A Comparative Study Between Software-Defined Networking and Traditional Ethernet Switches Applied to IEC 61850 GOOSE Messaging

Tarek Kaddoura, Muhammed Sheraz, and Kotb Eldeihey
Schweitzer Engineering Laboratories, Inc.

Presented at the
20th Annual GCC CIGRE International Conference
Manama, Bahrain
November 11–13, 2024

A Comparative Study Between Software-Defined Networking and Traditional Ethernet Switches Applied to IEC 61850 GOOSE Messaging

Tarek Kaddoura, Muhammed Sheraz, and Kotb Eldeihey, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—The evolution of operational technology (OT) communications has undergone a profound transformation due to the emergence and widespread adoption of software-defined networking (SDN). This technological paradigm shift has produced significant transformative changes in the field, reshaping the landscape of network infrastructure. SDN, with its network-programming capabilities, has played a pivotal role in revolutionizing OT communications by enabling deterministic data flow control and establishing predetermined recovery paths. Through the strategic utilization of SDN, networking reliability, performance, resilience, and robustness for critical infrastructure have experienced unprecedented enhancements, ushering in a new era of efficiency and dependability.**

An integral component significantly enhanced by this evolution is the IEC 61850 Generic Object-Oriented Substation Event (GOOSE) messaging protocol, which serves as a cornerstone in the design and implementation of efficient protection and control systems. This protocol enables high-priority multicast messages for peer-to-peer communication, offering flexibility for fast data transfer while maintaining system integrity and efficiency.

GOOSE messages operate through multicast communications, generating repetitive broadcast messages within the network whenever there is a data change. The repetition period typically begins at a minimum time, usually set in milliseconds, and gradually increases to a longer maximum time, such as 1 second. This ensures timely dissemination of critical information across the network.

Efficient flow optimization of IEC 61850 GOOSE messages across Ethernet networks is imperative for ensuring both efficiency and reliability. Traditional networks relying on the Rapid Spanning Tree Protocol (RSTP) encounter challenges during failover events, in which RSTP convergence time introduces delays in time-critical protection and control signals. In contrast, an SDN Ethernet network significantly enhances failover times, reducing them to less than 1 millisecond and thereby bolstering the reliability of GOOSE messages for time-critical applications.

This paper presents a comprehensive comparative study, drawing upon actual experiments, to assess the performance of conventional RSTP-based Ethernet networks versus SDN implementations. Evaluation criteria encompass packet delivery metrics and failover recovery performance across various network failure scenarios, shedding light on the comparative advantages and limitations of each approach.

***Keywords*—Software-Defined Networking - Network - Generic Object-Oriented Substation Event - Rapid Spanning Tree Protocol - Fast Recovery - Healing Time**

I. Introduction

To achieve better operational performance, the power system infrastructure of today demands modern solutions to overcome current shortcomings. Communications-assisted protection schemes depend on reliable and fast Ethernet communications to ensure safety and reliability. These systems are less tolerant of dropped packets and require swift data delivery. Generic Object-Oriented Substation Event (GOOSE)-based protection systems in substations are the prime example, in which a protection relay, upon detecting a power system fault, publishes a GOOSE Ethernet message and should promptly reach another protection relay to clear the fault. Delays in message delivery can cause damage to power system equipment and pose danger to nearby working personnel; this is mainly due to increased current flow into the fault. Therefore, Ethernet-based substation communications for protection and control systems, particularly IEC 61850 GOOSE messaging, should be reliable, fast, and deterministic [1].

Conventional Ethernet protocol, Rapid Spanning Tree Protocol (RSTP), is ineffective in meeting these requirements because it relies on the spanning tree algorithm (STA). The ineffectiveness is mainly because of the time RSTP takes to converge, which introduces delays in transmitting critical protection and control signals. During Ethernet topology changes, such as link failures, RSTP heals the network by finding new paths to deliver messages to their destinations. RSTP provides redundancy by focusing more on network healing rather than packet delivery during faulted network conditions, leading to potential packet drops during network reconfiguration and link restoration [2]. The recovery time of RSTP networks can vary substantially, ranging from a few to hundreds of milliseconds, depending on network size and complexity. This delay not only leads to the loss of numerous messages but also causes a temporary disruption in critical protection functions, particularly in systems like an arc flash where high-speed communications is essential for rapid response and system safety. The performance of RSTP is deemed acceptable in information technology (IT) applications, where latency and failover are generally less critical. IT networks often handle noncritical traffic like email, printing, and file sharing [3].

These challenges are addressed by software-defined networking (SDN), which helps in achieving the necessary performance levels for Ethernet-based communications-assisted protection schemes. SDN ensures that all communications with protection relays are purposefully engineered, guaranteeing a high level of reliability. It allows proactive engineering of network flows, including primary and secondary flows, to achieve the desired predictable and repeatable behaviors that are essential for critical protection systems [4]. The standout feature of SDN is its deny-by-default architecture and rapid failover times, reducing the times to under 1 millisecond, which makes it vital for time-critical applications of protection and control systems. This swift response enhances system reliability and ensures uninterrupted operations and system safety, especially in high-speed communications environments. This enhanced control ensures that critical communication pathways remain resilient and responsive, even during topology changes or link failures [5].

In [6] the authors describe how SDN is utilized to create secure, resilient, and high-performance Ethernet networks for IEC 61850-based OT-automated systems. In their project for a large generation utility, the need for fast and reliable recovery, data segregation, and enhanced cybersecurity, along with the geographic separation of three control rooms, led to the implementation of an SDN-based dual-ring topology for their substation automation system network. The paper outlines the thorough process of configuring and testing this SDN network. Based on these findings, the authors advocate for the use of this technology in all Ethernet communications-based OT applications to develop robust, secure, and resilient networks. In [7] the authors delve into the core principles of OT SDN and its benefits for OT networks. They present a method for designing and deploying an OT SDN network to harness these advantages. The paper covers design processes and phases, such as requirement gathering, topology design, and path planning, along with engineering considerations including automation, validation, and testing.

This paper explores applying SDN to an IEC 61850 GOOSE communications network, emphasizing purpose-engineered network control and the fast failover function in GOOSE communications. It presents the comparative study between traditional RSTP-based Ethernet networks and SDN implementations, evaluating packet delivery and failover recovery performance.

II. Generic Object-Oriented Substation Event (GOOSE)

The GOOSE object within IEC 61850 is intended for high-speed control messaging applications, such as high-speed protection-blocking schemes. Messages including statuses, controls, and measured values are broadcast over GOOSE onto the network for use by other devices (peer-to-peer communications). Each outgoing message from the sender device has a text identification string (GOOSE Control Block Reference) and an Ethernet multicast group address. Receiver devices use this text identification and multicast group to identify and filter incoming GOOSE messages.

GOOSE messages get published frequently to increase the chances that the end device receives the message. The GOOSE message gets retransmitted based on the configured minimum time (Min time) and maximum times (Max time). There are two other parameters that are frequently referred to ensure GOOSE message integrity: 1) State number (stNum), it gets incremented each time the event happens and the data change detects in the GOOSE data set and 2) Sequence Number (sqNum), it gets incremented each time a GOOSE message transmits to indicate it is not an original message resulting from a data change; SqNum also gets reset each time the data change detects in the GOOSE data set.

The first GOOSE message transmission happens immediately when the sender device detects a data change in GOOSE data set element. Following a minimum time interval, the device retransmits the previous GOOSE message and sqNum increments. Subsequently, the device doubles the time interval, increments the sqNum, and retransmits the message. This doubling of the retransmit interval and sqNum incrementation continues until the transmit interval surpasses the maximum time. At this point, the device adopts the maximum time as the retransmit interval for subsequent retransmissions, as seen in Fig. 1. The cycle resets when the sender device detects another data change, leading to an increment in the stNum and a reset of the sqNum value.

GOOSE is employed for high-priority protection with the shortest allowable delay. Control blocking schemes using GOOSE or other methods demand a 99.9999 percent success rate for digital message delivery. Consequently, mission-critical applications impose a strict latency limit of 18 milliseconds for message transfer. This ensures that the total application latency remains within the 20-millisecond maximum threshold. Consequently, direct tripping, facilitated by the delivery and processing of a GOOSE message, is anticipated to occur within this 20-millisecond window as outlined in [8].

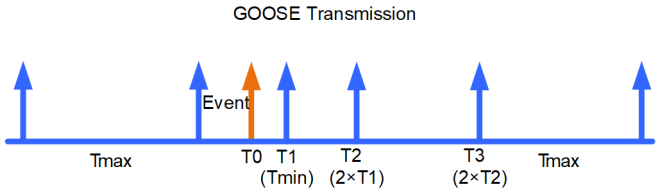

Fig. 1. GOOSE Retransmission Time Interval Behavior

III. Ethernet Network RSTP Versus SDN

A. SDN

SDN is an architectural networking concept that segregates the control plane that is responsible for deciding how to forward Ethernet frames out of the data plane comprising the SDN switches that forward the Ethernet frames and centralize it in software. SDN allows a programmatic change control platform, which allows the entire network to be managed as a single asset, simplifying the understanding of the network and enabling continuous monitoring in more detail, as shown in Fig. 2 [6]. This central software, known as an SDN controller, manages the fleet of SDN switches within its domain.

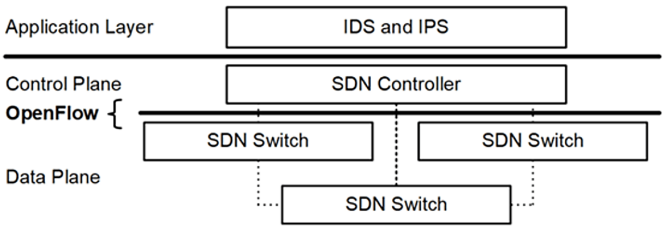

Fig. 2. SDN Architecture [6]

However, it is important to recognize that SDN remains an Ethernet technology built on the well-established and proven interoperability of IEEE 802.3. As a result, conventional Ethernet protocols and intelligent electronic devices (IEDs) that use these protocols for communications do not require any changes or modifications to operate within an SDN network. Furthermore, they cannot distinguish whether they are connected to a traditional STA network or an SDN network.

OT SDN utilizes proactively engineered flow entries for traffic matching and primary and failover paths. The OpenFlow protocol is utilized by the SDN controller to configure OpenFlow-based SDN switches, which follow a match-action scheme to manage the forwarding of Ethernet frames. When frames enter the switch, they are compared against a set of predefined rules that govern traffic flow with respect to Open Systems Interconnection (OSI) model Layers 1 to 4, as shown in Fig. 3. Based on which rule matches the Ethernet frame, it is either dropped or forwarded from the switch, and if an Ethernet frame does not match any flow entry, it is discarded, functioning as a deny-by-default filter. A flow entry represents the combination of a match and its corresponding action [9].

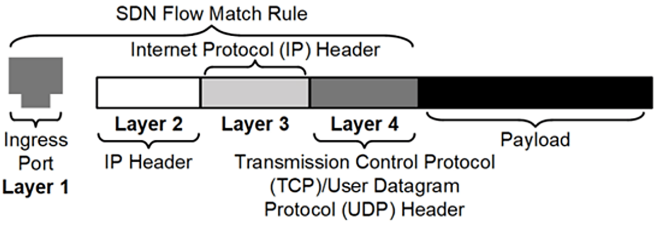

Fig. 3. Layer 1 to Layer 4 of the OSI Model [6]

B. RSTP

Each switch in conventional networks learns the media access control (MAC) addresses and locations of its surrounding devices by exploring entering Ethernet frames. The traditional switch then dynamically maintains that information in MAC tables and uses it to forward Ethernet frames to learned destinations in order for a packet to reach its destination. If a destination location is unknown, traditional switches send the packet to all participating ports, and only the intended destination is anticipated to respond to or utilize the packet. This provides plug-and-play capabilities to end-device traffic but while this gets packets to their destination, it does so at the cost of bandwidth and data exposure to any host on the network.

Additionally, since traditional switches do not do inspections on the packet level, the flow of legitimate traffic can be disrupted because the network can be flooded, leading to potential security breaches and operational inefficiencies, which, in turn, have an impact on network reliability as well.

On the other hand, given the typically static nature of control systems communications, employing OT SDN allows for a purpose-engineered approach. This involves proactively designing network flows and their redundant paths to anticipate potential traffic routes, including primary and failover paths. Such meticulous planning ensures the desired predictable and repeatable behavior necessary for effective control systems operations.

Another significant change introduced by SDN implementation is the removal of STA. As a result, dynamic topology discovery and loop mitigation convergence behaviors are no longer required.

RSTP is primarily used in traditional networks to address looping in ring network topologies. While it facilitates redundancy by disabling ports, this approach reduces switch efficiency.

C. Comparison of Traditional and SDN OT Networks

However, in the case of SDN, the flood-by-default Ethernet frame-forwarding method used in traditional networking is replaced with deny-by-default functionality. This eliminates loops in the network, without deploying RSTP. This is because the SDN switches do not have a convergence time since the failover path is predetermined and functions immediately after any network disturbance without the necessity of the control plane convergence being in the picture. Root bridge elections are not needed to be processed when a link or switch fails. As mentioned in [5], in some situations, healing occurs in less than 100 microseconds [5] [3].

For example, assume a catastrophic situation occurs when the network link fails at the same time a trip or an arc is detected. Then, assume that protection is achieved with GOOSE communications. The first packet may be lost, but the SDN device heals in less than 100 microseconds, so the next packet that is transmitted 4 milliseconds later will be received and the availability will be retained. A summary of a few points of comparison between conventional and OT SDN networks are in Table I.

TABLE I COMPARISON OF CONVENTIONAL AND OT SDN NETWORKS

Aspect	Conventional Network (RSTP)	OT SDN Network
Ethernet Frame Forwarding	MAC addresses dependent	Flow match action
Failover time	Topology, Number of nodes, RSTP other parameters dependent	Can reach less than 100 microseconds in some cases
Cybersecurity	Allow-by-default (blocklisting)	Deny-by-default (allowlisting)
Failover behavior	Dynamic (reactive)	Static (predetermined)

IV. Test Setup

A. Hardware Setup

The test setup is put in place where three switches are connected in a ring topology, as shown in Fig. 1. The GOOSE Transmitter IED (GOOSE TX), connected with Switch-A, transmits GOOSE messages to the GOOSE Receiver IED (GOOSE RX) connected with Switch C. The network switches are configured to make primary and backup paths as shown in Fig. 4.

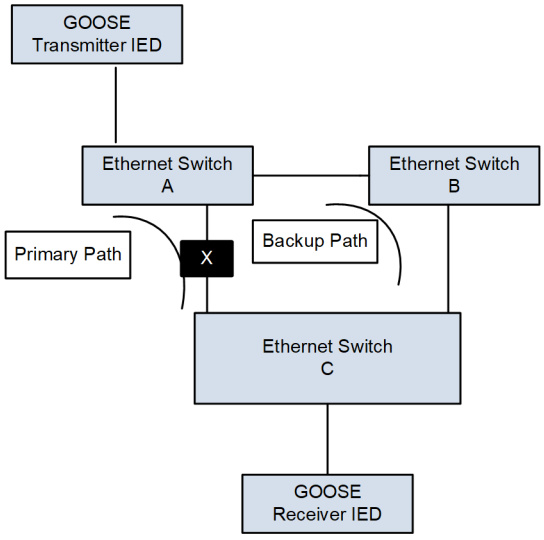

Fig. 4. Network Setup With Primary and Backup Paths, Used in Both SDN and RSTP Setups

B. GOOSE Configuration and Transmission Patterns

The GOOSE transmitter device is configured such that when initiated the GOOSE (using the pushbutton on the device), starts toggling the GOOSE bit and transmits the new GOOSE message every 16 milliseconds or less. It keeps this process of GOOSE toggling for few seconds before reaching steady state where GOOSE toggling stops and the last GOOSE message is transmitted every maximum-time interval. The minimum time and maximum time of GOOSE is set to 4 milliseconds and 1 second, respectively. The pattern of the GOOSE publishing from transmitter to the receiver device is shown in Fig. 5.

As can be seen in Fig. 5, every GOOSE message publishes three times before it toggles. First the transmission occurs immediately when the GOOSE bit toggles at T0, causing the stNum to increment and reset the sqNum to zero. The second transmission happens after a delta time of T1=4 ms (minimum time), it increments the sqNum to 1 but keeps the stNum same. Similarly, the third transmission happened after a delta time of T2=8 ms (2ominimum time), causing it to increment the sqNum to 2. The next transmission is expected after a delta time of T3=16 ms; however, the GOOSE bit toggles and resets the complete cycle. This toggle basically depicts that the transmitter device detects the power system event.

Each different color represents the toggling or change in the GOOSE bit from transmitter device. Failure can happen anywhere in-between these rapid status changes, as represented by X in Fig. 5. Delta time is the time difference between two consecutive packets. All times are given in milliseconds.

The GOOSE toggling repetition is selected to be less than 16 milliseconds, as it is related to personnel safety during an arc-flash incident. It is considered safe if the network recovers within 16 milliseconds; however, if it takes longer, the risk increases and the exposed energy levels become more harmful and unsafe, as shown in Fig. 6 [3]. In addition to that, it is also inline with IEC 61850-5 identification of Type 1A GOOSE Trip, as elaborated in [10], “IEC 61850-5 identifies [Type 1A GOOSE Trip as the most critical fast message in the substation] that perform[s] high-speed automation, protection and interlocking to meet or exceed a transmission of 3 milliseconds as Type 1A, Performance Class P2/P3.”

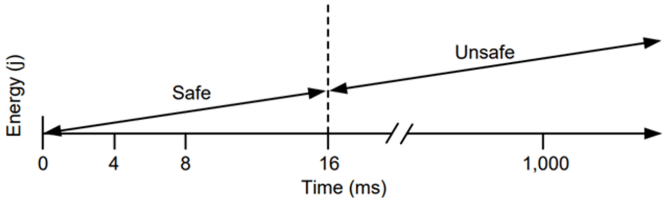

Fig. 6. GOOSE Transmit Intervals [3]

Furthermore, causing the network link failure within 16 milliseconds of the power system event detection will simulate the catastrophic situation in which the network failure and the power system event happen approximately at the same time.

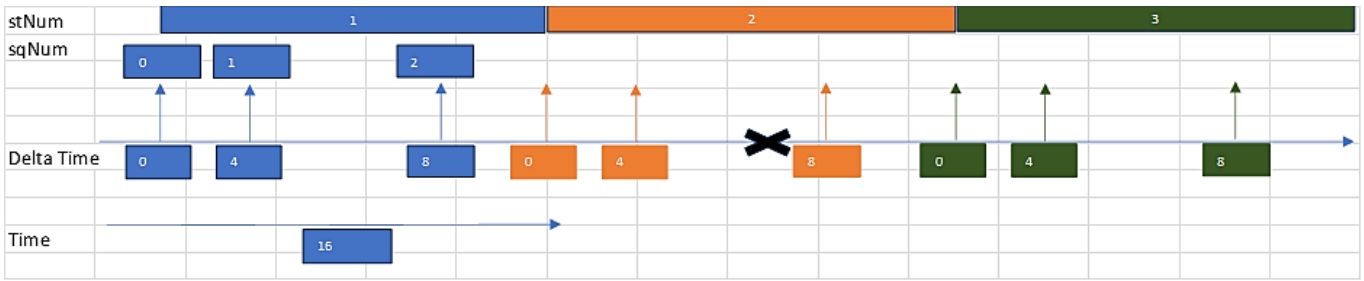

Fig. 5. Status Number, Sequence Number, and GOOSE Bit Toggling and Repetition Time Intervals in the Test Setup

C. GOOSE Traffic With No Link Failure

GOOSE traffic is captured using Wireshark in normal condition, that is without primary link failure. Wireshark capture is shown in Table II. The first three packets, shown in green, (stNum 1925), are the steady-state transmissions of the last GOOSE bit before it toggles at the delta time of 1 second (maximum time). The middle packets, shown in pink, depict the GOOSE bit toggling at T0 and its first (T1=4 ms) and second (T2=8 ms) transmission, with sqNum 0, 1, and 2, respectively. This process of GOOSE toggling will continue until it reaches a steady state. Note that not all middle packets are shown due to a large number of GOOSE bit changes and transmissions. The final packets, shown in green, represent the last status change and its steady-state repetitions at a maximum time of 1 second. Table II depicts the ideal situation where no GOOSE message is lost or dropped.

Note that column named as "No." represents the capture number of Wireshark for all Ethernet traffic, and it has a missing number because Wireshark capture is filtered to show only the GOOSE messages, so its sequence is not related to the GOOSE packets sequence. The Time column represents the total time passed. The Delta Time column includes the time differences between the shown packet and the previously shown ones. The Protocol column represents the Ethernet protocol, and the stNum and sqNum columns include GOOSE status numbers and sequence numbers, respectively.

TABLE II GOOSE TRAFFIC WHERE NO GOOSE MESSAGE IS LOST OR DROPPED

No.	Time	Delta_Time	Protocol	stNum	sqNum
1894	56.15857	0.999999	GOOSE	1925	514
1927	57.15878	1.000202	GOOSE	1925	515
1960	58.15858	0.999805	GOOSE	1925	516
1983	58.91045	0.751869	GOOSE	1926	0
1984	58.91448	0.004031	GOOSE	1926	1
1985	58.92256	0.00808	GOOSE	1926	2
1986	58.93444	0.011875	GOOSE	1927	0
1989	58.93852	0.004084	GOOSE	1927	1
1991	58.94657	0.008051	GOOSE	1927	2
...					
3137	66.23443	0.015604	GOOSE	2231	0
3138	66.23851	0.00408	GOOSE	2231	1
3139	66.24657	0.00806	GOOSE	2231	2
3140	66.25842	0.011854	GOOSE	2232	0
3141	66.26249	0.004068	GOOSE	2232	1
3142	66.27059	0.008105	GOOSE	2232	2
3143	66.28657	0.015974	GOOSE	2232	3
3146	66.31853	0.031966	GOOSE	2232	4

V. TEST RESULTS

To evaluate the network resilience, the primary path link is disconnected when the toggling of the GOOSE bit is happening and the GOOSE messages are not in a steady state; this will direct the GOOSE traffic to flow from the backup path, as shown in Fig. 4.

The test was first conducted with conventional RSTP Ethernet switches and then with SDN switches to compare the GOOSE performance over both networks.

A. Link Failure GOOSE Traffic in Conventional RSTP Switches

In the case of a conventional RSTP network when the primary link is broken, it was noticed that there are a number of GOOSE messages dropped. As can be seen in Table III, after the first transmission (sqNum 1) of stNum 384 there was a packet lost until stNum 399. This period depicts the healing time of the RSTP network, where it is converging and finding the backup path to transmit the GOOSE message. However, during the healing period there are several GOOSE messages that were lost, and this loss of message could cause misoperation and safety hazards.

TABLE III GOOSE TRAFFIC WITH PRIMARY LINK FAILURE IN THE CONVENTIONAL RSTP NETWORK

No.	Time	Delta Time	Protocol	stNum	sqNum
1678	43.82348	0.012141	GOOSE	381	0
1679	43.82737	0.003891	GOOSE	381	1
1680	43.83541	0.00804	GOOSE	381	2
1681	43.85141	0.015999	GOOSE	382	0
1682	43.85533	0.003914	GOOSE	382	1
1686	43.86331	0.007988	GOOSE	382	2
1687	43.87533	0.012019	GOOSE	383	0
1688	43.87939	0.004056	GOOSE	383	1
1689	43.88732	0.007932	GOOSE	383	2
1690	43.9035	0.016177	GOOSE	384	0
1691	43.9074	0.003904	GOOSE	384	1
1706	44.29946	0.392057	GOOSE	399	0
1707	44.30353	0.004071	GOOSE	399	1
1708	44.31143	0.007899	GOOSE	399	2

B. Link Failure GOOSE Traffic in SDN Switches

In the case of SDN, there was no packet loss during link failure and the stNum and sqNum patterns remain same as in the steady state (or in the ideal case shown in Table I) case; please refer to Fig. 7. To further demonstrate this finding and for better presentation of test results with SDN, the following criterion is devised. Due to the large size of Ethernet capture and for better presentation of the test results with SDN needed to extract data from Wireshark that show the last status

repetitions before toggling starts, the status change, its first repetition, its second repetition of the message during the toggling of its status until reaching the last status change, and four of its repetitions.

We used the following filter in Wireshark.

"goose && (goose.sqNum==113 or goose.sqNum==114 or goose.sqNum==115 or goose.sqNum==0 or goose.sqNum==1 or goose.sqNum==2 or goose.stNum==1001)" to show the following:

1. The last status repetitions before toggling started (sqNum 113, 114, 115 of stNum 681).
2. Status change, first, and second repetitions (sqNum 0, 1, 2) of the first two status changes (stNum 682, 683) and the last two status changes before toggling finishes (stNum 999, 1000) and their first and second repetitions.
3. The last status after toggling is finished and its four first repetitions (sqNum 0, 1, 2, 3, 4 of stNum 1001).

The resultant filtered capture was exported in CSV and imported to Excel. Rows 10 to 956 were hidden to avoid a large number of packets being represented; however, the count of the stNum difference and the hidden rows (after dividing by 3, as each status number is repeated three times) are equal to 315. This shows that zero packets lost happened during link failure between switches.

	A	B	C	D	E	F
1	No.	Time	Delta_Time	Protocol	stNum	sqNum
2	1190	2:23:03.649	0	GOOSE	681	113
3	1223	2:23:04.649	1.000037	GOOSE	681	114
4	1256	2:23:05.649	0.999968	GOOSE	681	115
5	1274	2:23:06.181	0.531879	GOOSE	682	0
6	1275	2:23:06.185	0.003985	GOOSE	682	1
7	1276	2:23:06.193	0.008136	GOOSE	682	2
8	1277	2:23:06.205	0.011875	GOOSE	683	0
9	1278	2:23:06.209	0.004152	GOOSE	683	1
10	1279	2:23:06.217	0.007985	GOOSE	683	2
956	2497	2:23:13.793	0.011901	GOOSE	999	0
957	2498	2:23:13.797	0.003946	GOOSE	999	1
958	2502	2:23:13.805	0.007984	GOOSE	999	2
959	2503	2:23:13.817	0.011973	GOOSE	1000	0
960	2504	2:23:13.821	0.004079	GOOSE	1000	1
961	2505	2:23:13.829	0.008088	GOOSE	1000	2
962	2506	2:23:13.841	0.012028	GOOSE	1001	0
963	2507	2:23:13.845	0.00381	GOOSE	1001	1
964	2508	2:23:13.853	0.008047	GOOSE	1001	2
965	2509	2:23:13.869	0.01593	GOOSE	1001	3
966	2513	2:23:13.901	0.032196	GOOSE	1001	4

Fig. 7. GOOSE Traffic With Primary Link Failure in the SDN Network

VI. Conclusion

This paper emphasizes the impact of SDN on OT communications, particularly in the context of the IEC 61850 GOOSE messaging protocol. The use of SDN can significantly optimize network reliability, performance, and resilience, setting a new standard for efficiency and dependability in critical systems. This paper compares traditional RSTP-based Ethernet networks with SDN implementations, and the test results highlight the superior performance of SDN in terms of packet delivery and failover recovery. The findings demonstrate that the ability of the SDN to reduce failover times to less than 1 millisecond enhances the reliability of GOOSE messages, ensuring prompt and accurate transmission of time-critical protection and control signals. These insights pave the way for further advancements in OT communications, advocating for the broader adoption of SDN to meet the evolving demands of modern network infrastructure.

VII. References

[1] IEC 61850-5, *Communication Networks and Systems for Power Utility Automation* – Part 5: Communication Requirements for Functions and Device Models, (2013).

[2] Q. Yang and R. Smith, "Improve Protection Communications Network Reliability Through Software-Defined Process Bus," (proceedings of the Grid of the Future Symposium, October 2018).

[3] M. Hadley, D. Nicol, and R. Smith, "Software-Defined Networking Redefines Performance for Ethernet Control Systems," (proceedings of the Power and Energy Automation Conference, Spokane, WA, March 2017).

[4] *Software Defined Networks: A Comprehensive Approach*, second edition, Morgan Kaufmann, 2016.

[5] A. Habeeb Moinudeen and A. Ali, "Software-Defined Networking Reinforces Security and Performance in Operational Technology Communications Networks," (proceedings of the GCC Power 2019 Conference & Exhibition Muscat, Oman, October 2019).

[6] A. Kalra, D. Dolezilek, J. Monzi Mathew, R. Raju, R. Meine, and D. Pawar, "Using Software-Defined Networking to Build Modern, Secure IEC 61850-Based Substation Automation Systems," (proceedings of the 15th International Conference on Developments in Power System Protection Liverpool, UK, March 2020).

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

[8] D. Bekker, T. Tibbals, and D. Dolezilek, "Defining and Designing Communications Determinism for Substation Applications," (proceedings of the 40th Annual Western Protective Relay Conference, Spokane, WA, October 2013).

[9] R. Bobba, D. R. Borries, R. Hilburn, J. Sanders, M. Hadley, and R. Smith, "Software-Defined Networking Addresses Control System Requirements," (April 2014).

[10] S. Chelluri, D. Dolezilek, J. Dearien, and A. Kalra, "Design and Validation Practices for Ethernet Networks to Support Automation and Control Applications," (proceedings of the Power and Energy Automation Conference, Spokane, WA, March 2014).

VIII. Biographies

Tarek Kaddoura is a project engineering in the Engineering Services division at Schweitzer Engineering Laboratories, Inc. He is located in the Khobar, Saudia Arabia office. His email address is *tarek_kaddoura@selinc.com*.

Muhammed Sheraz is an engineering manager in the Engineering Services division at Schweitzer Engineering Laboratories, Inc. He is located in the Khobar, Saudia Arabia office. His email address is *muhammad_sheraz@selinc.com*.

Kotb Eldeihey is an engineering manager in the Engineering Services division at Schweitzer Engineering Laboratories, Inc. He is located in the Khobar, Saudia Arabia office. His email address is *kotb_eldeihey@selinc.com*.

Previously presented at the 20th Annual GCC CIGRE International Conference in Manama, Bahrain
20250114 • TP7185