Simplifying Teleprotection Communications With New Packet Transport Technology

David Dolezilek, Colin Gordon, Dwight Anderson,
Steel McCreery, and William C. Edwards Jr.
Schweitzer Engineering Laboratories, Inc.

Presented at the
PAC World Africa Conference
Johannesburg, South Africa
November 12–13, 2015

Originally presented at the
5th International Scientific and Technical Conference: Actual Trends in Development of Power System Relay Protection and Automation, June 2015

Simplifying Teleprotection Communications With New Packet Transport Technology

David Dolezilek, Colin Gordon, Dwight Anderson, Steel McCreery,
and William C. Edwards Jr., Schweitzer Engineering Laboratories, Inc.

Email: papers@selinc.com

USA

1 Introduction

Availability, reliability, and simplicity are major metrics in the world of teleprotection communications. Protection applications based on exchanging command signals once thrived in the realm of analog signal and dedicated point-to-point communications links. Due to the trend of combining all communications on an Ethernet network (a shared-bandwidth communications medium by design, illustrated in the relevant IEEE standards), protection signals are being published over a network that was never meant to offer message delivery deterministic enough—or deterministic at all—for mission-critical applications.

A new Ethernet packet transport technology, software-defined networking (SDN) and its open-source protocol incarnation, OpenFlow™, promises to revolutionize the ways that traffic engineers design, build, operate, and maintain critical networks. OpenFlow promises improved performance on Ethernet networks via granular control over Layers 1 to 4 of the Open Systems Interconnect (OSI) model. It also promises to give network engineers the ability to abstract teleprotection communications out of the Ethernet world and back into the realm of dedicated virtual circuits without sacrificing simplicity, flexibility, and reliability. OpenFlow-enabled Ethernet hardware further promises the ready availability of inexpensive, nonproprietary hardware that can be modeled and controlled like a software application programming interface (API) and the elimination of the need for most proprietary Ethernet management protocols.

This paper provides a description of the requirements for protection-class Ethernet networks (PCENs) and the benefits of using SDN technology over traditional networking.

2 Implementing End-User Requirements for PCENs Using Traditional Networking Technologies

End users of mission-critical communications networks typically prioritize speed, reliability, maintainability, dependability, and security.

2.1 Speed

The transfer time specified for an application is the time allowed for a signal or data exchange to travel through a communications system. IEC 61850-5 illustrates transfer time (shown in Fig. 1) as the time duration between the action of communicating a value from the logic processing of one device to the logic processing within a second device as part of an application [1]. Transfer time includes the time to execute the communications processing algorithm in both the source and destination device and the transit time. Designs for PCENs do not influence source and destination device behavior and affect only the message delivery via network middleboxes and links. The designs are concerned with the transit time, t_b, which is the time duration for the message to travel through the communications network.

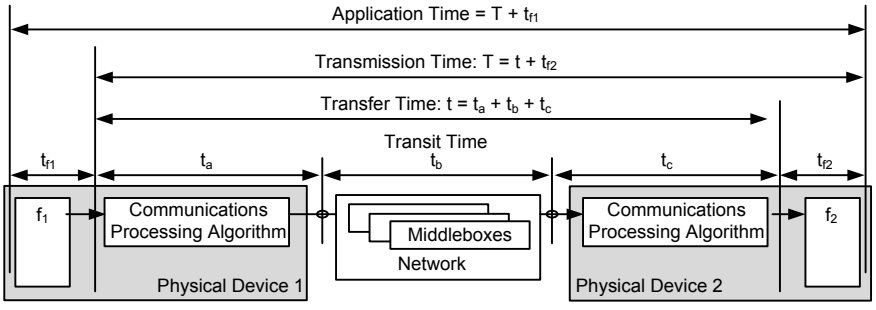

Fig. 1. Transmission time and transfer time illustration based on IEC 61850-5 [1]

IEC/TR 61850-90-4, Communication Networks and Systems for Power Utility Automation – Part 90-4: Network Engineering Guidelines clarifies many performance and test requirements [2]. Of note, it simplifies the discussion of transfer time requirements by documenting time classes for different types of messages and their associated transfer times, as shown in Table I. Using the IEC/TR 61850-90-4, network engineers can accurately specify and design local-area networks (LANs) to satisfy a transfer time class without the need to understand the underlying protection and automation applications.

Table I
Transfer Time Class Applications and Requirements Based on IEC/TR 61850-90-4 [2]

Transfer Time Class	Transfer Time (ms)	Application Example
TT0	>1,000	Files, events, and log contents
TT1	1,000	Events and alarms
TT2	500	Operator commands
TT3	100	Slow automatic interactions
TT4	20	Fast automatic interactions
TT5	10	Releases and status changes
TT6	3	Trips and blockings

2.2 Reliability, Dependability, and Security

The IEC/TR 61850-90-4 technical report defines latency of communication as the delay between the instant that data are ready for transmission and the moment they have been completely received at their destination(s) [2]. IEC 61850-5 describes the traffic recommendations specific to IEC 61850 [2]. It does not identify the other necessary traffic on the IEC 61850 Ethernet network for maintenance, telephony, video surveillance, and so on.

IEC 60834 requirements for reliability, dependability, and security guide engineering design criteria [3]. Networks must be available to deliver messages (reliability), to accomplish operations (dependability), and to prevent unwanted operations due to error or intrusion (security). From the recommendations for IEC 61850 Ethernet traffic in IEC 61850, it can be seen that Generic Object-Oriented Substation Event (GOOSE) communication used for protection has the highest priority and the shortest maximum delay. Control blocking schemes, via GOOSE or any other method, require a 99.99 percent success rate (dependability), and direct control schemes require a 99.9999 percent success rate of receipt of digital messages (reliability). Direct tripping, via the delivery and processing of a GOOSE or other message, is typically expected to occur within 20 milliseconds [3]. Failure is defined by the absence of the message at the receiving end or, for direct control, a delay in delivery greater than 18 milliseconds. Therefore, IEC 61850 Type 1A, Performance Class P2/P3, as part of a communications-assisted protection scheme, requires that the system meet the 3-millisecond transmission time 99.9999 percent of the time (identified as TT6 in Table I) and have a delay of no longer than 18 milliseconds for the remainder.

3 Implementing the Ethernet Network Between IEDs Performing Digital Signaling

The speed, reliability, dependability, and security of mission-critical communications-assisted applications are directly affected by the intelligent electronic devices (IEDs) exchanging signals, referred to as Physical Device 1 and Physical Device 2 in Fig. 1. However, the devices that make up the communications network, referred to as middleboxes (illustrated in Fig. 1), have as much or more impact on packet delivery and therefore the performance measures of speed, reliability, dependability, and security for these applications. The scope of this paper is to discuss current and new developments in the network technology available to measure and improve the speed, reliability, dependability, and security of the transport of Ethernet packets through the network, with speed measured as the transit time in Fig. 1.

From the previously mentioned standards for teleprotection security and reliability comes a number of related requirements for the Ethernet network or middlebox hardware itself.

3.1 Minimal Failure Recovery Times

Applications require fast and seamless communications recovery times both for middlebox hardware and Ethernet link failures. Traditionally, spanning tree algorithms (STAs) in each middlebox share data via Rapid Spanning Tree Protocol (RSTP) (particularly IEEE 802.1D-2004), which publishes Bridge Protocol Data Unit (BPDU) packets to communicate with other Ethernet middleboxes. This exchange provides information used by the STA to identify duplicate paths to a network address and forces one into hot standby because only one link can be active at a time. This same method identifies and logically breaks data flow loops by deactivating links and putting them in hot standby mode. Also, dual active connections to IEDs are not possible. When an IED is physically dual-connected, an STA disables one link and forces it into hot standby mode. When a middlebox failure (bridge death) or link failure (link loss) occurs, an STA executes in each middlebox to optimize a current RSTP network via statuses received within BPDUs published from other middleboxes. Next, each STA determines how to enable hot standby paths to reconfigure the network around the failure. Parts of the network that are affected by this reconfiguration may be unavailable to deliver packets during the transition period or period of network darkness. The ladder topology ensures fast reconfiguration times, but ring topologies take much longer [3]. During reconfiguration after a failed middlebox or link, the ladder topology with standardized and fast STA recovers every non-root bridge failure scenario in less than 15 milliseconds. It even resolves some root bridge failures in this time and resolves others in just slightly longer. However, any ring topology larger than three middleboxes will not reconfigure in less than 15 milliseconds [3]. Test and measurement of a four-node ring of middleboxes with traditional STA performance reveals transit latencies that grow from hundreds of milliseconds to tens of seconds.

3.2 Minimal Network Transit Latency Times

End users typically specify “latency low enough to satisfy the application” and leave the definition and design up to the network engineer. A common error is that engineers will assume that any Ethernet topology will satisfy latency and they will choose topologies based on low cost and convenience such as a ring. However, this ignores the first important step of engineering design where transit times for GOOSE messages and Sampled Values (SVs) messages must be less than 1 millisecond [1]. This transit time of 1 millisecond, combined with the duration of the subscriber communications processing algorithm illustrated as t_c in Fig. 2, must aggregate to a transfer time of less than 3 milliseconds, as shown in Table I. The overall message transit time through networks constructed as small rings and large and small ladder topologies is less than 1 millisecond when a small number of middleboxes exists between the GOOSE producers and consumers. Transit time is so small that it is often not measureable without very precise equipment. However, as rings grow in size, transit time grows as well and will eventually be longer than 1 millisecond when the number of middleboxes in the path between each GOOSE producer and consumer grows large enough. The challenge is that this change in latency is due to the ingress, egress, and switching delays associated with packets passing through each middlebox. When the number of middle boxes in the path cannot be predicted in advance, the system requires thorough testing of every message exchange scenario. An important benefit of the ladder topology is that all possible middlebox configurations are predictable and measurable to match results found during research of the design. Transit latency is calculated by aggregating the ingress and egress port delays on each middlebox plus the switching time within each middlebox the packets pass through. The time for packets to traverse all physical links is added. The physical link time is calculated as the speed of light through fiber over the link distance.

3.3 Per-Link and Per-Host Bandwidth Calculations

End users often require engineered calculations for bandwidth provisioning of application communications traffic, including the bandwidth consumption per host completed during the design stage. Protection engineers must calculate per-link bandwidth using per-host bandwidth calculations to ensure that link bandwidth capacities are not exceeded and thereby reduce the chances of dropped protection-related packets. In the ladder topology, all traffic is segregated so that the only traffic allowed on each ladder segment is the collection of messages required for the IEDs and devices on that segment. This filtering and blocking of all unneeded traffic at the links between middleboxes prevents unnecessary packet processing and link oversubscription and saturation, which can lead to packet transit delays. Ring topologies force all traffic through each and every link, which creates a lot of unnecessary packet processing and eventually leads to oversubscription and link saturation, resulting in unacceptable packet transit delays. However, even with correct bandwidth provisioning in

ladder and ring topologies, the nature of shared bandwidth links makes it impossible to guarantee performance. Other topologies are even more unpredictable and nondeterministic. Ethernet provides Class of Service (CoS) and prioritized packet processing, but it cannot provide quality of service, guaranteed latency, or determinism, especially because bandwidth usage varies so widely during both normal and abnormal circumstances.

3.4 System Performance Testing

In order to guarantee performance of the protection system, traffic engineers are tasked with both testing the system under normal loading with real or simulated traffic and under stress by emulating various failure scenarios. For all nonladder topologies, this process is extremely time intensive and manual because almost all PCENs are tailored to specific end-user requirements and there are no test examples available to show the runtime behavior without all physical hardware and links present. Also, the performance characteristics of the systems change each time new traffic or new nodes are added, which also requires new testing.

When these tests are performed on ladder networks, the measured times match calculated transit times, which confirms that the calculation method can be used without testing to simplify the verification process. Also, ladder behavior does not change when new traffic or new nodes are added to other LAN segments. Rings are tested with worst-case traffic conditions, but each link must be measured because they cannot be calculated. Unlike ring behavior, the performance of ladder topologies does not change each time new traffic and new nodes are added.

3.5 Separation of Traffic by Type and Application

Shared-bandwidth networks require that all system application traffic traverse the same links and that different traffic types be logically separated to further ensure the reliability and predictability of the PCEN. In some mission-critical applications, designs still specify that supervisory control and data acquisition (SCADA), synchrophasor, and engineering access communications be isolated onto a physical link separate from protection traffic in order to remove the possibility of protection packets queueing due to collisions. This is done by creating a second, smaller LAN for GOOSE only and separating it from the combined traffic LAN via IEEE 802.1 virtual LANs (VLANs). Traditional Ethernet does not support the separation of Layer 3 SCADA, engineering, synchrophasor, and maintenance traffic because they all share the same Ethertype for IP. When this capability is needed, complex middleboxes with routing capabilities must be added and configured to ensure the separation of Transmission Control Protocol/IP-based (TCP/IP-based) communications.

3.6 Maintainability

Internal end user information technology (IT) staff often know the methods for business IT networks but not the requirements for mission-critical operational technology (OT) for PCENs. Unfortunately, users of Ethernet for OT often become aware of their lack of in-house Ethernet OT networking skill after systems fail in service. At that time, IT staff become involved but are not prepared for building and maintaining PCENs with OT requirements. OT designers and IT staff should collaborate during the specification and design of systems. Also, end users often rely on contractors to build the networks, but they need the ability to maintain the networks themselves in the future and check system statuses, add or remove devices and hardware, or troubleshoot the system. Therefore, end users require that PCENs be as easy to maintain, understand, and troubleshoot as possible. Technologies used to implement PCENs should be nonproprietary to prevent reliance on specific individuals or a single manufacturer for success. Post-implementation concerns should not be understated; simple tasks, such as replacing a middlebox, become extraordinarily expensive and time intensive if the system operation is not well understood. This, in turn, leads to poor reliability and performance of the protection system, which relies on this new and difficult to maintain communications network.

3.7 Cybersecurity

Due to the criticality of PCENs, most end users view cybersecurity as important for overall system reliability. Due to the growing threat surface for multilevel cyberattacks, as well as growing regulatory pressure, cybersecurity should be considered equally important to application reliability and security. PCENs need to protect the availability, performance, integrity, and the confidentiality of information transferred as packets for OT and IT functions. Securing PCENs against growing malware threats over a long period of time is a difficult task that requires constant attention to intrusion prevention systems (IPS) and the added complexity of processor-intensive deep-packet inspection (DPI) devices.

4 Introducing SDN

SDN essentially allows networks to be managed as a single asset, giving network operators extremely granular levels of control over network functionality while simultaneously abstracting the complexity into a more traditional and functional programmatic interface. The effects of the abstraction and granular control are the simplification of the operation of the network, the ability for continuous monitoring in more detail, and holistic, centralized network control over the programming of individual middleboxes.

The fundamental shift in networking brought by SDN is the decoupling of the systems that decide where the traffic is sent (i.e., the control plane) from the systems that perform the forwarding of the traffic in the network (i.e., the data plane). The traditional network deployment process begins with designing the topology, then configuring the various network devices, and, finally, setting up the required network services. For traditional Ethernet PCENs, engineered to optimize speed, reliability, maintainability, and cybersecurity, the application data must flow on links determined by the various distributed STAs in the middleboxes supported by information published in RSTP packets. Messages must be designed with traffic control methods, including IEEE 802.1Q VLANs and media access control (MAC) addresses. In large networks, trying to match the network-discovered path with an application-desired data path may involve changing configurations in hundreds of devices with a variety of features and configuration parameters. All of this happens in addition to the traditional STA network monitoring to avoid loops, improve route convergence speed, and prioritize protection traffic. This complexity in management arises from the fact that each middlebox has the combination of control logic and data-forwarding logic integrated internally. For example, in traditional PCEN implementations, each Ethernet switch must run STAs and the switches exchange data in an iterative fashion to make network decisions. This takes time. Furthermore, the control plane in a traditional network is distributed among the STAs in the middleboxes, and as a consequence, changing the forwarding behavior of a network involves changing the configurations of many (potentially all) middleboxes. With respect to the IEDs, SDN eliminates the need to disable duplicate connections to IEDs and looping connections in networks. By defining the behavior of these paths, SDN allows them to be actively used simultaneously. Therefore, all Ethernet connections to the IEDs and middle boxes are used as designed and none are forced to hot standby mode.

SDN is a new architecture in networking that simplifies network management by abstracting the control plane from the data plane. Fig. 2 illustrates the building blocks of SDN, which are discussed in the following subsections.

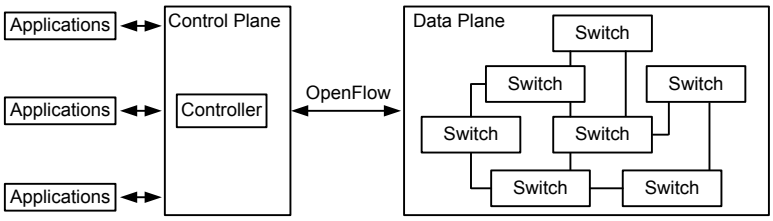

Fig. 2. SDN architecture overview [4]

4.1 Control Plane

At the heart of SDN is a controller that embodies the control plane. Specifically, controller software determines how packets (or frames) should flow (or be forwarded) in the network. The controller communicates this information to the network devices, which constitute the data plane, by setting their flow tables. This enables centralized configuration and management of a network. Many free, open-source controllers, such as that of the Linux Foundation OpenDaylight initiative [5], are readily available. Essentially, the controller manages the data flow algorithms and sends rules out to the middleboxes each time the network changes. If the network does not change after commissioning, such as in substation and industrial applications, the rules do not need to change. These rule sets are much faster, more precise, and give more control of packet flow management than iterative STAs.

4.2 Data Plane

The data plane consists of network devices that replace switches and routers. In SDN, these devices are very simple Ethernet packet forwarding devices with a communications interface to the controller to receive flow information. Many manufacturers today provide packet forwarding devices that are

SDN-enabled. These devices quickly execute rules for data flow to improve packet transit speed, reliability, availability, and dependability.

4.3 Control and Data Plane Interface

SDN requires a communications interface between network devices and the controller, as is evident from the description of control and data planes. A standardized interface between them will allow a controller to interoperate with different types of network devices and vice versa. The criterion for such a protocol is that is as widely supported as possible, with industry adoption by major switch and router manufacturers. However, it should be noted that any such protocol is simply a building block in the SDN architecture, and there are other open Internet Engineering Task Force (IETF) standards or manufacturer-specific standards that are either already available or are being developed.

5 Implementing the SDN Communications Interface with OpenFlow

OpenFlow is a protocol developed by the Open Networking Foundation (ONF) to fulfill the need for a communications interface for SDN [6]. OpenFlow enables one or more OpenFlow controllers to define the path of packets through an Ethernet network by manipulating the packet flow tables of OpenFlow-enabled hardware. OpenFlow provides four primary functions to help a controller make packet flow decisions based on packets entering the data plane of OpenFlow-enabled network hardware: matches, actions, counters, and statistics.

5.1 OpenFlow Matches

When a packet first ingresses the port of an OpenFlow-enabled network middlebox, the device examines the packet in an attempt to match fields, as illustrated in Fig. 3a, to one or more flow tables that have been defined by the OpenFlow controller(s).

(a)

Physical Layer (OSI Layer 1)			
Logical Port ID	Physical Port ID	Metadata	Tunnel ID
Data Link Layer (Ethernet, OSI Layer 2)			
Source MAC	Destination MAC	Type/ Length	VLAN ID	VLAN Priority
Network Layer (OSI Layer 3)			
IP DSCP	IP ECN	IP	IP Src Address	IP Dst Address
Transport Layer (OSI Layer 4)			
TCP Source Port	TCP Destination Port		
ICMPv4/ICMPv6 Type	ICMPv4/ICMPv6 Code		

(b)

Physical Layer (OSI Layer 1)			
Forward Logical Port ID	Set Queue ID	Forward Group ID	Set Tunnel ID
Data Link Layer (Ethernet, OSI Layer 2)			
Set Source MAC	Set Destination MAC	Set Type/ Length	Set/Push/Pop VLAN ID	Set VLAN Priority
Network Layer (OSI Layer 3)			
Set IP DSCP	Set IP ECN	Set IP	Set IP Src Address	Set IP Dst Address
Transport Layer (OSI Layer 4)			
Set TCP Source Port	Set TCP Destination Port		
Set ICMPv4/ICMPv6 Type	Set ICMPv4/ICMPv6 Code		

Fig. 3. OpenFlow possible match fields (a) and OpenFlow possible actions list (b)

If the ingressing packet does not match the present flow table, then the device can drop the packet, forward the packet to the OpenFlow controller for inspection, or forward the packet to another flow-table. If the packet matches an entry in the flow table, then the OpenFlow-enabled middlebox may then add one or more actions to the action set for the packet.

5.2 OpenFlow Actions

Various actions can be performed on packets that have matched a flow table, including copying, setting, or otherwise manipulating Ethernet/IP packet headers. The action set of the packet can also be marked to forward to other flow tables, groups of physical redundant or load-balancing ports, or metering groups for rate-limiting functions. See Fig. 3b for a list of possible OpenFlow actions. For each packet, the OpenFlow middlebox can also apply actions immediately to the packet, bypassing the current action set altogether, output the packet to a physical port, and perform other action set manipulation functions (such as clearing or updating flow tables).

5.3 OpenFlow Counters

Counters are maintained on OpenFlow-enabled middleboxes for every flow, flow table, port, group, and other point of interest. These counter data are polled by the OpenFlow controller to maintain granular data about the status of the network system as a whole.

Using OpenFlow matches, actions, and counters, an OpenFlow controller exercises complete, granular control over an Ethernet network while maintaining detailed information about the state of the Ethernet system without any additional protocols.

6 Implementing End-User Requirements for PCENs Via SDN Technology and OpenFlow

6.1 Speed and Reliability

6.1.1 Minimal Failure Recovery Times

Using the OpenFlow controller, logic is developed by network engineers to pre-engineer failover scenarios per middlebox and per link and then update the flow tables in all OpenFlow middleboxes with this precalculated logic. By using advanced failover algorithm techniques, it is possible to automate the calculation of advanced failover scenarios that would effectively reroute traffic with only the loss of the Ethernet frame that is currently on the affected link.

Therefore, appropriate engineering design based on OpenFlow creates networks in which the loss of a network link only directly affects the two OpenFlow middleboxes that are connected to it. Redundancy of links is easily performed by grouping ports together so that, on packet egress, the highest priority port currently “up” is used for the outgoing packet to the next hop or its final destination. In cases where ports are not grouped, it is possible to use logic to send a packet back out the original port from which it ingressed. This is impossible with traditional Ethernet but possible with OpenFlow and serves the purpose of redirecting traffic back into a switch for the purpose of resending it out an alternate port. With OpenFlow, using granular-enough rules, it is possible to ensure the safety of packets traversing links except for scenarios where a link fails while the packet is traversing the link itself or packets are queued in the outgoing port of the middlebox hardware.

6.1.2 Minimal Network Transit Latency Times

While OpenFlow methods alone do not automatically determine the shortest path between hosts, OpenFlow does provide the ability for network engineers to more easily implement network-specific algorithms that provide this, even for large networks. Furthermore, latency can be reduced for GOOSE message packets by implementing action logic in the very first flow table on the OpenFlow-enabled hardware, effectively minimizing the amount of logic to get the packet from one port to another. Furthermore, OpenFlow provides a built-in traffic shaping ability that can prioritize protection traffic without requiring the classical CoS IEEE 802.1Q tags.

6.1.3 Per-Link and Per-Host Bandwidth Calculations

SDN offers distinct advantages when it comes to bandwidth calculations. OpenFlow methods are capable of not just rate-limiting unicast, multicast, and broadcast traffic, but can limit any flow matching a particular flow table entry. OpenFlow is further able to take advantage of a traffic whitelist capability to simply allow only certain specific data types to be forwarded to a host on a strict preset bandwidth. A traffic whitelisting model is able to provide the most protection-based security for the link because it can simply drop unexpected traffic flow that may otherwise cause link or host availability problems. Because OpenFlow hardware can be modeled entirely in software by using emulators such as Mininet [7], bandwidth calculations of expected traffic based on predicted applications can be modeled accurately without the need for physical middleboxes, SCADA hardware, or protection devices. These calculations must next be considered in light of any additional network traffic. OpenFlow’s statistical gathering capabilities via counters also provide an easy method of retrieving detailed bandwidth data for real-time data flow analysis. The OpenFlow counter capabilities are granular enough to be able to gather data on both the hosts and the link, even under dynamic conditions.

6.1.4 System Performance Modeling and Testing

Because an SDN system can be modeled entirely in software, the OpenFlow controller, middlebox hardware, and most any traffic type can be modeled on one or more computers without hardware being involved. This technology provides the opportunity to develop tools based on these methods to fully stage, test, and simulate traffic and applications similar to the way power systems are modeled today. This will greatly speed up testing and implementation times and will be able to record more accurate numbers and prevent surprises in the field.

6.1.5 Separation of Traffic by Type and Application

The OpenFlow match/action flow-table sequences have the ability to separate traffic types to a highly granular level. As an example, OpenFlow can simply match and forward protection traffic as quickly as possible while assigning a low-priority queue to all other packets not matching protection packet

headers. OpenFlow also has the ability to physically separate out traffic by dedicating a group of redundant ports, a port group, specifically to whitelisted protection traffic and can be configured to simply drop all traffic not matching protection criteria on a particular physical link.

6.2 Maintainability

One of the promises of SDN is the decrease in day-to-day operational complexity. Because OpenFlow provides a network-wide snapshot of the system, it is easier to visualize and be alerted to any changes in the PCEN. Some modern OpenFlow controllers, such as Big Switch Floodlight [8], are attempting to integrate graphical user interfaces, with the goal of making operational changes to the network intuitive and interactive rather than a command-line task. Because network operators can update flow tables in real time, tasks that require temporarily re-architecting the network to add, remove, or upgrade hardware or firmware can be scheduled, potentially without packet loss. OpenFlow more easily supports test modes because test hardware on the network can be easily integrated by simply updating flow tables to forward test messages to specific manufacturer MAC addresses. Because network flow rules are kept in a centralized OpenFlow controller, replacing hardware is as simple as pushing out a set of rules to a new OpenFlow-enabled middlebox. Finally, the nonproprietary nature of OpenFlow holds the potential to drastically reduce operation and maintenance (O&M), documentation, and hardware costs for highly reliable PCENs through better ease of use and maintainability.

6.3 Cybersecurity

One of the most explicit SDN benefits is the ability for network operators to know exactly what communications flows should be on the network and what path they take, while giving the ability to deny all other traffic. The ability to manage communications by flow rather than packets improves end user ability to perform long-term network management. The OpenFlow protocol allows for hybrid whitelist/blacklist security models so that teleprotection communications can be detected and forwarded as quickly as possible while other traffic types can be forwarded to the controller for human alerting and approval input. Traffic matching known malware signatures can be immediately discarded. The OpenFlow traffic whitelist capability allows only specific data types to be forwarded to a host on a strict preset bandwidth. The whitelisting model provides the most protection-based security for the link because it can simply drop unexpected traffic flows that may otherwise cause link insecurity.

OpenFlow exceeds the traditional needs for an intrusion detection system (IDS) and IPS because it acts in either a reactive mode (allow wildcard communications, but send copies of flows to a centralized IDS) or a proactive mode (drop all unapproved flows). Because OpenFlow has visibility of OSI Layers 1 to 4, it can act as a traditional firewall with more granular per-header packet blocking capabilities.

7 Conclusion

OpenFlow has the potential to be vastly superior to present STA failure-handling methods because flow tables work with precalculated failover methods. SDN does not require enabling hot-standby links previously disabled by STA methods to prevent data flow loops. Also, SDN acts on link loss without disrupting large segments of a network. SDN networks can react much more quickly to disruptions than STA, which requires peer-to-peer RSTP communication with neighboring switches to determine active links.

OpenFlow can emulate the capabilities of traditional CoS tags and automatically prioritize protection traffic without requiring other priority indicators on the packets themselves, thereby reducing complexity for protection engineers. Traffic engineers can implement custom logic in an OpenFlow controller to reduce complex architectures and minimize latency, even under failover conditions. OpenFlow can be used to emulate entire environments purely in software, leading to the much quicker and easier calculation of host and network link bandwidth usage. The OpenFlow whitelisting capabilities can be used to minimize the amount of unexpected, nonprotection traffic between GOOSE hosts and prevent traffic floods that may otherwise render hosts inaccessible. While traditional rate-limiting and MAC address filtering may be used to achieve similar results, OpenFlow is more flexible and does not require manufacturer-specific protocols or features.

8 References

[1] IEC 61850-5, Communication Networks and Systems for Power Utility Automation – Part 5: Communication Requirements for Functions and Device Models.

[2] IEC/TR 61850-90-4, Communication Networks and Systems for Power Utility Automation – Part 90-4: Network Engineering Guidelines.

[3] S. Chelluri, D. Dolezilek, J. Dearien, and A. Kalra, “Understanding and Validating Ethernet Networks for Mission-Critical Protection, Automation, and Control Applications,” March 2014. Available: https://www.selinc.com.

[4] R. Bobba, D. R. Borries, R. Hilburn, J. Sanders, M. Hadley, and R. Smith, “Software-Defined Networking Addresses Control System Requirements,” April 2014. Available: https://www.selinc.com.

[5] Linux Foundation, OpenDaylight. Available: http://www.opendaylight.org.

[6] Open Networking Foundation. Available: https://www.opennetworking.org.

[7] Mininet Team, Mininet. Available: http://mininet.org.

[8] Project Floodlight, “Floodlight.” Available: http://www.projectfloodlight.org/floodlight.

9 Biographies

David Dolezilek received his B.S.E.E. from Montana State University and is the international technical director at Schweitzer Engineering Laboratories, Inc. He has experience in electric power protection, integration, automation, communication, control, SCADA, and EMS. He has authored numerous technical papers and continues to research innovative technology affecting the industry. David is a patented inventor and participates in numerous working groups and technical committees. He is a member of the IEEE, the IEEE Reliability Society, CIGRE working groups, and two International Electrotechnical Commission (IEC) technical committees tasked with the global standardization and security of communications networks and systems in substations.

Colin Gordon is an application engineer at Schweitzer Engineering Laboratories, Inc. (SEL) in the wired communications division, specializing in communications and cybersecurity solutions and services for critical infrastructure. His work experience includes secure network design, implementation, testing, and regulatory compliance consultation for utilities and asset owners in North America and abroad. He joined SEL in 2008 as a product management intern, and he holds a bachelor’s degree in computer engineering from the University of Idaho.

Dwight Anderson received his B.S. in electrical engineering from Steven’s Institute of Technology. He is now a security engineer for Schweitzer Engineering Laboratories, Inc. (SEL) in Pullman, Washington. Prior to joining SEL in 2005, he worked 20 years for Hewlett-Packard as a business development manager and systems engineer, working on projects ranging from signal intelligence systems to SCADA system programming. He is an active member of the FBI InfraGard team. He is a professional engineer in Texas and a Certified Information Systems Security Professional (CISSP).

Steel McCreery received his Electronics Engineering Technologist diploma from Humber College of AA&T in 1983. For the first 14 years after graduation, he worked in the automation industry commissioning automation projects and communications networks in the mining, steel, automotive, and pulp and paper industries. In 1999, Steel moved to the protective relaying industry as a senior technical consultant, providing consulting support on communications networks and automation to utility and industrial customers. In May 2012, Steel joined Schweitzer Engineering Laboratories, Inc. as an integration application specialist. In this role, he provides automation and communications support to utility and industrial customers.

William C. Edwards Jr. received his BSEE from the Georgia Institute of Technology in 2011. He joined Schweitzer Engineering Laboratories, Inc. (SEL) in 2011, where he is presently an automation engineer. Prior to joining SEL, he worked for the Nanotechnology Research Center, where he designed and constructed an autonomous plasma-enhanced chemical vapor deposition machine used for polymer lithographic printing. He is a member of IEEE.

Abstract—Microseconds, reliability, and simplicity are major metrics in the world of teleprotection communications. Protection applications based on exchanging command signals once thrived in the realm of analog signal and dedicated point-to-point communications links. Due to the trend to perform all communications on a single shared network, protection signaling applications are now migrating to complex, shared, queued-packet architectures that were never meant to offer deterministic message delivery for critical systems. The performance of these applications when done via Ethernet packet methods is often unmeasured and usually inadequate.

Command signals cause remote devices to trip remote circuit breakers either directly (direct tripping) or only after the remote device enables the function (permissive tripping). Other protection schemes involve tripping prevention by the local protection device (blocking). The command signals must be received at the remote end in the shortest possible time, and interference on the communications channel must never cause unwanted operation of the protection.

New packet transport software-defined networking (SDN) offers improved performance to network engineers by bringing teleprotection communications back into the realm of dedicated virtual circuits, without sacrificing simplicity, flexibility, determinism, or the ready availability of inexpensive, nonproprietary hardware. This paper provides an understanding of what SDN is and how it works. A description of the process for building and designing teleprotection networks reveals the benefits that are derived from using SDN technology over traditional networking.

There are many ways that SDN is preparing to disrupt traditional, corporate-based information technology (IT) when implemented in critical infrastructure:

- Ease and simplicity of planning, design, and testing stages for greenfield and existing architectures.
- Change control and scalability of the post-commissioning phase.
- Pre-engineering of virtual communications circuits and performance metrics.
- System-wide visualization and supervision.
- Integrated deny-by-default cybersecurity model for the entire network.

This paper explores the typical pitfalls of designing and managing teleprotection networks and compares them to the benefits of using an SDN approach versus a traditional IT approach.

20150422 • TP6697