller (OFC) can use LLDP to discover the entire network topology via discovering switches and switch adjacencies. Using this topological model, OFC can compute all the parallel physical paths, including paths that share some network nodes and other paths that are entirely disjoint (and therefore offer higher reliability). OFC can then assign each flow across the network fabric to a specific path and configure the OpenFlow switches’ flow tables accordingly. The OFC can then offer shared and disjoint multi-pathing as network services that can be delivered to applications. With appropriate processing power, the OFC can support very large scale networks and high availability via path redundancy and fast convergence following link or node failures.

- **OpenFlow Security Services and Load Balancer Services**
  By virtue of Layer 2-4 flow matching capability, OpenFlow access switches can perform filtering of packets as they enter the network, acting as simple firewalls at the edge. With OpenFlow switches that support modification of packet headers, the OpenFlow Controller will also be able to have the switch redirect certain suspicious traffic flows to higher-layer security controls, such as IDS/IPS systems, application firewalls, and Data Loss Prevention (DLP) devices. Other security applications built on OpenFlow controller can match suspicious flows to databases of malware signatures or divert DDoS attacks. Another possible security application of OpenFlow would be in Network Access Control (NAC).

  OpenFlow with packet header modification will also allow the switch to function as a simple, cost-effective load-balancing device. With modification functionality, a new flow can result in a new flow table entry that includes an action to modify the destination MAC and IP addresses. The modified address can be used to direct traffic to the server selected by the controller load balancing application.

  Indiana University (IU) has developed an OpenFlow-based, load-balancing application called FlowScale. According to the University, “FlowScale provides complex, distributed load balancing of network traffic using an OpenFlow-capable Top of Rack (ToR) switch. IU deployed the application into its Intrusion Detection System (IDS) to distribute traffic evenly to sensors. When fully deployed, the system will span the IU Bloomington and IUPUI networks and have the capability to distribute traffic at rates exceeding 500Gb/s.”

- **Network Taps**
  With OpenFlow virtual ports, the functionality of a network tap can be programmed into the OpenFlow switch, allowing selected traffic to be monitored without deploying physical taps. Traffic can also be replicated and redirected and directed to any monitoring device in the network.

- **Service Insertion/Chaining**
  OpenFlow’s ability to dynamically reroute flows allows network services provided by physical or virtual appliances (e.g., firewalls, NATs, load balancers, and WOCs) to be inserted in the path of the flow. Redirecting the flow to the next service can be based on encapsulation or rewrite of the destination MAC address.

- **Circuit Provisioning**
  With extensions in V1.3 and V1.4 OpenFlow can support circuit-switched paradigms, including CWDM, DWDM, and MPLS with specific path selection and requested levels of CBR and priority. Circuits can be provisioned on a dynamic, scheduled, or permanent basis. Recovery from failed circuits can be via predetermined backup paths or by dynamic path selection. Circuit provisioning can take into account performance metrics, port states, and endpoint utilization.

About the Webtorials® Editorial/Analyst Division

The Webtorials® Editorial/Analyst Division, a joint venture of industry veterans Steven Taylor and Jim Metzler, is devoted to performing in-depth analysis and research in focused areas such as Metro Ethernet and MPLS, as well as in areas that cross the traditional functional boundaries of IT, such as Unified Communications and Application Delivery. The Editorial/Analyst Division's focus is on providing actionable insight through custom research with a forward looking viewpoint. Through reports that examine industry dynamics from both a demand and a supply perspective, the firm educates the marketplace both on emerging trends and the role that IT products, services and processes play in responding to those trends.

For more information and for additional Webtorials® Editorial/Analyst Division products, please contact Jim Metzler or Steven Taylor .

Published by Webtorials Editorial/Analyst Division
www.Webtorials.com

Division Cofounders:
Jim Metzler
jim@webtorials.com
Steven Taylor
taylor@webtorials.com

Professional Opinions Disclaimer

All information presented and opinions expressed in this publication represent the current opinions of the author(s) based on professional judgment and best available information at the time of the presentation. Consequently, the information is subject to change, and no liability for advice presented is assumed. Ultimate responsibility for choice of appropriate solutions remains with the reader.