Software-Defined Networking Changes the Paradigm for Mission-Critical Operational Technology Networks

Paul Robertson

Introduction

While software-defined networking (SDN) was developed primarily to solve an information technology (IT) problem by providing a more efficient approach to managing dynamic high-traffic networks, SDN also offers significant advantages for operational technology (OT) networks.

While both OT and IT networks manage the transfer of data, OT networks support process-oriented applications. They manage the process and automation systems that control power systems, industrial plants, transportation infrastructure, and other critical systems. Unlike IT networks that deal primarily with the movement of data, OT networks are focused on the real-time decision making necessary for the operation of mission-critical systems. They carry information from sensors, transducers, intelligent electronic devices, and automation controllers.

Ethernet has become an established technology for data networking, based largely on its success in creating a common interface and protocols to enable interoperable communications between devices and applications from multiple manufacturers. Ethernet's convenience has led to its widespread use in OT networks. However, for managing real-time processes, traditional Ethernet has several limitations: slow healing times, forced behavior, no inherent security, difficult testing, and no network visibility.

This white paper originally appeared as an article in [1].

SDN Explained

SDN is a new approach for managing the forwarding of Ethernet packets through a network. It is revolutionizing the management and operation of large-scale IT enterprise networks, cloud infrastructures, and data center systems. SDN provides better support for the highly dynamic characteristics of these networks. SDN is ideally suited for OT networking as well because it resolves the limitations of applying Ethernet to real-time OT systems.

Before explaining how SDN addresses the unique challenges of OT systems and how OT-focused SDN offers a different value proposition than IT-focused SDN solutions, it is helpful to review the technology. A basic SDN architecture is shown in Figure 1.

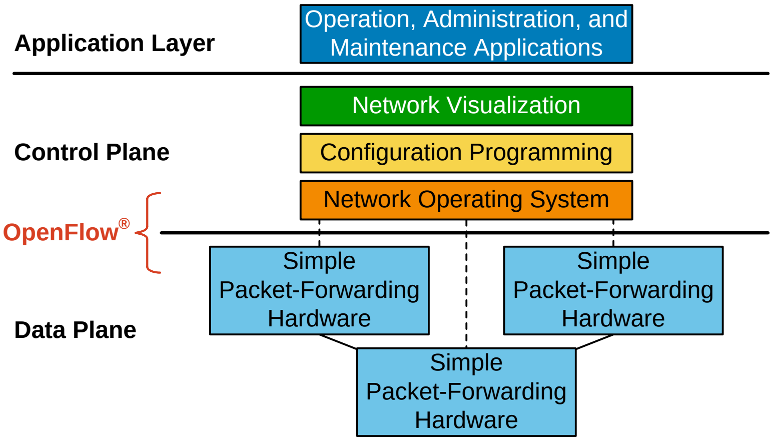
Application Layer
Operation, Administration, and Maintenance Applications
Network Visualization
Control Plane
Configuration Programming
Network Operating System
OpenFlow®
Simple Packet-Forwarding Hardware
Simple Packet-Forwarding Hardware
Data Plane
Simple Packet-Forwarding Hardware

Figure 1 SDN Architecture

In SDN, the control plane—which determines what to do with each Ethernet packet by making forwarding decisions—is separated from the data plane, which physically forwards the Ethernet data packets.

In traditional Ethernet, the control plane is located in network appliance hardware (i.e., switches). In SDN, however, the control plane is located in software called a flow controller. This allows the switching hardware to be simpler and optimized for packet forwarding. At the same time, the flow controller is able to take advantage of a software-oriented architecture to program and configure all of the switches in the network. The combination provides a programmatic approach to implementing networking functions that results in enormous flexibility, greater speed, and increased security.

Several protocols have been defined to provide an interface between the control plane and the data plane. OpenFlow, for example, is an interface specification developed by the Open Networking Foundation to promote the adoption of SDN and enable interoperability between different flow controller implementations and different manufacturer's switch hardware.

In addition to the control plane, SDN defines an application layer that provides a method for supporting operation, administration, and maintenance tasks. The application layer provides a mechanism to extract meaning from and interact with the complexity of the low-level programming rules and actions that control how each packet is forwarded. This simplifies the network configuration process and enables users to configure their networks using a higher level of abstraction.

How SDN Works

SDN uses the following terminology:

- Match criteria specify Ethernet frame header information attributes that each ingress packet is compared against.
- An instruction defines a specific action or set of actions that is applied to all packets that meet specified match criteria. Instructions define which port to forward a packet out of and what path the packet takes across the network.
- A flow is a group of attributes that defines a single communications session. Each flow is defined by a set of match criteria and corresponding instructions that are applied to each packet that ingresses a switch.
- A flow table contains all of the flows defined within a switch.
- The flow controller is the centralized controller that programs the rules and flow tables in each switch.

The flow controller gives each switch a set of rules containing match criteria and instructions to determine the operations that should be performed on each packet the switch receives. These rules are stored in a flow table. The switch matches the packet header information against each rule contained in the flow table. If a matching flow table entry is found for a packet, the actions associated with that flow table entry are executed, such as sending the packet out a specific port number, modifying a field, or dropping the packet.

Instructions can be predefined for a situation in which an SDN switch receives a packet it has never seen before (i.e., for which it has no matching flow table entries). One very powerful attribute of SDN is that it allows a deny-by-default security profile that can instruct each switch to drop every packet that does not meet a predefined rule. Counters are maintained for each flow table entry and used to maintain statistics on traffic by flow, port, queue, and so on.

Each switch has the ability to look deep into an Ethernet frame and use that information to determine a rule and an associated action. This makes SDN switches true multilayer devices that can operate across Layer 1 to Layer 4 of the Ethernet frame (see Figure 2).

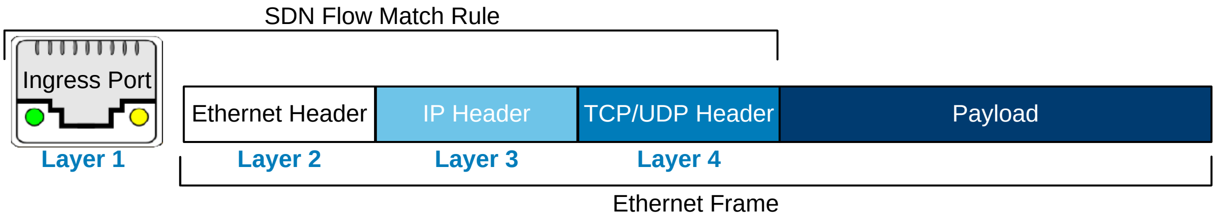

Figure 2 Multilayer Match Rules Forward Approved Packets

Layer 1 is the physical layer that defines the physical port and media that data are received and transmitted over.

Layer 2 is the data link layer that establishes and terminates connections between physically connected devices. It uses media access control (MAC) addresses to uniquely identify each device.

Layer 3 is the network layer. It provides the mechanism to transfer variable-length data from one node to another. It is responsible for packet forwarding between routers using Internet Protocol (IP) address information.

Layer 4 is the transport layer. It is responsible for establishing connectivity between applications on host devices. It uses sockets to define the end points in the network. A socket address is a combination of the IP address and the port number. The port number defines the communications protocol and associated application. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the most commonly used protocols. In the following example, TCP Port 20000 is used, which defines Distributed Network Protocol (DNP3) for supervisory control and data acquisition (SCADA) applications.

Because each SDN switch can operate as a multilayer device, it is possible to create complex match and action rules that are application-aware. In the example shown in Figure 3, the switch has been configured to only forward SCADA traffic between two specific devices. A rule has been created to only forward DNP3 packets on TCP Port 20000 that are received through Port 1 and meet the following criteria:

- Source MAC address: **00:30:A7:06:29:01**.
- Destination MAC address: **00:30:A7:06:13:29**.
- Source IP address: **1.1.1.2**.
- Destination IP address: **2.2.2.2**.
- Destination TCP port: **20000**.

If the packet matches these criteria, the switch forwards the packet out Port 3.

Physical Port ID	Source MAC Address	Destination MAC Address	EtherType	VLAN ID	IPv4 Source	IPv4 Destination	TCP/UDP Source	TCP/UDP Destination
1	00:30:A7:06:29:01	00:30:A7:06:13:29	*	*	1.1.1.2	2.2.2.2	*	20000

Action
Output Forward Port 3

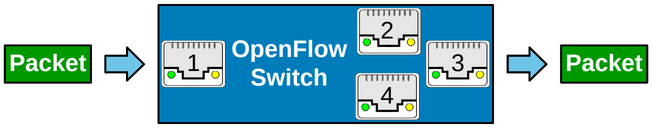

Figure 3 OpenFlow Match and Action Example

This multilayer rule example demonstrates several powerful concepts. SDN allows application-specific network paths to be engineered; it enables each network device to operate as a Layer 2, 3, and 4 device; and it supports the ability of each switch to operate as a deny-by-default gateway.

Applying SDN to OT

To date, SDN's success has been attributed to its ability to solve challenges associated with managing the large, dynamic IT networks with high and variable traffic loads primarily found in data centers. These IT networks have changing usage patterns, unpredictable application demand, and varying traffic volumes.

In contrast, OT networks are static. In an OT network, all applications and services are known at the design stage, and a rule set can be predefined for each switch to support the application and connectivity requirements of the network. Typically, very few changes are required to the network over time, and traffic volume remains predictable.

Microsecond Failover Times

By optimizing switching hardware to efficiently execute the match rules within the flow tables, it is possible to achieve microsecond-speed packet inspection, decision, and forwarding times. In addition, by predetermining each primary and failover path, it is possible to perform path healing in tens of microseconds, which is two orders of magnitude faster than the spanning-tree algorithms (STAs) used by traditional Ethernet. Networks can heal quickly enough that the only data loss is that of the packet being transferred during the link failure. In many cases, zero packets are lost during failover events.

Predictable Behavior

With SDN, it is possible to define different forwarding paths for different applications. In the example shown in Figure 4, engineering access, SCADA, and Generic Object-Oriented Substation Event (GOOSE) traffic have each been given different forwarding paths across the network. There are two paths defined for GOOSE, a primary path (**GOOSE 1**) and a failover path (**GOOSE 2**). The GOOSE 2 path is on a dedicated port and cable that are separate from the engineering access path.

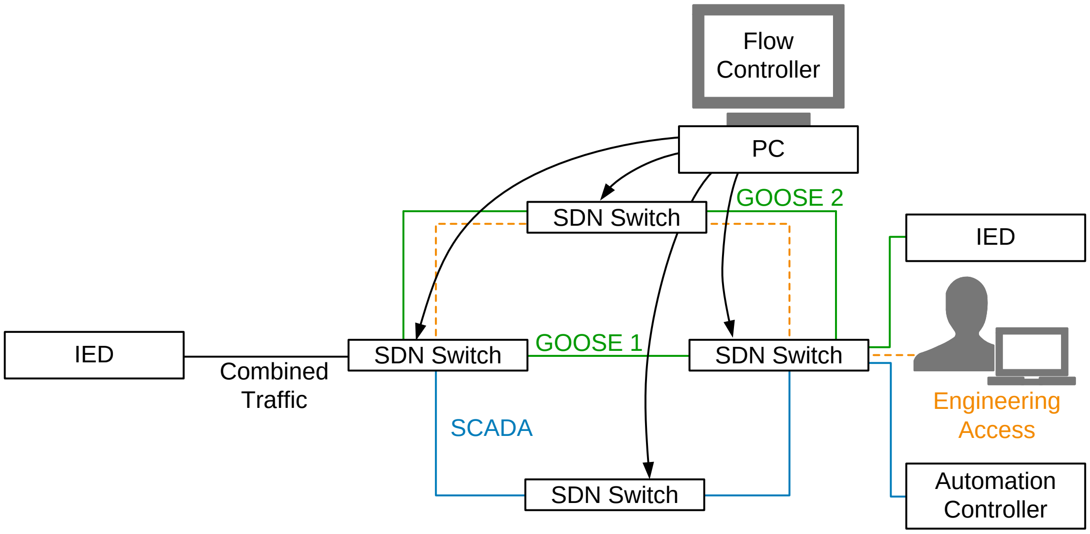

Figure 4 Control Packet Forwarding by Application

With SDN, it is possible to have multiple physical connections between the same switches to provide physical redundancy and traffic segregation. This is not possible with traditional Ethernet because STAs prevent multiple physical connections by blocking ports. By taking this traffic engineering approach, predictable delivery and failover performance can be achieved, and network performance can be validated under all contingencies prior to commissioning.

Stronger Network Security

Within an OT environment, new applications should not start communicating across the network to meet changing end user needs. As a consequence, the network does not rely on constant participation with the flow controller to manage updates to flow tables. After the initial configuration of each switch by the flow controller, the network can be locked down with a rule set that only allows predefined application traffic to be forwarded. This provides a significant security advantage over traditional Ethernet. Each switch provides a deny-by-default function on every incoming packet. If the packet does not match a predefined rule, it is dropped by default. Alternatively, it can be forwarded to a dedicated Intrusion Detection System (IDS) for further analysis. SDN offers significant network security advantages to help address concerns about the vulnerability of critical infrastructure systems, such as the power grid, to cyberattacks.

Simplified Design and Test

One of the biggest challenges with designing and implementing Ethernet networks is verifying that the network will behave as designed, particularly during failover scenarios. There is no easy way to perform contingency testing other than pulling cables and using a protocol analyzer to verify that the STA being used has converged as expected based on the root switch and network topology.

SDN changes the paradigm for network design and testing. Using software to abstract the implementation of each rule, the network designer can specify the path each application should take through a network and predefine each failover path. The network can then be validated for each contingency using a programmatic approach that ensures all failure scenarios are tested and validated.

Interoperability

SDN is based on Ethernet and does not require the modification of the Ethernet packet in order to be implemented, so it is interoperable with standard Ethernet. Care should be taken when establishing the boundaries between SDN and traditional managed Ethernet networks that rely on protocols to operate. Protocols, such as Border Gateway Protocol (BGP), Rapid Spanning Tree Protocol (RSTP), and Open Shortest Path First (OSPF), are in constant conversation in a traditional managed Ethernet network. SDN does not use these protocols to manage network operation, and therefore, SDN cannot be mixed with managed Ethernet switches across a single topology. Distinct boundaries between an SDN and a managed Ethernet network should be created, and rules should be implemented to instruct an SDN switch on how to appropriately deal with managed Ethernet protocol packets.

Network Visibility

Lack of network management capability has always been a limitation of traditional Ethernet. By putting the control plane in a centralized entity, the flow controller can monitor the configuration, health, and status of the complete network.

Conclusion

SDN offers a fundamental change to the way Ethernet networks determine how to forward packets and to how they are managed by the network administrators. SDN removes the control plane from the physical switching hardware and locates it in a centralized flow controller. By doing so, switches change from being distributed, intelligent, autonomous devices to being simpler entities that perform a set of instructions provided by the flow controller. A suite of protocols communicating in the background is no longer needed to manage how each switch discovers neighbors, prevents loops, and recovers from link failures.

The primary benefit of SDN is its ability to support rapid changes to network operation and topology to address fast-changing user needs. Many view SDN primarily as a solution for dynamic IT networks. However, the principles of SDN offer significant benefits to time-critical OT networking by addressing the limitations that traditional managed Ethernet has in terms of slow healing, forced behavior, lack of security, difficult testing, and lack of network visibility. Table 1 provides a summary of the key advantages that SDN provides over traditional Ethernet. SDN offers a performance breakthrough in all of these areas and promises to redefine Ethernet for mission-critical applications.

Table 1 Comparison of SDN and Ethernet Performance

Attribute	SDN	Traditional Ethernet
Healing	<0.1 ms	>10 ms
Predictability	Yes	No
Security	Deny-by-default	Based on trust
Design and testing	Simple	Difficult
Interoperability	Yes	Yes
Network visibility	Yes	No

Reference

[1] P. Robertson, “Software-Defined Networking for Mission-Critical Operations,” *Industrial Ethernet Book*, Issue 98, February 2017. Available: http://www.iebmedia.com.

Biography

Paul Robertson is a senior marketing program manager for the communications product lines at Schweitzer Engineering Laboratories, Inc. (SEL). He has over 25 years of experience developing and marketing products for the telecommunications industry, spanning cellular wireless and wire line communications systems. Paul worked in various technical and marketing roles for Motorola, Hewlett-Packard, and Agilent Technologies before joining SEL. He has a B.Eng. in electrical and electronic engineering from the University of Strathclyde and an M.B.A. from Edinburgh Business School.

SEL Making Electric Power Safer,
More Reliable, and More Economical

Schweitzer Engineering Laboratories, Inc.
Tel: +1.509.332.1890 | Email: info@selinc.com | Web: www.selinc.com