Case Study: Implementing Redundant Logic Controllers in Substations

Sagar Dayabhai
CONCO Energy Solutions

Brian Waldron and Marcel van Rensburg
Schweitzer Engineering Laboratories, Inc.

Presented at the
Power and Energy Automation Conference
Seattle, Washington
March 3–4, 2020

Case Study: Implementing Redundant Logic Controllers in Substations

Sagar Dayabhai, *CONCO Energy Solutions*
Brian Waldron and Marcel van Rensburg, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—The introduction of next generation smart grid technologies and intelligent electronic devices (IEDs) has increased the level of integration and information that exists in a digital substation. This advent has propelled the use of central logic controllers to allow system operators to safely monitor and operate the power system. This information is intelligently processed and analyzed by these controllers to improve power system reliability and availability and facilitate operations and maintenance.**

In addition to performing core supervisory control and data acquisition (SCADA) functions and processing, modern central logic controllers are equipped with advanced automation features designed to perform logic, arithmetic, and complex algorithms. These functions are employed to facilitate station-wide interlocking, measurement comparisons, power flow summations and load flow, power plant control functions, live busbar transfer routines, distribution automation sequences, and more. The criticality and dependency of the central logic controller in a substation automation system often necessitate the implementation of redundant controllers to enable a fault-tolerant design. There has been significant conversation in the power system industry about communication redundancy protocols, when redundant IEDs are needed, and cost effectiveness. There has, however, been comparatively little conversation about how redundant controllers are implemented and how they coordinate information collection and synchronize control logic between the redundant controllers. This paper covers the efficient and reliable methods of collecting information for logic controllers, coordinating information and the decision-making processes between controllers, and how that information is sent to the SCADA master station. The advantages and disadvantages of this implementation experience are documented through a case study that covers what worked well and the challenges encountered during the design, commissioning, and maintenance phase of the substation.

I. Introduction

The role of the data concentrator or remote terminal unit (RTU) in the substation has evolved substantially over the last three decades. What once simply collected field I/O and transmitted the status and analog values back to the control center now manages data collection from a wide variety of sources, alarm annunciation, engineering access, data visualization, and other additional functions. Because these devices have access to so much information from the system around them, they become a natural location for making decisions that require input from multiple sources. These devices have extensive logic written to control various power system assets and change operating modes of individual intelligent electronic devices (IEDs). Just as redundant protection IEDs are important to protect valuable power system assets in the same way, information availability and control decisions have also become important in critical systems that require redundancy. This has driven the need for replacing the traditional single data concentrator with redundant substation controllers. There are many technologies for communication redundancy that keep information available to controllers so they can make communications-assisted decisions. While redundant communication methods are important, they provide only a portion of the system that provides redundancy in control algorithms. This paper examines the methods and techniques used to deliver information to substation logic controllers. It then describes how substation controllers manage duplicate information sources, selects a primary controller to make control decisions, and explains how to send information back into the system without duplicate information from the redundant controllers.

Creating a robust redundant controller solution is not a one-size-fits-all approach. The redundancy solution is often heavily dependent on the communications infrastructure available in the system. The solution can be divided into several categories, but this paper focuses primarily on three logical communication sections:

- Controller-to-IEDs
- Controller-to-controller
- Controller-to-control center/supervisory control and data acquisition (SCADA)

The communication mediums and technologies used in these sections affect how the controllers can implement redundancy. The technology selected for each of these communication paths does not necessarily need to be the same to create a reliable solution. In practice, there are often significant differences between these communication paths. Each path impacts how the redundant controllers manage information and their control decisions. This paper covers technologies and methods to create a robust, reliable redundant controller solution with lessons learned from recent installations of redundant controllers.

II. Importance of Redundancy

The systems used in mission-critical applications such as renewable energy facilities and transmission substations usually require more than 99.9 percent availability. In transmission networks, it is unacceptable to have large network disturbances that can threaten the stability of the power system. These networks operate several mission-critical applications including station-wide interlocking, use of IEC 61850 GOOSE for protection and control, plant control, condition-based

monitoring, substation automation, load-shedding schemes, etc. Furthermore, the information that these SCADA systems exchange with control centers through substation gateways and logic controllers is considered mission-critical; it provides awareness and visibility to system operators in real time.

In the case of a renewable energy facility, the highly variable nature of such resources as wind and solar power make it essential for such facilities to operate on a maximum availability factor of the plant. Unnecessary or preventable plant outages affect not only the ability of the facility to provide power at its maximum capacity, but also directly impact the revenue-generating capacity of the plant.

Because of the critical nature of these facilities, it is common for substation designs to incorporate a high-availability architecture. This includes redundancy in IEDs, logic controllers and substation gateways, SCADA infrastructure, networking equipment, cabling, etc. In the context of logic controllers, this redundancy is achieved by considering both hardware and software within a logic controller.

To address hardware failures, it is common to introduce a second redundant logic controller to improve the reliability and availability of a system. This redundancy accommodates failures in power supplies, communication ports, hard-disk drives, and various embedded ancillary components that may compromise system functionality in the event of a failure. In the case of software, redundancy in a logic controller is achieved by considering several aspects including the following:

- Synchronization of databases
- Telecommunications paths
- Redundancy management between the logic controllers
- SCADA communication to end devices

III. Redundancy Communication Paths and Methods

The primary purpose of this paper is not to discuss communication redundancy, but delivering data to controllers is an important part of the system. This section briefly covers several commonly used technologies that offer redundancy in delivering data to controllers.

A. Parallel Redundancy Protocol (IEC 62439-3 PRP)

PRP is a protocol used to achieve communication path redundancy on a substation Ethernet local-area network (LAN) for mission-critical communications. PRP solutions duplicate a message on two independent networks on separate Ethernet ports on the sending device. The receiving end, that must be a PRP-compliant entity, accepts the message arriving first and discards the other [1]. The risk with using PRP is that individual PRP paths are not monitored by the sending and receiving devices, and only after both paths have failed does the user become aware of a network failure. PRP is mostly applied on substation LANs, but not as popular on wide-area networks (WANs) because of the duplicate network device requirements, which is costly.

Logic controllers support PRP in order to accommodate fault-tolerant independent communications networks to be able to communicate to a substation IED LAN with mission-critical protocols such as IEC 61850 GOOSE. These applications typically have the logic controller configured to control processes such as load shedding, remedial action schemes, etc. This makes it an essential component of the solution. The logic controller can also be responsible for SCADA communications in such an application that requires it to be connected to a separate WAN.

B. Software-Defined Networking (SDN)

SDN was first used in the information technology (IT) industry for applications such as data center networks and software-defined wide-area networking. However, the SDN architecture and its many features (e.g., network traffic programmability and network statistics) can provide innovative solutions to networking challenges in other industries. Operational technology (OT) SDN does not change the basic architecture of SDN. It is rather a method of applying SDN to solve unique challenges in automation networks composed of devices like switches and programmable logic controllers [2].

SDN provides a solution for detecting network failures and measuring network path latency, which is essential for monitoring mission-critical communications such as IEC 61850-9-2 solutions. In these solutions, it is critical for the IEDs to publish and subscribe to the data streams with as short a delay as possible to meet performance and high signaling requirements. SDN provides a fully programmable solution for using one network infrastructure to provide multiple paths for information exchange. Dedicated communication paths can be designed and monitored based on latency, which ensures the most efficient use of physical network infrastructure. It is also a natively better cybersecure solution with a deny-by-default design because the SDN requires configuration before it operates [2].

The SDN is managed and configured on the Ethernet switch, so logic controllers can use SDN solutions without requiring any special protocol support. This allows the logic controllers to achieve redundancy without having to dedicate a specific Ethernet port for a single function. However, if the hardware supports the usage of a second Ethernet port, this increases the reliability in the case of a port or cable failure.

C. Rapid Spanning Tree Protocol (RSTP)

RSTP, defined in the IEEE 802.1D-2004 standard, is a protocol used by Ethernet networking devices to detect, isolate, and restore network paths between RSTP-supported devices without human intervention. RSTP requires the engineer to design Ethernet networks with RSTP devices arranged in multiple paths for information to be sent between IEDs [3]. Typically, IEDs do not partake in RSTP network topologies, but they rather connect to RSTP-based Ethernet network switches, with the IEDs as edge devices.

Considering applications where logic controllers are connected to networks using RSTP, the least costly path time for information exchange should be measured. This must be used for protocol-based time-out settings and must also be taken into consideration for critical solutions, such as load-shedding schemes and interlocking, to ensure that coordination

is maintained. The challenge with RSTP is that the engineer should design redundant paths with placement and connections of supported Ethernet devices to reliably determine which path the Ethernet switches will use for each network failure location. Failover time in RSTP networks is in the range of tens to hundreds of milliseconds [4].

IV. Controller-to-IEDs

When there are redundant controllers in the system there is a desire to have both controllers run the exact same configuration and have seamless transfer between the two controllers. However, this is not always possible depending on how information is available to the two controllers. IEDs with information to be collected, may only have a single serial port available for controller communications, other devices such as a human-machine interface (HMI) or other data collection services may also restrict the number of communication sessions available to the redundant controllers. This section discusses methods and previously implemented solutions for managing and collecting information from IEDs.

A. Controllers Collecting Data Simultaneously

If system IEDs allow both controllers to collect information from the IEDs at the same time, redundancy configuration is often simplified. No additional logic is necessary in the controller to determine when information collection should be active. This also means each controller has the latest information and latency in data collection and decision making should be minimal for a transfer of control between the controllers.

B. One Controller Collects Data at a Time

Additional logic is necessary when only a single controller can collect information from the source IEDs, but this typically adds no significant complication to the logic of the system. The controllers must know when the system should collect information and turn off collections services. However, the redundant controllers often already have logic that determines which controller is primary or active. This indication can be used to tell the controller when information collection should occur. This often just results in a few additional lines of code to turn protocol collection on and off. One advantage of collecting information from only one controller at a time is that it simplifies the connection to the SCADA master station and eliminates the need for event management. Only one controller has events to report to the SCADA master station at a time, and no logic or controller feature is necessary to manage events previously sent. While this is an advantage, it may not be worth the several disadvantages of this approach. Because there is only one communication path to the IED, any interruption of that communication path causes both controllers to be unable to collect that information. The time to switch between controllers increases because each protocol communication interface must initialize and start collecting data. This interrupts data to SCADA and any control algorithms that are using those data.

An alternative to minimize the impact of a single communication interface to the IEDs is to support an additional communication interface between the two controllers so that the primary controller passes the information to the backup controller. This means that the backup controller may have reduced downtime when a switch between controllers occurs, because the backup controller already has the last current state of all the information. While this alternative helps mitigate some of the issues with the single communication interface, it adds complexity to the redundancy configuration because the controllers must identify which set of information should be used in the transfer of data between the controllers. Depending on the application and system requirements, this trade off may not be worth the benefits of mitigating the disadvantages of single-path communication interfaces.

V. Controller-to-Controller

A. Determining Which Logic Controller Should Be Active

The two key aspects in a redundancy architecture is management of redundant data/events and state synchronization of the logic controller to ensure that information is reliably transported to and from the logic controller without degrading the performance of the system when a single failure occurs with any of the controllers. This setup is analogous to the setup of a cluster configuration typically used between redundant firewalls.

State synchronization information refers to internal system data used in the redundancy scheme to allow the logic controllers to synchronize the database and make the appropriate decisions when managing the active, backup, and maintenance states.

The redundant logic controllers must share a communication channel between the two controllers to keep track of the availability of each controller and determine which controller should be active and issuing control commands. The communication medium, protocol, and speed do not have significant impact when examining the logic of this decision-making process. Each of these items has an impact on the speed of failover and some affect the reliability of the communication channel. They do not, however, affect how logic decisions are made to determine which controller is active, so this section does not discuss protocol and connection method. In the initial examination of the logic, it appears that determination of which controller should be active in either processing SCADA or control algorithms is straightforward. Configure heartbeat logic between the two controllers by using either a Boolean indicator that changes on some interval or a counter value that increments on that interval. When the backup/inactive controller no longer sees the heartbeat from the primary controller, it is time for the backup controller to become the primary controller. However, the following factors can complicate this decision.

B. Lost Communications or Logic Controller Without Power

A difficult aspect of communications between two controllers is how one controller loses detection of communications, while the other controller is still active and processing data. Perhaps a detrimental event such as a hardware problem or loss of power occurred to the other controller

preventing it from performing its intended function. While the issue may look a little different in each communication medium (hardwire contact, serial, and Ethernet), it poses the same fundamental challenge: Did something happen to the communication medium, or is the controller no longer sending the heartbeat? There are generally two approaches to determining the answer. First, consider implementing multiple communication channels between the two controllers. Fig. 1 shows several different communication options.

- Topology 1 uses two independent serial links to reliably transport state synchronization data and redundant information and events. Typically, both state synchronization information and redundant data are transmitted over both links for added reliability.
- Topology 2 uses the Ethernet-based communications network and a single serial communications link to transport both state synchronization information and redundant data.
- Topology 3 uses the Ethernet-based communications network to transport both state synchronization information and redundant data. This topology uses two physical Ethernet ports to achieve separate communication paths and is reliant on the operability of the communications network.
- Topology 4 uses two separate Ethernet-based communications networks to transport both state synchronization information and redundant data. This topology uses two physically separated Ethernet ports and networking devices to achieve separate communication paths.

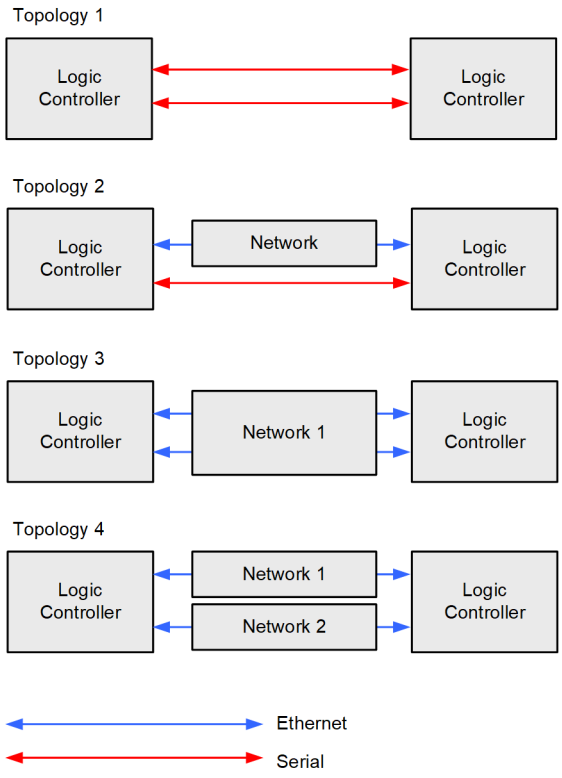

Fig. 1. Communication Topologies Between Controllers

Topology 2, that utilizes both Ethernet and serial communications, is an attractive choice to synchronize data between logic controllers. It allows for one communication medium type to fail and still maintain communications. The serial communication path can have some limitations on the amount of data or which applications can utilize that communication channel, but if something specifically affects the Ethernet communications channel it is less likely to affect the serial communications and allow synchronization between the two controllers to continue. Topologies 1 and 4 also allow for one communication method to fail and still maintain synchronization between controllers.

Another advantage of using communication channels between controllers separate from the interface that communicates with the SCADA or WAN interface is that it allows the logic controllers to detect if the WAN interface or SCADA communications are lost. This allows the controllers to switch the active state when both units are processing logic and communications between the logic controllers are good. This increases the reliability for remote communication status and controls.

The second approach involves a third device in a separate location that communicates with the two redundant logic controllers. The entire purpose of this third device is to monitor the availability of the other two devices. If one device is physically powered down it provides an additional piece of information to the other logic controller that lets it know that both the logic controller and the third device do not see activity from the powered-down logic controller. This allows the backup device to make the appropriate decision. If communications between the two logic controllers is lost, then communication with the third logic controller informs the other two logic controllers how to behave. While this method still uses communications, it alleviates a single point of failure between the two logic controllers.

In the unlikely case that either of the two approaches discussed previously completely fail, it results in the possible scenario where both logic controllers are powered on, and the synchronization logic makes both controllers active at the same time. This creates potential conflicts in the system depending on other communications the controllers have to IEDs and SCADA in the system. There are three possible approaches; allow both controllers to be active, turn both controllers off, or program only one controller to be active if communications are lost.

1) Both Controllers Are Active

In this scenario the default synchronization logic wants to turn both controllers on. It is unlikely that most system operators prefer this behavior. Having both controllers active can send conflicting controls to IEDs, send multiple data responses to SCADA, or send conflicting data to other IEDs if both units are not processing the same data.

2) Turn Both Controllers Off

Because it is difficult for the controller to detect if a loss of communications is the result of a hardware failure or only a communication failure turning off, both controllers during a

loss of communications defeats the purpose of redundant controllers in the case of hardware failure of a single controller. In this case, no application control logic would be sent to IEDs or no communication to SCADA.

3) Default to a Single Controller

Select a single controller to be active if communications between devices is lost. In this mode, the system behavior is desirable if both controllers are operating as normal, but lose all communication between controllers. Depending on system conditions, control decisions to IEDs are maintained, as well as SCADA communications. But, if a hardware failure occurs, it is unpredictable whether the controller selected to become active fails. Either no controller is active, or a non-default controller has hardware failure by chance and the default controller continues processing as normal.

In terms of total risk, choosing to default to a single controller provides the most constant desirable behavior when communications are lost. When hardware failure occurs, it may provide the most desirable behavior depending on which controller fails. In the option of both controllers becoming active, it always provides the most desirable behavior in the event of a controller hardware failure. However, it potentially provides the least desirable option when only communications between controllers fail. Ultimately, system operators must analyze their applications and systems to determine which trade-off is most acceptable to them.

C. Primary-Primary vs. Primary-Backup

Use of two redundancy controllers typically requires the logical decision of several operation modes. In this paper, primary-primary refers to the functionality where one controller processes the information, makes control decisions, and then issues those commands out of the box. Once the waiting controller detects that the decision-making box is unavailable, it picks up the decision-making and control-issuing functionality. If the first controller becomes available again during this time, it becomes the controller-in-waiting to make decisions and issue the commands. In the primary-backup designations, it works similarly to primary-primary, but the backup controller returns control to the original controller once the backup controller becomes active and recognizes that the primary controller has returned. Which mode to select is based on system operation preference. The primary-backup method introduces a small and insignificant amount of complication into the process of deciding which controller should be active.

D. Regaining Synchronization After Communications Are Lost

An aspect to consider is which redundant controller should be active after power is lost and then restored to a controller that causes a loss-of-synchronization between the controllers. If the mode is primary-backup, the logic to determine that controller one should be active requires no modification. If the primary controller is available it will become active. However, if the mode is primary-primary, the decision is less straightforward. Primary-primary mode operates on the condition that if the other controller is nonresponsive, then the controller that determined the other controller is unresponsive becomes active. It is best to create a voting scheme to decide which controller should be active if synchronization between controllers is lost. A simple voting scheme to determine which controller should be active is to determine which controller has been active for a longer period of time. To do this, record how long each controller has been active and exchange that information between the two controllers. It is important that the amount of active time is precise, perhaps to the millisecond if the controller is capable of this. This is important in a scenario where power to both controllers is lost and then restored simultaneously. Depending on the configuration and the behavior of the controller, both controllers can begin processing redundancy logic within the same second. It is unlikely that both units would start the redundancy logic within the exact same millisecond. This higher level of time accuracy prevents a tie in this voting scheme. This becomes a simple calculation to determine which controller should be active and covers many scenarios that result in a complete loss of communications or a loss of synchronization between the controllers.

E. Maintenance Mode

Another desired behavior when using redundant logic controllers is to implement a maintenance, or testing mode, where one controller can be removed from the redundancy scheme to allow testing of the controller, updating of the configuration, or updating of the hardware. This requires the two controllers to exchange an additional Boolean piece of information to instruct that a selected controller should not participate in a redundancy scheme and to also inform the other controller that its partner is not participating in the redundancy scheme. This is especially helpful in the primary-backup mode, where the backup mode transfers the active mode back to the primary. If the controller is undergoing testing, it will still maintain a communications channel to the remote controller to confirm its behavior.

VI. CONTROLLER-TO-SCADA

When both controllers actively collect data from IEDs, often the logic controllers still provide data concentration for communications to the SCADA master station. Often these SCADA connections either use DNP3, IEC 60870-5-101/104, Modbus, or IEC 61850 Manufacturing Message Specification (MMS). There are many other legacy or proprietary protocols (e.g., LG8979) that were commonly used for SCADA communications in the past. However, this is uncommon on newer redundant systems and is outside the scope of this paper.

If the connection to the SCADA master station uses IEC 61850 MMS, there is no need for concern with the redundant controllers sending duplicate events or data to SCADA because the IEC 61850 standard makes the client responsible for managing which information has already been received by assigning the desired entry ID when connecting to a buffered report in the MMS server.

The Modbus protocol has no provision for reported events or data with time stamp or quality. Modbus provides only present status, so the client collects just the present value of

each point in the active controller. While Modbus provides many advantages for its simplicity, it typically does not make a good SCADA protocol because of its lack of time stamps with data.

This leaves two primary protocols for redundant controllers to address synchronizing data from two controllers: DNP3 and IEC 60870-5-101/104. Both protocols support time-stamped data and reporting data changes either unsolicited or in a polled mechanism. When these protocols have sent their changes to the client, the client sends back an acknowledgment so that the servers know what data changes have been sent to the client. If the server receives no acknowledgment, the protocols retransmit the data. Each protocol defines factors that determine interval and frequency of data collection. The ability of each server to keep track of whether the client has already acknowledged data is critical for managing the duplicate data that each logic controller collects. This functionality, typically offered as a firmware feature in substation controllers, requires only user configuration. Manufacturers use different methods to manage this functionality. End users need only confirm the communications channel planned between the controllers to accommodate this functionality.

VII. Controller Ethernet Interface Considerations

Depending on the application requirements, the system may benefit from both logic controllers sharing a single Internet Protocol (IP) address. A single IP address allows both controllers to appear as a single controller inside the substation. This shared IP address is typically used on the WAN interface on the controllers. Outside systems will therefore act as if they are interacting with only a single device in the substation. Outside systems do not need to have failover detection logic or any information about the redundant system inside the substation. This provides a straightforward and simple approach for SCADA master stations.

However, sharing a single IP address for the LAN inside the substation is unlikely to provide the same advantages as sharing an IP address on the WAN interface. With the LAN it is very likely that both controllers must communicate with a variety of devices in the substation simultaneously. Sharing a single IP address forces only one controller to communicate to other devices at a time. It also makes it difficult to communicate to each controller for engineering access unless there is a separate network for this functionality. A shared IP address could be implemented on the LAN with additional unique IP aliases on each controller. However, this begins to unnecessarily complicate the network. For most redundant controller implementations, it is best to have three network interfaces with which the controller communicates:

- A WAN connection to communicate with SCADA and other systems outside the substation
- A LAN connection to communicate with IEDs in the substation, perform data collection, and send control signals out
- A connection directly between controllers to communicate all necessary information for a redundant controller solution

VIII. Experiences From the Field

This paper has covered a variety of technologies, communication topologies, and logic control decision-making discussions up to this point. When designing a redundant system for logic controllers, each of these areas must be considered and design decisions selected. The following section covers a recent redundant logic controller solution that was implemented at a utility. This case study discusses the functional and design requirements of the utility, what decisions were selected from the redundancy design previously discussed in this paper, and some challenges encountered during the implementation.

A. Redundant Logic Controller Functionality Requirements and System Design

A solution using redundant logic controllers has been implemented at a utility for use in its transmission substations. The design incorporates dual main protection and automation schemes, lending itself to the concept of a segregated control room for new transmission substations.

The segregated control room design includes areas designated for Main 1 (primary) equipment and Main 2 (backup) equipment. This design allows for the testing or complete replacement of equipment in either of the main control rooms while the primary equipment remains in service via the alternate backup equipment. The following requirements must be achieved for the redundant logic controllers solution:

- There must be a primary and logic controller that would automatically fail over to the backup controller in the event of a failure.
- The database of both logic controllers must be synchronized and maintained continuously.
- The logic controllers collate all information received and transmit this information to multiple SCADA master stations and external HMI clients.
- Control operations are managed between the logic controllers to ensure that only a single control is submitted for operation at a time.
- The redundancy scheme allows for seamless connection to the SCADA master stations in the event of a failover. The SCADA master station is unaware of the redundant logic controllers. The controllers manage the communication requests from the master station, and only the primary controller responds.
- Should the SCADA master station acknowledge events on the primary logic controller, there is no reporting of that same event when the master station begins communicating with the backup logic controller. This prevents transmission of duplicate events to the same SCADA master station.
- Both controllers must connect to main and backup protection IEDs in the substation and use the IEC 61850 MMS protocol to acquire data from all IEDs. Redundant signals are managed accordingly to ensure that duplicate signals are not transmitted to the SCADA master stations.

- The solution should support a maintenance mode embedded within the logic controllers to facilitate testing and maintenance.

B. Redundancy Design Choices

Fig. 2 shows the logical connections between the parts of the system discussed in this paper, the relays to logic controller, between the logic controllers, and to the SCADA connection.

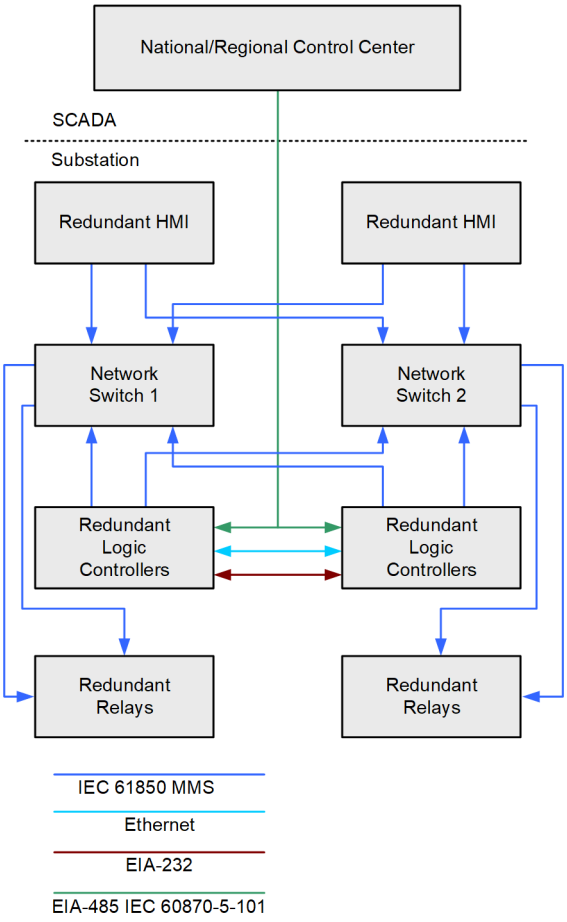

Fig. 2. Logical Communication Topology

1) Logic Controller to IEDs

In this system each of the redundant controllers collected data from the IEDs via the IEC 61850 MMS protocol simultaneously. This allowed each logic controller to have the most recent data from the IEDs. In IEC 61850 MMS it is the responsibility of the clients to keep track of what data has been collected from the IED. This relieves the relays from managing which data has been already sent to the client. Because this is a new installation and all intra-substation communications are Ethernet with modern IEDs, this allows for simultaneous data collection to the redundant logic controllers and no data collection services are required to be coordinated.

2) Communication Between Logic Controllers

The redundant logic controllers share two communication sessions between each other in this design: an Ethernet connection and an EIA-232 serial connection. This was Topology 2 from Fig. 1. Over these connections the logic controllers share primary/backup status and coordinate what information has already been sent to the SCADA system. This topology allows for a failure of the network communications while keeping logic controller synchronization active.

3) SCADA Connection

A unique aspect of this system compared to many others is that the SCADA system sends data requests to each logic controller simultaneously via a serial connection. The controllers are wired in the same way traditional EIA-485 multidrop systems are, but both controllers share the same protocol address. In addition to each logic controller keeping track of which data has already been transmitted to the SCADA system, the logic controllers must keep track of which logic controller should respond to the requests that are sent to both logic controllers. This type of topology is not common and was not covered in the fundamental portion of this paper. Usually when there are redundant logic controllers inside a substation that wish to appear as a single device to SCADA control centers the redundant logic controllers will share a single IP address.

C. Challenges During Implementation

We satisfied and implemented the previously listed requirements but experienced several other challenges. Following are a few of these challenges together with their solutions.

1) Multiple Communication Failures Between Logic Controllers

The solution incorporated a primary/backup logic controller redundancy scheme. To increase the availability and reliability of the system and mitigate a split-brain situation, we selected Topology 2 from Fig. 1 as the communications scheme between the logic controllers for primary/backup and data synchronization with Ethernet and serial connections for inter-controller communication. The logic controllers use both links to determine the primary, backup, and maintenance states. In the unlikely event that both communication links fail (Ethernet and serial communications), the primary logic controller would deactivate and the backup controller would assume the active state, and maintenance mode would be prohibited. This fail-safe scenario prevents the possibility of both controllers assuming the active state at the same time (including at startup) and both controllers attempting to make control decisions or respond to SCADA.

2) SCADA Communications

A key challenge that needed to be addressed included SCADA protocol redundancy and event synchronization between the logic controllers. The solution required using the IEC 60870-5-101 serial-based SCADA protocol on only a single controller to communicate to more than one SCADA master station (national control and regional control). The SCADA master station would be unaware of the redundancy scheme and would not be performing any failover if communications to the primary controller failed. This required that each logic controller share the same address and manage the failover. When the master station confirms the reception of an event from the primary logic controller, a message is sent to the backup controller through use of the synchronization links. The backup controller uses this information to identify which event the SCADA master station receives, acknowledges it, and recognizes the same event in the backup controller system database. This prevents transmission of duplicate events to the same master station if a transition between controllers occurs. This type of SCADA connection was challenging because the serial communications wiring did not physically allow for a mechanism for only one controller to recognize the SCADA communications like an Ethernet system provides (both units shared a single IP address). This forced additional logic in the logic controllers to determine when an IEC 60870-5-101 SCADA server should respond to a valid data request, and when it should ignore the data request based on when the logic controller was the primary controller.

If any of the controllers were in maintenance mode, the redundant controller would assume the role of primary controller and communicate with the SCADA master stations. The controller under maintenance would then suspend all communications to the SCADA master stations. This maintenance state was synchronized between the controllers and allowed for project updates and testing on the logic controller when it was in maintenance mode.

3) Managing Data From Redundant IEDs

Since this system contained redundant protection IEDs, each controller collected data from both redundant IEDs. This required each logic controller to subscribe to independent datasets and reports in the redundant IEDs. The controller then had two copies of the information it needed to send to SCADA and use in its logic applications. The controller selected an information source based on the quality and communication status of the IED. If quality was valid and IED communications were good, each controller was programmed to have a preferred data source.

4) Managing Controls to Redundant IEDs

Since this system contained redundant IEDs for protection, there was a primary and a backup IED. When the logic controller needed to send a control operation it was individually managed between the logic controllers and sent to the active IED. To ensure that authorization of the bays was correctly managed for safe operation during a failure of the active logic controller, control authority information was synchronized between the controllers.

IX. Conclusion

When designing a substation or system for which you are considering redundant controllers, it is easy to think about redundancy as a single feature in a substation controller. While substation controllers typically offer a number of features to implement redundant systems, many factors are still configured and managed by the end user. These are choices such as selecting how many communication interfaces each controller will have, how data are collected from IEDs, how duplicate data are sent to SCADA, selecting an operation scheme between controllers, and managing control logic. Each site has a variety of other design parameters that require a combination of functionalities that results in a customized solution for increased system reliability. There is no single feature or check box; the end user must instead make a series of choices to create a secure system.

X. References

[1] IEC 62439-3:2016 Industrial Communication Networks – High Availability Automation Networks – Part 3: Parallel Redundancy Protocol (PRP) and High-Availability Seamless Redundancy (HSR), 2016.

[2] 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.

[3] IEEE 802.1D IEEE Standard for Local and Metropolitan Area Networks, 2004.

[4] Q. Yang and R. Smith, "Improve Protection Communications Network Reliability Through Software-Defined Process Bus," proceedings of the Power and Energy Automation Conference, Spokane, WA, March 2019.

XI. Biographies

Sagar Dayabhai is currently pursuing his PhD at the University of Witwatersrand (Wits), South Africa in the field of smart grids. He received his BSc Eng. (electrical) degree from Wits in 2009 and earned his MSc Eng. (electrical) degree in 2014. He is a registered professional engineer with the Engineering Council of South Africa (ECSA). Sagar was one of the IEC Young Professionals elected for South Africa in 2016 and is currently a member of IEC TC57 WG10 and WG15. After working at Eskom in the field of telecommunications and SCADA, he moved to Consolidated Power Projects (CONCO) as a Senior SCADA/Automation Engineer. Sagar now holds the position of System Control Manager at CONCO Energy Solutions.

Brian Waldron is a development lead automation engineer with Schweitzer Engineering Laboratories, Inc. He has several years of experience in designing and troubleshooting automation systems and communications networks. He has authored several technical papers, application guides, and teaching presentations focusing on integrating automation products. Brian graduated from Gonzaga University with a B.S. degree in electrical engineering.

Marcel van Rensburg is currently pursuing his MS at Montana Tech in power systems. He received his BTech from Tshwane University of Technology, South Africa in 2016. He is a member of IEEE and a secretary for his local Montana IEEE section. Marcel worked as an application engineer at Schweitzer Engineering Laboratories, Inc. for 6 years and is now a product engineer.