Solving Performance and Cybersecurity Challenges in Substation and Industrial Networks With Software-Defined Networking

Dalton Shaffer and Devin Thewlis
City Light & Power, Inc.

Tom Bartman and Tristan Atiyeh
Schweitzer Engineering Laboratories, Inc.

Presented at the
7th Annual PAC World Americas Conference
Virtual Format
September 1–2, 2020

Originally presented at the
Power and Energy Automation Conference, March 2020

Solving Performance and Cybersecurity Challenges in Substation and Industrial Networks With Software-Defined Networking

Dalton Shaffer and Devin Thewlis, *City Light & Power, Inc.*
Tom Bartman and Tristan Atiyeh, *Schweitzer Engineering Laboratories, Inc.*

Abstract—**Software-defined networking (SDN) is a method for moving packets between an origin and destination that relies on strict forwarding rules for switches. With SDN, you define which devices on your network can communicate and how they communicate.**

This paper discusses performance and cybersecurity challenges and how SDN addresses them. It reviews the application, performance, and failover of IEC 61850 Generic Object-Oriented Substation Event (GOOSE) protocol while using SDN in a substation environment. The paper also discusses SDN in securing engineering access and supervisory control and data acquisition (SCADA) communications. Finally, the paper highlights the major cybersecurity benefits of SDN employed in a substation or industrial control system network and describes what happens when an unapproved device attempts to connect.

I. Introduction

Software-defined networking (SDN) is a type of network architecture that enables central control and programming for improved performance, strict cybersecurity, and network visualization. When compared to traditional Ethernet networks, SDN redefines the performance and security standards for operational technology (OT) in critical industrial networks, such as electric power substations. SDN not only defines which devices can communicate across the network, but it also defines the protocols in which they are allowed to communicate. This higher level of network security addresses the latest cybersecurity concerns regarding mission-critical networks. In addition, SDN maximizes network performance by using pre-engineered failover paths. SDN also gives a near real-time view of what is happening on a network and what applications and devices are present on it.

The increase in performance, tight cybersecurity, and network awareness make SDN an excellent solution for substations. This paper examines the application of SDN in an electric power substation where security and Generic Object-Oriented Substation Event (GOOSE) protocol performance are critical to successful facility operations.

II. Traditional Ethernet vs. SDN

SDN has become an attractive solution for addressing performance and cybersecurity challenges. For example, large organizations such as Google rely heavily on SDN [1]. However, while Google uses SDN mainly for dynamic bandwidth control, substation operators rely on SDN to provide fast failover, cybersecurity, and network visibility.

A. Challenges With Traditional Ethernet

Some of the challenges with using traditional Ethernet in substations and critical industrial networks are vulnerable plug-and-play connections, long network failover and healing times, cybersecurity risks, and problems with network visibility.

1) Plug-and-Play Communications

With traditional Ethernet communications, data packets move between Ethernet switches based on the media access control (MAC) address of the connected devices. A MAC address is unique to a device. When a device is connected to an Ethernet switch, the switch learns the MAC address and stores it in a MAC table. Because the switch knows which physical port is associated with the MAC address of each connected device, switches are able to forward and receive packets to and from each device in the network.

MAC address learning was a key concept in the development of traditional Ethernet networking and the idea of plug-and-play communications. When devices are connected to a traditional Ethernet switch, they begin communicating without any human involvement. However, while plug-and-play technology is convenient, it lacks a cybersecurity profile and is based on a trusted-user model, which assumes that users are communicating with devices and other users in the intended manner.

2) Network Failover and Healing

In the event of an Ethernet switch failure or cable break, redundancy is required to reduce the impact on the network. In a traditional Ethernet network, switches are configured in a ring topology to achieve redundancy, as shown in Fig. 1.

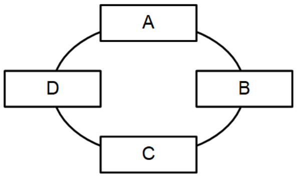


Fig. 1. Network Switch Ring Topology

A ring topology is a visually simple way of illustrating the concept of redundancy. In Fig. 1, if Ethernet packets from Switch A are intended for Switch B, but the cable between the switches is broken, traffic will take the backup path through Switches D and C to Switch B. The algorithm that directs traffic to take a backup path is Spanning Tree Protocol (STP).

The performance of STP is acceptable in an information technology (IT) application where latency and failover are typically not as important as in OT applications. IT networks usually include non-critical traffic such as email, print, and file servers.

The performance of STP in an OT network is unacceptable for some protocols such as IEC 61850-9-2 (Sampled Values process bus). An IEC 61850-9-2 merging unit publishes data at a fixed rate of 4,800 samples per 60 Hz power system cycle, resulting in data being transmitted every 208 µs [2]. If a relay misses three consecutive samples, it transitions to an offline status. This means that failover for this type of network must be faster than 624 µs. STP, therefore, is too slow to meet the requirements of IEC 61850-9-2. Furthermore, STP takes time to heal the network (i.e., convergence time) when failover becomes active, often as long as 10 to 50 ms.

3) *Cybersecurity*

Traditional OT Ethernet networks lack inherent cybersecurity without additional equipment such as an intrusion detection system (IDS), which requires updates, patches, and personnel to maintain. In addition, the plug-and-play architecture lacks a cybersecurity profile because it is a trusted model. Advanced persistent threats (APTs) remain a concern in today's networks. An APT remains undetected, moves slowly, and in some cases may shut down to avoid detection before later performing probes of the network. The application of SDN solves many of these cybersecurity challenges (discussed later in this paper).

4) *Network Visibility*

Traditional Ethernet networking shows devices and switches and how they are connected; however, it does not show the conversations occurring between devices. The reason for this is that legacy networking operates at the MAC layer, which does not provide information about conversations on Transmission Control Protocol/User Datagram Protocol (TCP/UDP) ports. In addition, testing all possible fault conditions can be extremely difficult, time-consuming, and impractical.

B. *SDN Principles*

Before discussing how SDN solves traditional Ethernet challenges, it is important to understand the principles of SDN.

As discussed previously, a standard Layer 2 Ethernet switch learns and stores the MAC address of a connected device. Packets are forwarded to specific ports based on these MAC addresses. This operation consists of two planes: the control plane and the data plane. The data plane consists of the Ethernet ports of the switches where packets ingress and egress. The control plane is where the decisions are made on forwarding packets (i.e., the MAC address tables).

The fundamental concept of SDN is that the control plane is removed from the switch. The decision-making process, therefore, is no longer performed by the switch. Fig. 2 illustrates the separation of the control plane and the data plane.

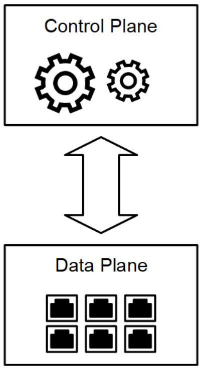


Fig. 2. Separating the Control Plane From the Data Plane

In an SDN environment, the control plane is a centrally based controller consisting of hardware and software. A typical SDN controller is a computer with SDN configuration software. Fig. 3 represents this architecture, where the controller is the brain that directs how the switches forward traffic across the network.

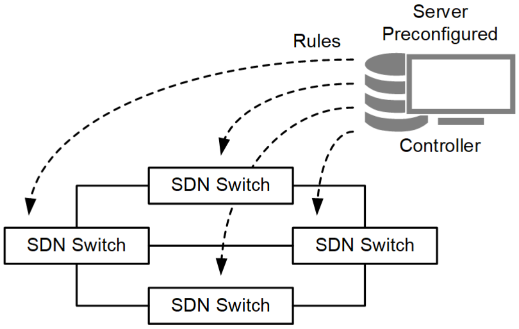


Fig. 3. Central Controller in an SDN Environment

The central controller in an SDN architecture determines how traffic will flow between devices, the protocols through which they can communicate, the paths the traffic can take, and how to perform failover in the event of a broken link or switch failure. The SDN solution used in the application discussed later in this paper is an OT SDN solution in which the controller is not required for network operation; i.e., after the flows are pushed to the switches, the controller is not required to direct traffic flow.

C. *SDN Solves Traditional Ethernet Challenges*

Section II discusses how traditional Ethernet, by its nature, lacks a cybersecurity profile in its plug-and-play model, does not meet the network failover speeds IEC 61850-9-2 requires, and lacks network visibility. This section discusses how SDN solves these challenges for use in substation and industrial networks.

1) *Strict Control of Device Communication*

SDN in critical applications is not based on a plug-and-play model. Instead, it is a deny-by-default model that gives the engineer control over which devices on the network are allowed to communicate and with which other devices. This is discussed in greater detail in Subsection 3.

2) *Fast Failover*

In SDN, fast failover is achieved through predefined paths. Rather than waiting for convergence to occur (as in a traditional STP Ethernet switch), an SDN switch uses predefined failover paths in response to a broken link or failed switch. In SDN, the health of the primary and backup links is always being monitored, and contrary to STP convergence, fast failover is predefined and automatically conducted through a simple if-then process (as represented in the following statement).

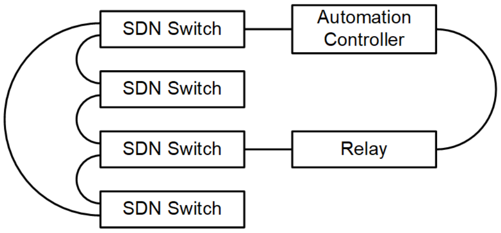
```
if primary_link == UP
    use(primary_link);
else
    use(backup_link);
```

Fast failover is a requirement in the application of GOOSE messages. SDN constantly monitors the health of primary, backup, and secondary paths on the network. By using predefined, redundant paths, SDN can perform failover in under 100 µs. This speed is sufficient for applications that require faster failover than STP.

3) *Cybersecurity*

SDN achieves cybersecurity through its deny-by-default policy, its use of device approval, and its use of flows (or permissions). When a device is connected to the port of an SDN switch, the controller recognizes a new device not previously approved. A device must be approved for it to be allowed on the network. Even when a device is approved, however, it cannot communicate until a flow is created to define traffic between itself and the other devices on the network. As mentioned previously, SDN not only determines *which* devices communicate, but also *how* they communicate.

Fig. 4 shows a flow (i.e., a permission between an automation controller and a relay on an SDN network) and logical connections (i.e., how devices are allowed to communicate). In this example, the automation controller and relay can only communicate via GOOSE messages. Address Resolution Protocol (ARP) is also included because it is required for Ethernet devices to begin communicating.



Local Connections

From Automation Controller	From Relay
ARP UnicastBidirectional Success	ARP UnicastBidirectional Success
GOOSE MulticastBidirectional Success	GOOSE MulticastBidirectional Success

Fig. 4. SDN Logical Connections

For SDN to allow traffic, a set of rules must be set (i.e., a flow). A flow is a set of match rules that must all be met before an action is performed on a packet. Fig. 5 shows a typical match rule based on Layers 1 through 4. A match rule can contain one or more layers.

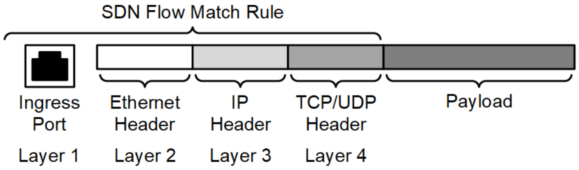


Fig. 5. SDN Match Rule Using Layers 1 Through 4 in an Ethernet Frame

As each packet ingresses into an SDN switch, the packet is inspected. If the packet matches the flow match rule, the predefined action for it is taken. Fig. 6 is an example of a GOOSE flow match rule for a relay with a virtual local-area network (VLAN) ID of 10. The rule inspects the Layer 2 Ethernet source and destination (EthSrc and EthDst), Ethertype (EthType), and VLAN ID (VlanVid).

Match Name: ChesterSub_Goose
Match Fields

Name	Value
EthDst	01:0c:cd:01:00:01
EthSrc	00:03:a7:09:3a:c0
EthType	GOOSE
VlanVid	10

Fig. 6. GOOSE Flow Match Rule With VlanVid Match Field

Another application of a flow match rule is when using a device that is configured through a web interface. For example, if a user needs to configure a device via a web browser, a flow match rule is applied, as shown in Fig. 7. In this case, the rule examines a packet ingressing into a specific port.

Match Name: ChesterSub_RTU_Web
Match Fields

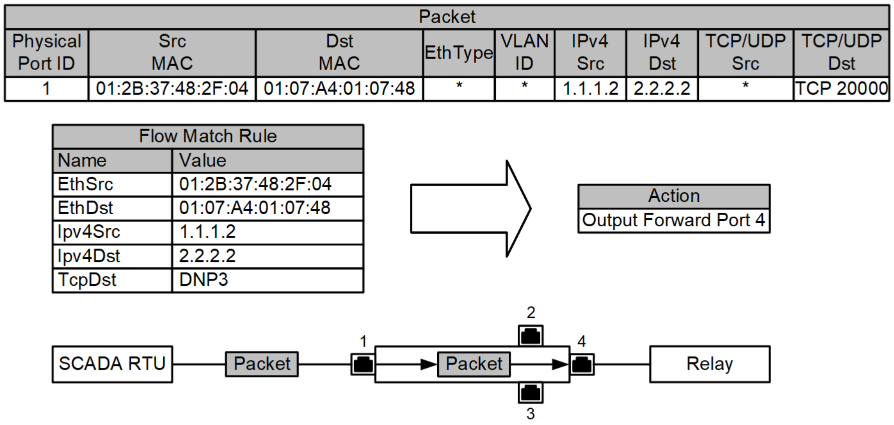
Name	Value
EthType	Ipv4
IpProto	TCP
TcpDst	HTTPS

Fig. 7. Secure Web Flow Match Rule With TcpDst Match Field

The flow rules in Fig. 7 match with a set of conditions, so SDN gives the user the flexibility to build the network around the devices.

If a packet ingresses into a switch and does not satisfy a matching rule, the packet is dropped. For applications designed to provide a deeper level of analysis, a dropped packet (e.g., a malicious packet) is given a specific priority. Packets that match this specified priority can be forwarded to a specific SDN switch port for further analysis.

For packets that satisfy a flow match rule, the next step is an action. An action is simply an instruction of where a switch forwards a packet.



Packet								
Physical Port ID	Src MAC	Dst MAC	EthType	VLAN ID	IPv4 Src	IPv4 Dst	TCP/UDP Src	TCP/UDP Dst
1	01:2B:37:48:2F:04	01:07:A4:01:07:48	*	*	1.1.1.2	2.2.2.2	*	TCP 20000

Flow Match Rule	
Name	Value
EthSrc	01:2B:37:48:2F:04
EthDst	01:07:A4:01:07:48
Ipv4Src	1.1.1.2
Ipv4Dst	2.2.2.2
TcpDst	DNP3

Action
Output Forward Port 4

Fig. 8. Packet Forwarding Via Flow Match Rule

In this example, an Ethernet packet from a supervisory control and data acquisition remote terminal unit (SCADA RTU) has arrived at Port 1 of an SDN switch. The packet is checked against the flow match rule for Port 1. In the example in Fig. 8, the flow match rule examines five portions of the Ethernet frame: the source and destination MAC addresses (Ethernet source and destination), the Internet Protocol (IP) address source and destination, and the Transmission Control Protocol (TCP) port destination (in this case Port 20000 is used for DNP3, which is a popular protocol in electrical substation applications). The packet matches the rule and then performs the designated action. In this example, the packet is forwarded to Port 4 of the connected substation relay.

This strict control over approved devices on the network, and the protocols for which they are approved to communicate, illustrates the benefit of SDN from a network engineering and cybersecurity perspective.

SDN will not allow a device to connect to the network without prior approval. This requires authentication that the device is known and trusted. Fig. 9 shows an unapproved device connected to an SDN network. In this example, a device with an IP address of 192.168.1.15 is connected to an SDN switch with a cable. The SDN switch recognizes the IP address of the connected device and displays it on the user interface. It remains in an unapproved state until approved or removed from the network, and a log entry is made recording that the device was connected.

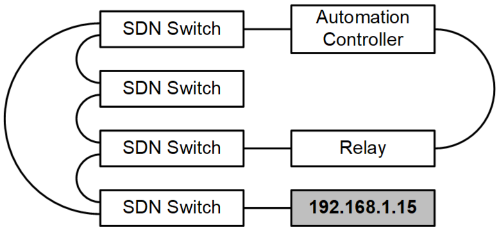


Fig. 9. Unapproved Connected Device

4) *Network Visibility*

SDN provides greater network visibility than traditional Ethernet. A great benefit of SDN is that it provides the user the ability to see which devices are on the network, how they are connected, and what conversations are occurring between devices. For example, Fig. 10 gives a visual representation of how devices are connected in an SDN network, both physically and logically.

In Fig. 10, physical connections are shown on the left side of the network, and flows, i.e., logical connections (permissions), are displayed on the right.

Table I gives a comparison between traditional Ethernet and SDN based on the main characteristics described in this paper.

TABLE I
COMPARISON BETWEEN TRADITIONAL ETHERNET AND SDN

	Traditional Ethernet	**SDN**
Failover Time	Slow; usually greater than 50 ms	Fast; less than 100 µs
Cybersecurity	Very limited; devices communicate with no security model	High; devices require permission to communicate, and how devices communicate can be controlled
Network Visibility	Low; information about conversations between devices *is not* available	High; information about conversations between devices *is* available

D. *Application of SDN and GOOSE in a Substation*

When applying SDN and GOOSE in a substation, SDN switches are configured to primarily allow DNP3 and GOOSE protocols to transfer information between devices in an electrical distribution network. DNP3 protocol is used for SCADA applications, and GOOSE protocol is used in high-speed protection blocking schemes.

The communications system is designed to place a group of intelligent electronic devices (IEDs) on a unique VLAN to prioritize and limit the amount of broadcast traffic on the network. The result is a high-speed, reliable communications network with a high degree of cybersecurity.

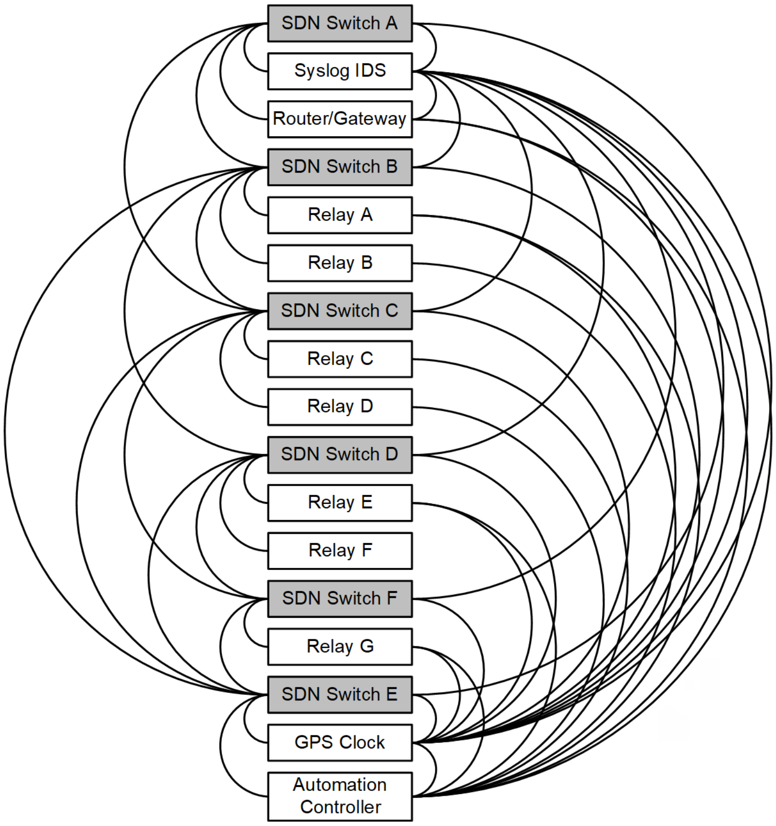


Fig. 10. SDN Network Connections: Physical (Left) and Logical (Right)

In SDN, packet forwarding is based on the applications using the network, so it is possible to define different forwarding paths for different applications. In the example in Fig. 11, engineering access, SCADA, and GOOSE traffic are each given different forwarding paths across the network. The application uses a different VLAN for the GOOSE messages of each relay.

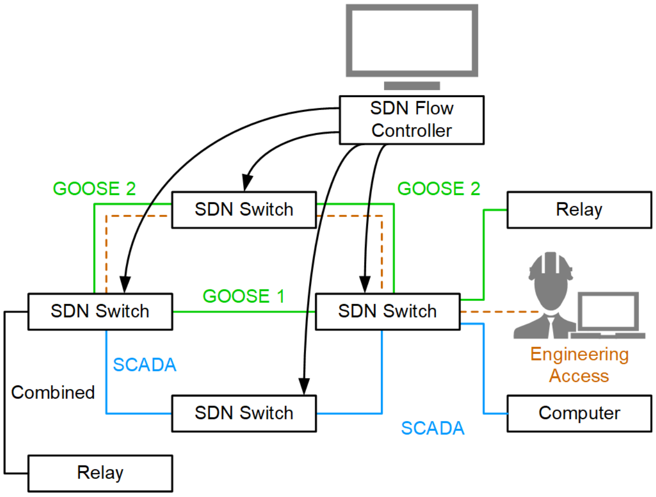


Fig. 11. Traffic Engineering With SDN

III. The SDN Controller and Physical Security

With SDN, the controller is the main threat vector. “The controller is the brains of the network” [3]. It allows the engineer access to the entire network; therefore, it must be protected from physical and cyber threats.

SDN southbound application program interfaces are used for communication between the SDN controller and the individual switches. An SDN northbound application programming interface on the controller enables applications to program the network and request services. An attack or compromise of these channels can lead to unauthorized changes to the network. Both of these interfaces must use encryption to provide a confidential and authenticated communications channel. In addition, changes to the configuration should be monitored and audited, and role-based access policies that are audited and reviewed on a consistent basis should be used [4]. When planning the deployment of an SDN solution, security must be a consideration from the beginning.

IV. Conclusion

This paper discusses the application of SDN in a substation environment using IEC 61850 GOOSE protocol and the benefits of SDN over traditional Ethernet networking. Specifically, it discusses the advantages of SDN over traditional Ethernet in OT applications, including fast failover, network visibility, network engineering, and cybersecurity.

V. References

[1] M. Robuck, "Google Leans Heavily on SDN for Reliability, Velocity and Availability of Its Network," *FierceTelecom*, December 2018. Available: https://www.fiercetelecom.com/telecom/google-leans-heavily-sdn-for-reliability-velocity-and-availability-its-network.

[2] Q. Yang, D. Keckalo, D. Dolezilek, and E. Cenzon, "Testing IEC 61850 Merging Units," proceedings of the 44th Annual Western Protective Relay Conference, Spokane, WA, October 2017.

[3] R. Meine, "A Practical Guide to Designing and Deploying OT SDN Networks," proceedings of the Power and Energy Automation Conference, Spokane, WA, March 2019.

[4] M. Dargin, "Secure Your SDN Controller," *Network World*, January 2018. Available: https://www.networkworld.com/article/3245173/secure-your-sdn-controller.html.

VI. Biographies

Dalton Shaffer received his BS in engineering with an electrical specialty emphasis in 2005 and an MS in engineering systems in 2007, both from the Colorado School of Mines. After graduation, he worked for engineering consulting firms focusing on industrial systems and power generation. He then joined City Light & Power, Inc. as a senior engineering team lead, where he oversees substation and distribution system engineering and design from concept through final field commissioning. Dalton has been a registered professional engineer since 2010 and is registered in multiple states.

Devin Thewlis received his BS in electrical engineering from the Colorado School of Mines in 2015. Upon graduation, he began working for City Light & Power, Inc., first as an electrical engineer and then as an automation engineer.

Tom Bartman joined Schweitzer Engineering Laboratories, Inc. (SEL) in 2006 as an engineering technician for industrial system products. He is now an application specialist in communications. Prior to joining SEL, he served in the U.S. Navy as an electronics technician with an emphasis on avionics and secure communications. After leaving the Navy, he worked for Harris, Inc. as an electronics engineer in the broadcast communications division. He has a degree in applied computer science, is a member of the Information Systems Security Association (ISSA) and International Information System Security Certification Consortium (ISC)^2, and obtained his Certified Information Systems Security Professional (CISSP) certification in 2013. Tom holds a patent for validation of arc-flash protection.

Tristan Atiyeh received his BS in electrical engineering from Widener University in May 2018. Prior to graduation, Tristan worked at Schweitzer Engineering Laboratories, Inc. (SEL) as an application engineer intern. Upon graduation, he began working for SEL as an associate application engineer. Tristan currently holds his Engineer In Training (EIT) certification.

© 2020 by City Light & Power, Inc. and
Schweitzer Engineering Laboratories, Inc.
All rights reserved.
20200128 • TP6946-01