Design and Application Considerations for User-Programmable Bits Over Protection Channels

Zachary Eyasu
Southwest Transmission Cooperative, Inc.

Bin Le, Tony Lee, Ken Behrendt, and Veselin Skendzic
Schweitzer Engineering Laboratories, Inc.

Published in the
proceedings of the 67th Annual Georgia Tech
Protective Relaying Conference as an alternate
Atlanta, Georgia
May 8–10, 2013

Design and Application Considerations for User-Programmable Bits Over Protection Channels

Zachary Eyasu, *Southwest Transmission Cooperative, Inc.*
Bin Le, Tony Lee, Ken Behrendt, and Veselin Skendzic, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—Modern digital relays use a wide variety of communications channels designed for protection applications. Communications channels can include a dedicated serial peer-to-peer communications channel, a line current differential (87L) channel used to exchange analog samples, and high-speed Generic Object-Oriented Substation Event (GOOSE) messaging over Ethernet. Communications channels are used to exchange a wide variety of status bits that can be predefined or programmed by the user. Exchanged bits are often very critical and must be protected in order to prevent protection system misoperation.**

This paper describes the methods used to detect communications errors, explains how to convert the commonly used bit error rate (BER) measurement obtained from a given communications channel into a quantitative estimate of undetected bit errors, and discusses diagnostic features found in modern digital relays. The discussion is supported with real-world application examples and further verified using mathematical analysis and long-term tests.

The paper describes the typical solutions of using timers or combining multiple bits to ensure security for very noisy channels. Disturbance detection and redundant channels provide an opportunity to dramatically improve security for critical signals, such as direct transfer tripping. Two field misoperation events are included to prove the importance of correctly applying channel addressing.

The paper reviews the preferable treatment (logic) for a number of typical bits used in line current differential applications. This discussion explains different fallback schemes when a packet is corrupted or lost. Recommendations are given for selecting the optimal protocol in the contexts of security and speed while taking into account the intended application and channel characteristics.

I. Introduction

Communications-based protection schemes are becoming an increasingly important tool for power system protection. This process is driven by many factors, the most important of which are decreasing cost and exponentially increasing communications system capacity.

Driven by consumer applications, communications networks have spread over the world, enabling unprecedented connectivity among people as well as among machines. Power system protection communications form a very small but exceptionally important subset of this worldwide data deluge.

When faced with video streaming and terabit bandwidth requirements, it is very easy to forget the basic need of ensuring the guaranteed transfer of several bits of mission-critical protection information. This paper provides a detailed look at the theory and practice behind the exchange of programmable bits using protection-quality channels. This type of exchange is well known in the power industry by the common name of *communications-assisted protection scheme*.

II. Communications Protocols for Protection and Error Detection Methods

Before taking a detailed look at methods used to ensure the protection channel data integrity, it is helpful to examine the basic communications system environment. We start with the communications medium. The most popular choices for utility applications are as follows:

- Wired communications.
- Wireless communications.
- Optical fiber.

Typical examples of each include power line carrier communications, radio and microwave radio links, and synchronous optical network-based (SONET-based) or synchronous digital hierarchy-based (SDH-based) wide-area network systems.

The data transmission schemes used on each medium strongly depend on the available link bandwidth and equipment age, but it can safely be said that virtually all systems are migrating towards native support for digital data transmission. Depending on the application, a given communications link may be dedicated (e.g., a dedicated fiber link) or shared with a number of digital communications multiplexed onto a common wide-band channel. Typical examples of multiplexed communications include time-division multiplexing (TDM), such as SONET; digital microwave; Ethernet radio; and multiprotocol label switching-based (MPLS-based) wide-area networks. Multiplexing can also be done in the frequency domain, with the most notable examples being microwave channel banks and optical wavelength-division multiplexing.

Dedicated links are easiest to use and understand, but the cost often makes it difficult to justify dedicating an entire link to a single application. Multiplexing therefore becomes a preferred application method as soon as the transmission link bandwidth becomes large enough to support multiple applications.

Multiplexed communications add a new set of considerations that must be addressed before a given communications scheme becomes usable for power system applications. Typical considerations include channel swapping (accidental cross-connection), traffic mirroring, 87L data alignment errors, and unintended loopback. Methods appropriate for increasing communications scheme security are further discussed in Section V.

The key to making communications-assisted protection and control schemes successful is to choose a secure digital communications protocol that can provide a robust channel for the exchange of the critical data. This section expands on the structure of four candidate protocols that are specifically created for such a task. Those protocols are as follows:

- A peer-to-peer asynchronous protocol.
- IEC 61850 Generic Object-Oriented Substation Event (GOOSE) messaging.
- 87L differential protection over a 64 kbps serial channel.
- 87L differential protection over Ethernet.

The basic message structure for each of the protocols is shown in Fig. 1. All four protocols contain address information, data payload, and an error-check sequence at the end.

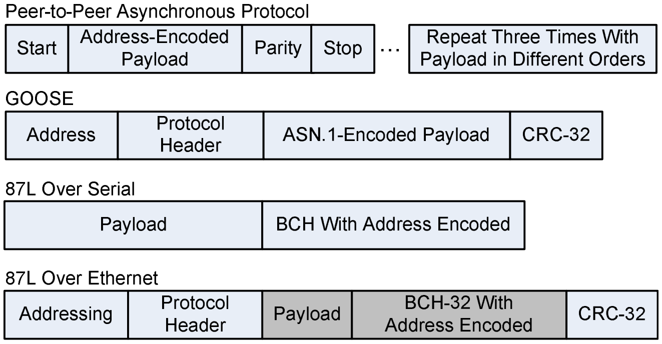


Fig. 1. Basic message structure for the four example protocols (box size is not proportional to the message length).

The first candidate, a peer-to-peer asynchronous protocol, is the simplest protocol and is explicitly optimized to enhance security on low-speed (less than 36 kbps) asynchronous serial links. In one well-accepted design, the message is very short (36 bits) and is capable of securely transmitting an 8-bit payload. To send 8 data bits with the desired security to meet IEC 60834 requirements for permissive teleprotection schemes, the message is configured to repeat all 8 data bits four times within the contents of four 9-bit characters. Each character contains 1 start bit, 6 data bits, 1 parity check bit, and 1 stop bit. The receiving device decodes the message, checks that all the start, stop, and parity bits are correct, and checks to ensure that all three copies of the 8 data bits match.

The second candidate is an IEC 61850-compliant GOOSE message that is intended for the reliable transmission of discrete state changes. GOOSE messaging is optimized for Ethernet-based local-area networks (LANs) and is typically much larger than peer-to-peer asynchronous communications protocol messages (one hundred to several hundred bytes). GOOSE message utilization (the ratio between effective payload and the associated message overhead) is very low (less than 20 percent) and depends on the type of data contained in the message. GOOSE messaging is user-configurable, relying on the Abstract Syntax Notation One (ASN.1) encoding to provide self-descriptive data types. Although not explicitly intended for this purpose, the use of ASN.1 encoding significantly enhances security because the self-describing data structure can be explicitly verified upon reception, thus further strengthening the protection provided by the Ethernet frame 32-bit cyclic redundancy check (CRC-32) code.

The third candidate, which is 87L over a serial channel, achieves a high payload-to-overhead ratio for transmitting differential current measurements between two or more relays and is explicitly optimized for low-speed (64 kbps) synchronous serial links, which are normally created using TDM multiplexers or direct point-to-point fiber. Addressing capability is achieved implicitly by encoding the information to the Bose, Ray-Chaudhuri, Hocquenghem (BCH) code field. Basically, the sending device systematically appends a special check sequence to the transmitted data. The receiving device also calculates a check sequence based on the received copy of the payload to detect errors introduced during transmission.

Message security can be enhanced by expanding the BCH field. Additional discussion showing the benefits and performance of this approach is provided in Section III.

The final candidate is represented with the 87L differential relay message (as with the third candidate) optimized for transmission over Ethernet. In this case, the serial message is simply encapsulated in a fully addressable Ethernet wrapper, which inherently appends a CRC-32 sequence. Cascading 32-bit BCH (BCH-32) code with CRC-32 provides exceptional security, going multiple orders of magnitude beyond the typical power system protection application requirements.

For the rest of this paper, we select the widely used 87L over serial frame protected by cyclic data integrity code to be the target of our discussion on the probability of undetected errors.

III. Error Detection Codes

After years of digital communications technology being developed and applied, a few block-check codes for binary code word protection stand out and have become industry standards. Among many of them are Ethernet IEEE 802.3 CRC-32 and various BCH codes.

A 16-bit BCH code was initially the choice of some relay manufacturers to guard their 87L packets against channel noise. This particular BCH code has a generator polynomial of 267543 when expressed in octal format or DED81 in hexadecimal format [1]. Previous work reported in [2], [3], and [4] states that the probability of a binary linear code not detecting a corrupted packet is bounded by $\frac{1}{2^{16}}$, which gives a generic estimation without considering the bit error rate (BER) and the pattern of error distribution. Following this theory, the probability that a corrupted code word, which includes a frame-check sequence (FCS) field in addition to the payload data, remains divisible by the generator polynomial is approximately 0.0016 percent, or less than one undetected error per 65,535 noise bursts.

From a code-specific point of view, according to [1], this particular 16-bit BCH code has a Hamming distance of at least 5 when the packet size is limited to 255 bits. In other words, all errors with up to 4 bits of data inverted are guaranteed to be recognized by this code. It provides a similar error-detecting capability to that of the prevailing Ethernet IEEE 802.3 CRC-32 polynomial, which has a Hamming distance of 6 for any data word containing up to 268 bits [5], but it can be implemented with fewer hardware resources.

When estimating the probability of undetected error under a given BER, the most conservative assumption is that all errors with 5 or more bits corrupted are undetectable. To simplify that analysis, we also assume that the noise in a digital communications channel has a binominal distribution characteristic. This means that the probability of any bit being inverted to an opposite state is equal to the BER and consecutive errors are statistically independent. The probabilities of all possible combinations of corruption in a 255-bit packet are enumerated as follows.

The probability of no bits being corrupted is shown in (1), the probability of having 1 bit corrupted is shown in (2), and the probability of having *n* bits corrupted is shown in (3).

$$C_{255}^{0} \bullet \mathrm{BER}^{0} \bullet (1-\mathrm{BER})^{255} \tag{1}$$

$$C_{255}^{1} \bullet \mathrm{BER}^{1} \bullet (1-\mathrm{BER})^{254} \tag{2}$$

$$C_{255}^{n} \bullet \mathrm{BER}^{n} \bullet (1-\mathrm{BER})^{(255-n)} \tag{3}$$

where:

C_{m}^{n} stands for the number of combinations by selecting *n* bit positions out of a group of *m* bits.

Adding together the probabilities of all the instances of error affecting more than 4 bits gives us the probability of this code not being able to identify a corrupted packet (P_{undet}). However, when the packet size grows, this method also becomes complicated. A better strategy is to sum the probabilities of having less than or equal to 4 bit errors, which yields the probability of guaranteed detection (P_{det}). Then, the probability of having an undetectable error is simply $1 - P_{det}$.

Applying a common BER value of $4 \bullet 10^{-4}$ (1 flipped bit in every 10 packets) into the calculation for the previously described 255-bit packet with a 16-bit BCH gives us a P_{undet} of $8.1 \bullet 10^{-8}$, which is much smaller (better) than the value from the generic analytical method where BER is left out of the analysis. Furthermore, keep in mind that this analysis gives the worst-case result intended for a very conservative security assessment. In reality, the detection failure rate should be even lower because most of the errors with more than 4 bits inverted are actually detectable.

The noise-triggering source in a typical utility network tends to cause burst errors, which appear as a large number of consecutive bits having the same state, either all zeros or all ones. Such a pattern is usually the result of electromagnetic interference from certain switching events. All the bits transmitted during that period of time are subjected to the same rapid change in magnetic field and are likely forced to the same logic state. The effectiveness of cyclic block code in detecting noise burst is in fact superior to its performance when challenged by scarcely scattered bit flips. It was concluded in [5] that a burst error with a duration of less than the number of error-check bits can always be detected. In other words, if the first corrupted bit and the last corrupted bit are separated by less than the number of error-check bits, the burst is always detected by the receiving device. More specifically, the 16-bit BCH code described previously will always detect all bursts of errors that span less than 16 bits. Although the undetected error rate from the preceding analysis is very low, modern protective relays publish 87L packets at a very high frequency for the sake of operating speed. A small probability of a corrupted packet going undetected multiplied by a large number of corruption attempts will eventually result in a finite number of undetected errors.

In one specific design, in order to provide subcycle fault clearing capabilities, the digital relay sends 250 packets per second. Every year, a total of $250 \bullet 60 \bullet 60 \bullet 24 \bullet 365 = 7.884 \bullet 10^{9}$ packets sent by this relay will attempt to travel through the communications link to reach its remote peer. Using the pessimistic detection failure rate of $8.1 \bullet 10^{-8}$ and multiplying it by the number of trials every year ($7.884 \bullet 10^{9}$), we estimate that about 650 packets with errors will escape the data integrity check if the communications device consistently works on the verge of failing.

In addition, consider the number of devices relying on communication to protect the assets critical to power system stability in a substation. If we assume that there are 15 such devices constantly transmitting and receiving packets at the previously mentioned rate and all involved channels operate under the same BER of $4 \bullet 10^{-4}$ 100 percent of the time, the substation protection scheme can statistically experience undetected errors $650 \bullet 15 = 9{,}750$ times every single year. We believe this number is certainly not acceptable from the point of view of a protection engineer.

IV. Error Detection Simulation and Test Result

In this section, we validate the theoretically derived probability from the previous section with a MATLAB® model as well as physical noise injection tests.

When simulating the behavior of a 16-bit BCH code as described in Section III, the data packet size is chosen as 255 bits, with 239 of them being the actual payload bits. The first step is to create the payload data by sequentially filling a 1 by 239 matrix with zeros and ones generated by the built-in MATLAB pseudorandom number generator. The last 16 bits are calculated by dividing the payload by the BCH generator polynomial. The binary division is implemented using the left-end-shift-in technique, as described in [6].

For each simulation case, another pseudorandom number generator, the output of which has a uniform distribution characteristic, randomly selects a predefined number of unique bit positions. The bits in these positions are corrupted. On the receiving end, the entire corrupted packet is divided by the same polynomial one more time. The intentional corruption is detected if the remainder of this division is non-zero. If the remainder is zero, the corruption is not detected by the receiving device and might then cause a protection problem. The same process repeats 30 million times for each simulation case.

Fig. 2 plots the rate of undetected errors from the batch execution of a MATLAB script with respect to the number of corrupted bits in every single packet. It was cited in the previous section that the theoretical probability of failure for a 16-bit binary code is approximately 0.0016 percent in a very noisy channel (multiple bits in every single packet are corrupted). The result produced by the MATLAB model supports the theory well. The trace shown in Fig. 2 also agrees with the observation made in [3], which suggests that the probability of undetected error peaks when the number of corrupted bits is slightly above the Hamming distance. When the noise inverts fewer than 5 bits, there is no recorded failure of detection. This result is as expected because all corruptions of fewer than 5 bits are detectable by the chosen 16-bit BCH.

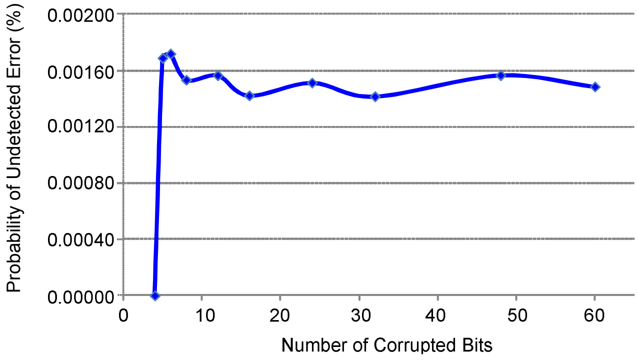


Fig. 2. 16-bit BCH MATLAB simulation result.

The simulation result in Fig. 2 also draws our attention to persistent channel noise that might be induced by some marginally failing communications equipment (more details of this phenomenon are given in Section V). In an operating environment where repeatedly there are more error bits than the Hamming distance in the arrived packets, we cannot overlook the chance that a corrupted packet will go undetected by the receiving device.

With the critical nature of protection applications in mind, we now evaluate the security enhancement of a stronger BCH code. The generic method from [2] and [3] predicts the failure rate of a 32-bit binary code to be $2.3 \bullet 10^{-10}$. Following the BER- and Hamming distance-specific analysis developed in the previous section, we determine that the probability for the data integrity check to fail can also be significantly reduced. Using the same BER of $4 \bullet 10^{-4}$, the probability of undetected error is $2.6 \bullet 10^{-15}$ for any code that features a Hamming distance of 9. Multiply this rate by the number of packets that the relay sends out every year, and it will take almost 48,000 years for the BCH check to theoretically be defeated even once.

The performance enhancement from moving to a BCH-32 was first examined using the same MATLAB model as used in the 16-bit BCH test. The payload data were reduced to 223 bits so that the overall size of the packet remained unchanged. Two levels of data corruption, randomly selecting 48 or 60 bits to invert in every packet, were explored. Each simulation case underwent 100 million repetitions, and no detection failure was found.

Next, the strength of this extended BCH code as implemented in a line current differential relay was verified by using a channel simulator that physically injects noise into the 87L channel.

As shown in Fig. 3, the two relays under test exchange 87L data over their serial interface modules. A channel simulator bridges the communications link between these two relays and corrupts every other packet in both directions. The duration of every noise burst is 224 bit times, and it happens to every other packet. Whenever the channel is impaired by noise, each bit passing through the simulator has a 50 percent chance to be inverted. If a received packet fails the BCH check, the entire packet is discarded and no data from this packet are used.

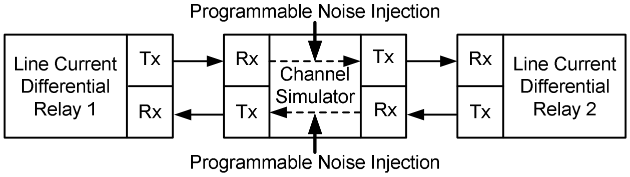

Programmable Noise Injection
Line Current Differential Relay 1
Tx
Rx
Rx
Tx
Channel Simulator
Tx
Rx
Rx
Tx
Line Current Differential Relay 2
Programmable Noise Injection

Fig. 3. Noise injection test setup.

In addition to analog samples, 4 bits per 87L channel are made available to the user for flexible high-speed digital I/O applications. The bits have all been programmed to constantly send zeros to the other end. The relay has been programmed for the sake of this test to trip unconditionally if any of those programmable bits are other than zero in an accepted (presumably uncorrupted) packet. At the time of publication, the described setup has been running in the test laboratory for over two months. No tripping event has been recorded so far, after 1.3 billion error bursts have been injected.

V. Methods to Improve Security

This section presents three tried-and-true methods that can be easily applied by the user to reduce the probability of undetected errors even further. The methods include the use of security counters, bit payload repetition, and disturbance detector-based supervision.

The discussion presented in this section is based on a well-established peer-to-peer protocol suitable for transmission of communications-assisted tripping signals (e.g., permissive overreaching transfer trip [POTT], directional comparison unblocking [DCUB], and directional comparison blocking [DCB]). The protocol uses triple-redundant error checking with additional parity and framing checks to achieve 24 total redundant bits [7]. Laboratory tests confirm that this protocol allows less than one undetected error per 10 million noise bursts, making it suitable for blocking schemes per the requirements of IEC 60834-1. However, that standard recommends the security of less than one undetected error in 100 million noise bursts for protocols used for direct transfer tripping or overreaching transfer tripping.

The devices that implement this protocol include security counters as a very powerful means to further reduce the rate of undetected errors. The security counters require reception of *n* successive messages, all with a given signal asserted, before allowing that signal to assert internally in the relay. For example, a setting of *n* = 2 requires the relay to receive two successive messages, both indicating an internal signal should transition, before the relay actually allows the internal signal to do so. This simple check doubles the amount of redundancy, from 24 redundant bits to 48 redundant bits. With a setting of *n* = 2, we expect less than one undetected error per 100 trillion noise bursts, 1 million times the security required by IEC 60834-1 for direct transfer tripping.

In practice, settings larger than *n* = 2 are not necessary and only serve to slow the transferred signal. For example, at a serial communications data rate of 9,600 bps, one message from the same protocol is transmitted every 4.167 milliseconds, or 240 times per second. When setting *n* = 1, the signal is delayed by normal channel delays that depend on the communications equipment, plus about 4 milliseconds of serializing and deserializing delay. Setting *n* = 2 adds another 4 milliseconds of delay while the relay waits for the second sequential message, for a total of about 8 milliseconds plus the channel delay. For most direct transfer trip (DTT) or POTT schemes, 8 milliseconds plus the channel delay is sufficiently fast, and the cost of increased delay to get increased security when setting *n* = 2 is justified. Larger additional security counter delays of 12 or 16 milliseconds for *n* = 3 or *n* = 4, respectively, are not justified.

In general, a security counter that requires reception of *n* successive messages delays the signals by an additional $n \cdot M$ milliseconds, where *M* is the serializing and deserializing delay when *n* = 1, and increases security by a factor of $2^{R(n-1)}$, where *R* is the number of redundant bits used by the protocol to detect errors. In the previous example, *R* = 24, *n* = 2, and the additional security count from *n* = 1 to *n* = 2 increased the security by a factor of $2^{24(2-1)} = 2^{24}$ = 16.7 million.

A much less powerful method of increasing security is to require multiple bits from the same channel to assert before allowing the received signal to transition. For example, the protocol described previously exchanges 8 programmable bits between two devices. Assume the first of 8 bits is used for direct transfer tripping. We expect about half the undetected errors to cause the first of 8 bits to be asserted, resulting in an unwanted direct trip command. In other words, half of all impaired 8-bit messages have the first bit asserted. The user can configure the receiving device to require Bit 1 and Bit 2 to assert before declaring that a transfer trip signal has been received. These two bits can appear in four possible combinations (i.e., 00, 01, 10, and 11) so that one-fourth of those bad messages will have these two selected bits set to the pattern that meets our required logic. In that case, we would expect an undetected error to result in an unwanted direct trip signal one time out of four, an increase in security by a factor of $2^{2-1} = 2$. If we require all 8 bits to be asserted to declare a received direct trip signal, we have increased security by a factor of $2^{8-1} = 128$. Compare this increase in security with the previous case of a simple security counter of two, which increased security by a factor of over 16 million.

Assume one of the programmable bits previously described is used in a DTT scheme. Another means to increase security is to build the system such that a received direct trip signal will not be used by the receiving relay unless that relay detects a disturbance on the power system. This scheme is called disturbance detector supervision, and suitable logic is shown in Fig. 4.

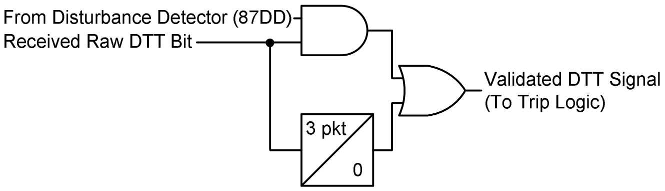


Fig. 4. Disturbance detector supervision on received DTT bit.

The received DTT signal causes an immediate TRIP output if signal 87DD is already asserted. In one design, 87DD asserts for 10 cycles after the relay detects any change in voltage or current. If DTT asserts for three packets in a row, the relay generates a TRIP output even if 87DD has not asserted. We expect the received DTT bit to assert for only a single message time due to an undetected error. In the previously described protocol, an undetected error is expected to cause DTT to assert for about 4 milliseconds, which would not cause a trip in this case if the power system was operating normally with no faults or disturbances.

Quantifying the increased security from deploying the disturbance detector supervision is more difficult than for the other methods discussed. Security from unwanted trips due to undetected errors is nearly perfect if there is no simultaneous power system disturbance. However, that security is totally lacking if there is a simultaneous power system disturbance or if the power system disturbance and channel disturbance that result in an undetected error have the same cause.

Disturbance detector supervision is especially helpful when combined with channel monitoring. Many 87L protocols include channel monitoring that can alarm for conditions ranging from a single bit error to a continuous dropout that lasts longer than several seconds and can alarm for a low-grade channel degradation that causes the BER to increase slowly over time.

Consider a fiber-optic transmitter that degrades gradually, with ever-decreasing transmit power. Eventually, the optical power that arrives at the associated receiver decreases to near the noise level, and the receiver begins to incorrectly differentiate between zeros and ones. The BER climbs. The average time between undetected errors in such a case may be thousands of years or longer. Such numbers are of little consolation to the protection engineer responsible for the relay that allows an undetected error, however unlikely, one week after the channel degradation begins.

Now, consider that same situation, but with disturbance detector supervision. Strong error detection combined with disturbance detector supervision gives the protection engineer time to react to the problem, diagnose the cause, and rectify it before the statistical number becomes a reality.

VI. Importance of Channel Addressing

Another layer of security is to ensure that the local protection and control device is routed to the correct remote protection and control device at all times. This objective can be achieved by incorporating transmit and receive addressing into the data integrity code.

In the protocol described previously, each message is also encoded with the transmit address setting of the sending device by inverting selected data bits in each message. The transmit address setting of the sending device must match the receive address setting of the receiving device for the frame to be accepted as valid. To ensure that a device cannot accept its own message (in case the communications channel is looped back to itself), the devices are designed such that the transmit and receive address settings in each device cannot be set the same. The importance of channel addressing is demonstrated in the following two real-world examples.

Fig. 5 shows an example of communications through multiplexers. The intention is for intelligent electronic devices (IEDs) IED-1A and IED-1B to communicate with each other. Likewise, IED-2A and IED-2B are intended to communicate with each other. Note that the transmit and receive identifiers (IDs) on both sets of IEDs are set to 1.

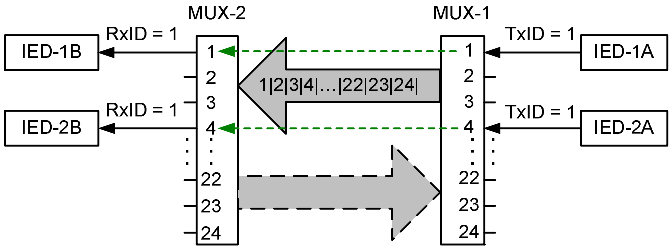


Fig. 5. IED communication through multiplexers.

The multiplexers in Fig. 5 apply TDM. This multiplexing technique takes the information from up to 24 IEDs and encodes the information from each IED in a message that sequentially orders the information into 24 time slots. On the receiving end, the message is decoded such that each sequential time slot is handed off to its respective IED connection. In this example, IED-1A and IED-1B use Time Slot 1 and IED-2A and IED-2B use Time Slot 4.

Fig. 6 shows a field case where a malfunctioning multiplexer swapped the data in two time slots. Because IED-1A, IED-1B, IED-2A, and IED-2B had matching transmit and receive addresses, IED-2B accepted the information from IED-1A and IED-1B accepted the information from IED-2A. This resulted in an undesired breaker trip.

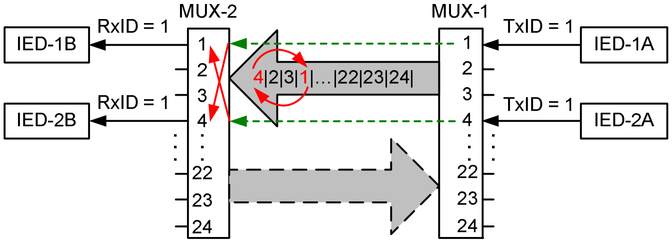


Fig. 6. IED communication through multiplexers with swapped time slots.

Fig. 7 shows the same case as Fig. 6, except that the transmit and receive IDs have been changed to make them unique. This prevents the IEDs from incorrectly receiving messages from the wrong IED.

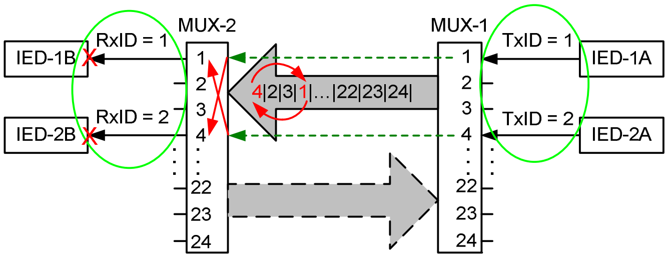


Fig. 7. IED communication with unique transmit and receive IDs.

Line current differential relays can also be susceptible to misdirected message packets. In Fig. 8, a three-terminal line is protected by line current differential relays that are sharing line current information through multiplexed communications channels. Channel X on each relay is sending its data to Channel Y on one of the remote relays.

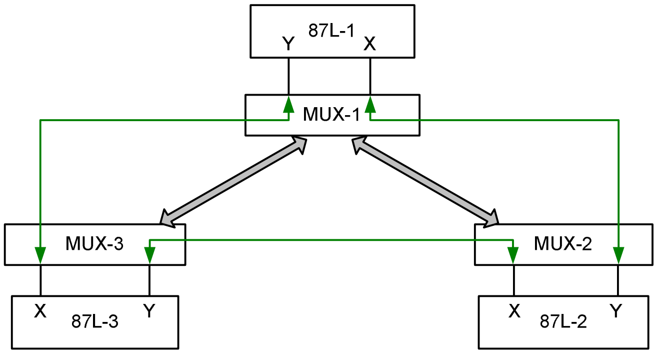


Fig. 8. Line current differential communication for a three-terminal line protection application.

During communications system maintenance, the technician hot-swapped a card in one of the multiplexers. This caused data to be incorrectly sent from Channel X to Channel Y on the same relay, as shown in Fig. 9.

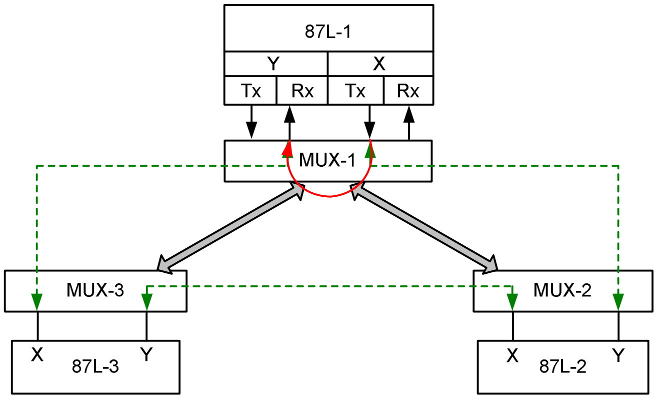


Fig. 9. Accidental loopback formed in a multiplexer (MUX-1).

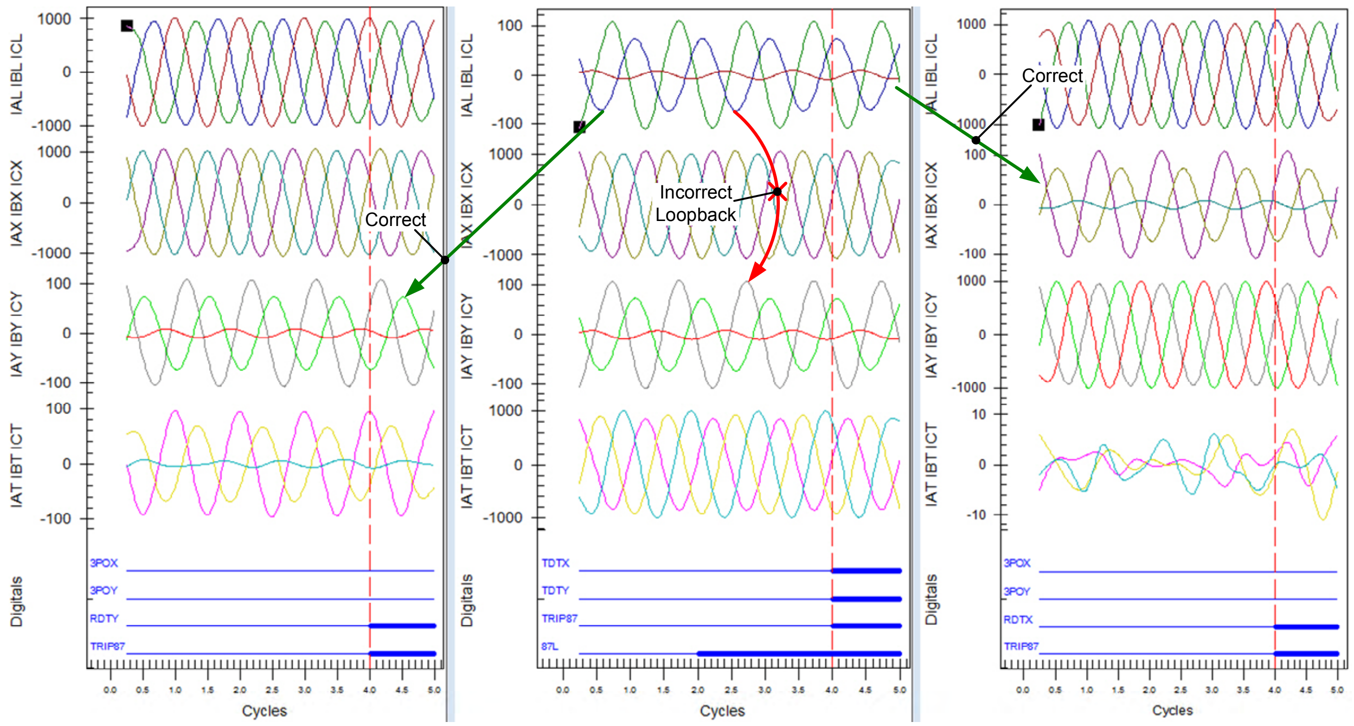


Fig. 10. Line current differential oscillography showing data loopback on one relay.

With the relay data looped back to itself, the 87L function operated and sent a transfer trip to the other two relays. The signals from all three relay terminals are shown in Fig. 10. The oscillographic data displayed in the left- and right-side panes of Fig. 10 show that two of the three relays were correctly sending and receiving the line current data. The middle pane shows that the local currents (IAL, IBL, and ICL) incorrectly appear as Channel Y currents (IAY, IBY, and ICY). These relays were installed with address checking disabled. This was subsequently corrected by enabling address checking and assigning unique addresses on the X and Y communications channels of each relay, as shown in Fig. 11. With properly applied address checking, any loopback or cross-channel condition is immediately detected by the 87L scheme and an alarm generated, preventing false tripping. This allows precious time for subsequent human intervention to correct the communications problem.

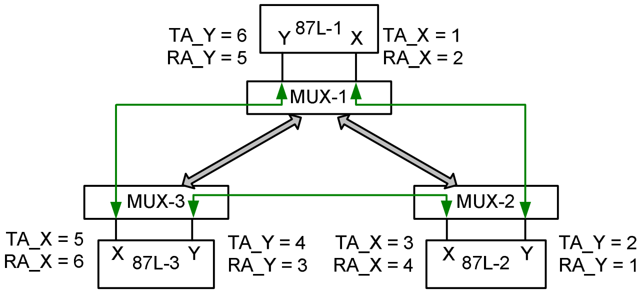


Fig. 11. 87L relays with unique transmit and receive addressing.

VII. Utility Communications System Example

The need of enhancing security against undetected data errors is justified in this section with the description of how extensively digital status bit exchange is used in typical utility protection designs. A challenging communications link set up by Southwest Transmission Cooperative, Inc. (SWTC) to carry pilot protection signals is identified to show the importance of channel monitoring as detailed in Section VIII.

SWTC is a transmission cooperative that serves customers from the southern part of Arizona near Apache Generating Station in Cochise all the way north to Bullhead City, Arizona. In addition to the transmission assets, the cooperative operates private and leased telephone circuits, fiber-optic communications links, and more than 30 microwave towers. Each segment of the communications system satisfies a particular need or application.

As many other utilities have, SWTC has standardized the concept of using an intelligent integrated protection and control scheme that relies on peer-to-peer communications and pilot protection technologies. This concept improves reliability, reduces cost, and provides advanced substation and transmission line equipment protection.

A typical peer-to-peer application deployed at SWTC substations includes a breaker failure scheme, relay cross-tripping (e.g., transformer relay trips involve the feeder relays), close blocking, reclose initiation, a remedial action scheme, a fast bus differential scheme, a main and transfer bus scheme, circuit switcher status report, and a motor-operated disconnect switch and circuit switcher control as well as breaker disconnect switch control.

Pilot relaying at SWTC is unique in that the channel needs to meet very stringent and seemingly contradictory security and operating speed requirements. A typical pilot relaying scheme at SWTC includes line current differential, POTT, and DTT schemes. SWTC makes use of both fiber and microwave channels to achieve the pilot protection scheme.

Fig. 12 shows the protection design and communications network layout between the Morenci and Hackberry substations.

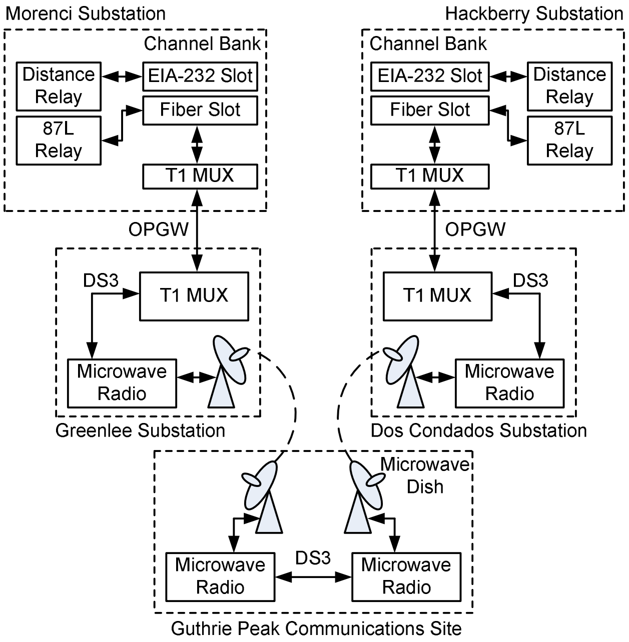


Fig. 12. Morenci-to-Hackberry substation communications path.

The 230 kV transmission line connecting these two substations is protected by a line current differential scheme as the main protection and a POTT scheme as the backup. Both the 87L and distance relays first interface with a channel bank. The channel bank merges the individual data streams and routes the traffic via a T1 (1.544 Mbps) channel to a fiber-optic multiplexer (MUX). The OC-12-capable (622.08 Mbps) multiplexer collects the data from all connected channel bank circuits before it moves the interleaved signals to the adjacent substation through an optical ground wire (OPGW) cable.

For any data to travel from Morenci to Hackberry, the data need to pass through three intermediate locations (Greenlee substation, Guthrie Peak communications site, and Dos Condados substation). Restricted by the cost of running fibers over mountainous terrain, SWTC employs digital microwave radio technology (DS3) to establish the communication between either one of the two pairs of intermediate locations (i.e., Greenlee to Guthrie Peak and Guthrie Peak to Dos Condados). As shown by the channel monitoring log in Fig. 14 in Section VIII, such a network configuration suffers from intermittent communications dropouts on a regular basis due to the nature of microwave channels. Without a communications report, SWTC can neither confirm that this channel is troublesome nor understand the risk of deployed protection schemes.

VIII. Communications Channel Monitoring

A study of protection system misoperations in North America that occurred over a one-year period between 2011 and 2012 found that the vast majority of misoperations (94 percent) were false trips. Only about 6 percent of the reported misoperations were the result of slow tripping or a failure to trip. Of those misoperations attributed to communications-related problems, the data indicated that approximately 25 percent of the false trips occurred during nonfaulted conditions. Therefore, it is beneficial to identify the weak links in the system from a communications report so that the protection scheme can be designed to suit the characteristics of the channel. Fortunately, the majority of modern relays and communications devices keep operation logs for analysis and corrective action.

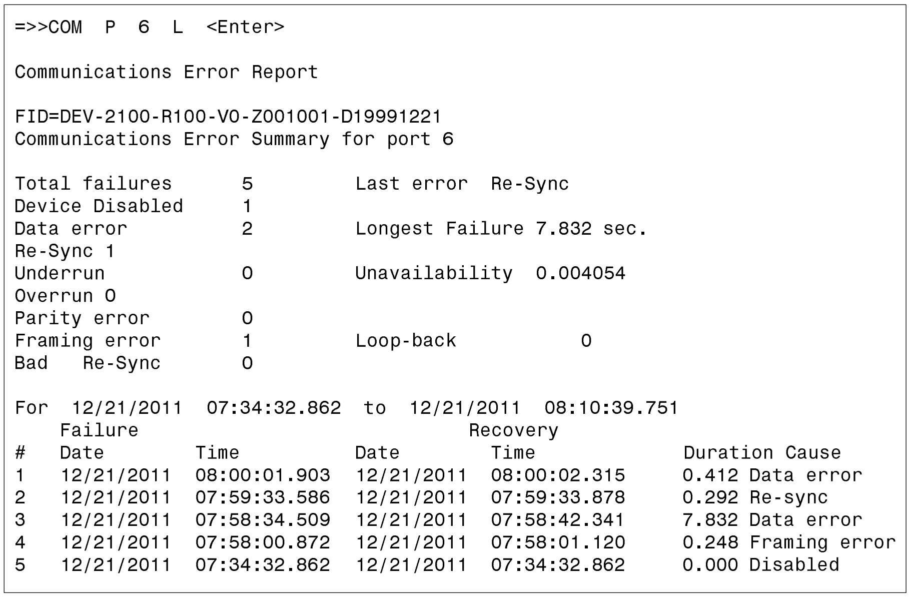
Table I describes typical errors reported on a bidirectional asynchronous channel. Errors detected by the relay should be compiled for analysis to determine the source of the errors. A typical error record should include the following fields:

- Time the problem started.
- Time the problem stopped.
- Duration of the problem.
- Reason for dropout.

Table I
Error Types From a Communications Report

Error Type	Description
Underrun	Multiple messages transmitted without one of them being received
Overrun	Receive buffer overflowed
Resynchronization	Remote communications device detected an error and sent a resynchronization message
Data error	Received data were not self-consistent, or the address was wrong
Relay disabled	Relay protection functions disabled
Loopback	Channel loopback detected

A communications report is generated for each communications port. It contains a summary report followed by a detailed listing of each communications problem detected on the port.



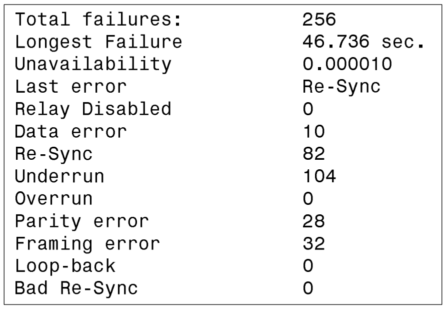
```
=>>COM  P  6  L  <Enter>

Communications Error Report

FID=DEV-2100-R100-V0-Z001001-D19991221
Communications Error Summary for port 6

Total failures      5         Last error  Re-Sync
Device Disabled     1
Data error          2         Longest Failure 7.832 sec.
Re-Sync 1
Underrun            0         Unavailability  0.004054
Overrun 0
Parity error        0
Framing error       1         Loop-back           0
Bad   Re-Sync       0

For  12/21/2011  07:34:32.862  to  12/21/2011  08:10:39.751
    Failure                              Recovery
#   Date        Time          Date        Time              Duration Cause
1   12/21/2011  08:00:01.903  12/21/2011  08:00:02.315      0.412 Data error
2   12/21/2011  07:59:33.586  12/21/2011  07:59:33.878      0.292 Re-sync
3   12/21/2011  07:58:34.509  12/21/2011  07:58:42.341      7.832 Data error
4   12/21/2011  07:58:00.872  12/21/2011  07:58:01.120      0.248 Framing error
5   12/21/2011  07:34:32.862  12/21/2011  07:34:32.862      0.000 Disabled
```

Fig. 13. Sample communications report.

Fig. 13 shows an example of a comprehensive communications report from Port 6 of a device. The device is communicating via an asynchronous peer-to-peer protocol. The report was generated on December 21, 2011, at 08:10 and shows that there have been five communications disturbances since the last time the report was cleared.

The example in Fig. 13 illustrates the wide variety of errors captured by the report. Actual field data from devices communicating in the peer-to-peer protocol described previously are provided in the following paragraphs.

Fig. 14 shows failure statistics from SWTC between June 17, 2012, 13:35:02.701 and January 18, 2013, 13:50:08.672, when the Morenci-to-Hackberry communications link incurred 256 message failures.



```
Total failures:            256
Longest Failure            46.736 sec.
Unavailability             0.000010
Last error                 Re-Sync
Relay Disabled             0
Data error                 10
Re-Sync                    82
Underrun                   104
Overrun                    0
Parity error               28
Framing error              32
Loop-back                  0
Bad Re-Sync                0
```

Fig. 14. Morenci-to-Hackberry link failure statistics collected from June 17, 2012, to January 18, 2013.

Fig. 15 plots the duration of the individual message failures by date, which provides an indication of the nature of message failures on a communications link consisting predominantly of microwave hops.

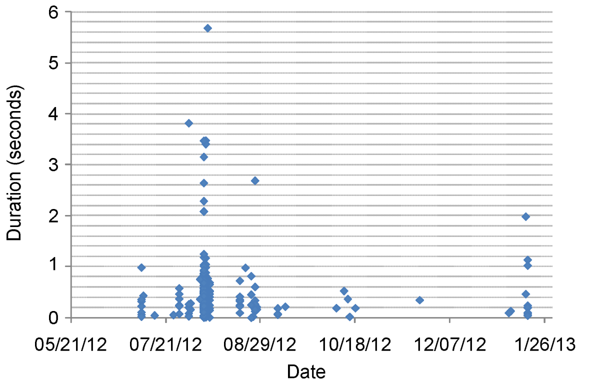


Fig. 15. Morenci-to-Hackberry message error duration in seconds from May 21, 2012, to January 26, 2013.

Microwave radio is susceptible to signal fades, which can be caused by rain, fog, or reflection-induced multipathing. Increasing channel bandwidth to fit in all the applications in the presence of noise can also contribute to intermittent communications message failures. The communications message failures on the Morenci-to-Hackberry microwave link are sporadic in nature, with numerous message errors concentrated in short periods of time. The short periods with multiple message errors are most likely the result of weather-induced communications problems.

Communications over fiber-optic links are virtually immune to weather-induced communications issues. Fiber communications paths tend to be nearly error free until they fail. An examination of message failures on a different fiber link between the SWTC Bicknell and Sahuarita substations shows that there were 26 message failures between March 4, 2010, and January 13, 2013. Seven of the 26 failures occurred on October 26, 2012, and the other 19 failures occurred on May 5, 2012, presumably when some work was being done on the fiber links. There were zero message failures for the other roughly 1,000 days during this period. This rate exceeds the commonly specified BER of 10^{-9} for fiber-optic channels.

With the channel characteristic information available to us, we can select the most suitable protection schemes. For example, an OPGW cable runs on top of the Bicknell-to-Sahuarita line. This field data-proven robust channel gives us the confidence to apply a line differential scheme where the protection completely depends on the communication.

For the line between the Morenci and Hackberry substations, because of the high possibility of packet corruption, the status change of a digital bit may not be correctly received and accepted when a power system fault coincides with channel events. The presently deployed POTT scheme might fail to operate in the event of a line fault. The highly secure POTT scheme might not be the most appropriate choice for a channel that is not dependable. We can consider converting the POTT scheme to a DCB scheme that is more dependable because it does not require a good channel for tripping, only for blocking. This choice, however, increases the risk of overtripping for a fault on an adjacent line if the blocking signal is not received correctly or within the allotted coordination delay time. This is the traditional tradeoff—dependability versus security. Increasing one generally results in decreasing the other. However, before making that decision, we should examine some innovative features available in modern communications channel protocols that can offer improvements in both dependability and security instead of simply increasing one at the expense of the other.

IX. Innovative Uses of Real-Time Channel Monitoring Function

Besides the communications channel report, the increasing availability of real-time monitoring provides the opportunity to dynamically adjust the speed and security balance of channel-dependent protection elements when the channel quality changes. Dynamic adjustment automatically shifts its bias towards security when the system becomes degraded, without penalizing the protection speed under normal operating conditions. The most straightforward security enhancement method, as described in Section V, is simply increasing the security counter when the channel quality is poor, which can be implemented as shown in Fig. 16.

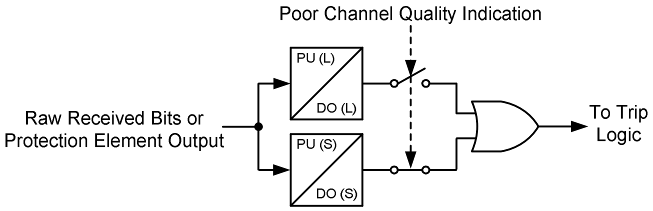


Fig. 16. Channel quality-based dynamic delay control.

Any raw bit received directly from the communications channel or the output of a protection element that uses the remote analog data as its input is debounced by a pair of counters connected in parallel. The pickup (PU) and dropout (DO) delays of the top counter are selected with added security in mind while the corresponding settings for the bottom counter can be given more consideration for speed. A digital bit derived from the channel monitoring function toggles a switch that decides which counter output is routed to the OR logic gate. For example, if the short-term unavailability index goes above 2,000 packets per minute (ppm) or there are more than 20 packets lost among the last 10,000, this supervisory bit would assert. Therefore, as long as the channel has good quality in the context of an intended application, the bottom counter remains effective. Once the real-time monitoring alarms for abnormal channel activities, an extra delay is applied. This may be acceptable as a short-term solution. Attending to the alarm and rectifying the communications problem are the keys to restoring the system back to its normal state.

A discussion of the recommended security counter increase needs to cover two aspects: the raw bits received from the channel, such as DTT signals, and the output from a protection element that might be exposed to channel noise (e.g., an 87L element). The recommendation of additional security counts for directly received digital bits is well established in Section V.

In the case of protection elements that use analog data from the channel, one extra packet delay might not be adequate. In any digital encoding scheme, the corruption of certain bits of the encoded analog data can lead to a huge increase in operating quantities. If an incorrect analog value somehow leaks into the relay digital filter, the impact is not only instantaneous but the output of the filter remains invalid as long as the bad value is still in the filter memory. The existence of such a possibility also disqualifies the more intuitive approach of adjusting the pickup sensitivity. The characteristics (time constant, buffer size, and so on) of the digital filter that supplies analog signals to the downstream protection element need to be taken into account when determining the required extra debounce time. The delay needs to be extended at least until bad data leave the filter memory.

The proposed control action can also be implemented by means of either switching settings groups or using a torque-control method.

Another powerful approach is to combine the channel monitoring with disturbance detector supervision to get even more security with the logic shown in Fig. 17.

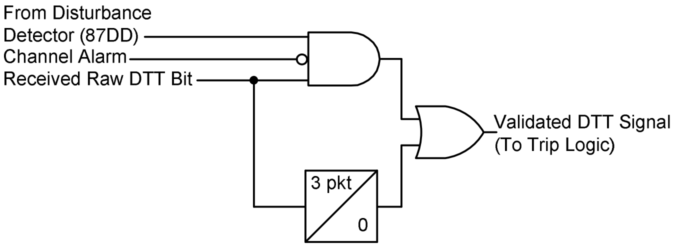


Fig. 17. Instantaneous trip path disabled by channel alarm.

Instantaneous transfer tripping is only allowed when a power system disturbance has been detected while there is not a channel alarm condition.

Considering the innovative ways we can compensate for adverse channel characteristics, we revisit the scheme selection for the Morenci-to-Hackberry line protection. The POTT scheme is secure because it requires a permissive signal and a forward overreaching protection element to trip without additional time delay. Reviewing the communications performance report from Fig. 14, of the 256 recorded communications problems, 70 of them involved some type of signal corruption (data error, framing error, or parity error), and 104 of these were underruns, where the relay did not receive a signal in the expected time frame. The remaining 82 records were for resynchronization signals, which actually indicate that errors were detected in the other direction on the same channel, so the other terminal initiated a signal resynchronization process. This brings up an interesting point: bidirectional communications channels can have a different characteristic in each direction, primarily dependent on path and channel traffic. In this case, either a corrupt signal or the lack of a timely signal results in a slow trip, possibly as slow as the Zone 2 time-delayed step-distance trip time if a good signal is not received.

The traditional alternative scheme, DCB, is dependable because it does not require the communications channel to be in service to trip. The communications channel is only required to block the remote terminal from tripping if a fault is detected in the reverse direction, which means the fault is not on the protected line segment. The security of this scheme is lower than that for the POTT scheme because of the risk of overtripping on an adjacent line fault if the communications channel is in a failed state. The DCB scheme also requires a coordination delay to account for the normal channel delay incurred between the time the blocking signal is sent from one terminal and received at the other. Variable channel delay due to path switching can be extremely detrimental to DCB scheme tripping speed and security. The coordination delay must be set greater than the maximum expected channel latency. If the channel latency exceeds the coordination delay setting, there is a risk of tripping prematurely for faults on adjacent lines. The ramifications of this overtripping can be mitigated by employing high-speed reclosing for pilot scheme trips (even an erroneous one in this case). Protection and planning personnel need to determine if the risk of overtripping with a follow-on high-speed reclose is more or less detrimental to the system integrity than a slow trip.

If the channel monitoring system provides real-time channel status, then we can improve the POTT scheme dependability by using an OR operator to combine the permissive signal with the inverse of the channel state (one if disabled and zero if enabled). With this logic, the scheme is much more dependable for all faults on the line. However, if the fault is on an adjacent line within the reach of the overreaching protection elements, there is a risk of overtripping provided that the channel state is disabled. This is the same problem we would encounter by switching to DCB scheme logic.

These evaluations suggest a hybrid communications-assisted protection scheme that would offer high-speed clearing of more faults while being as dependable as the DCB scheme and as secure as the POTT scheme: the best of the existing communications schemes. This scheme was first proposed in [8]. Fig. 18 shows the logic for the hybrid protection scheme. While the channel is in service, the scheme can trip three different ways:

- A fault is detected in Zone 2 for carrier coordination (CC) time if no block trip signal is received (traditional DCB).
- A permissive trip signal is received, and the fault is not behind the terminal (DCB from perspective of remote end).
- A direct trip signal is received (saves 0.5 cycles for lower-resistance faults).

If the channel fails (NOT ROK), the DCB logic (AND Gate 1) is blocked to maintain security. A 10-cycle tripping window is opened (AND Gate 2) that allows the relay to trip if a forward fault is detected. This improves dependability in instances where the channel fails as a result of the fault. A single reverse-blocking (RB) timer disables the permissive tripping logic and extends the block trip signal to provide security during current reversals.

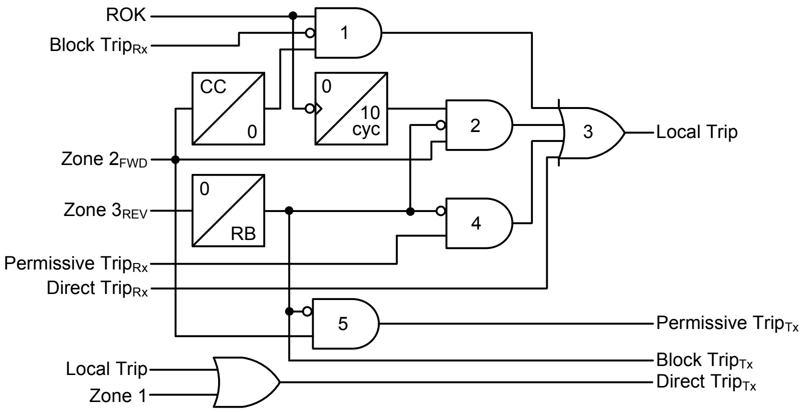


Fig. 18. Hybrid communications-assisted pilot scheme logic.

The dependability and security of this protection scheme are both high, with performance identical to a DCUB scheme. All faults detected at either end are cleared at both ends in 5 cycles or less.

A Zone 1 element is used to send the direct trip signal—thus three zones are used by the hybrid scheme, which is easily accomplished with a microprocessor-based distance relay. The direct trip Rx (receive) signal can be routed directly to the local trip output, as shown in Fig. 18, or for added security, it can first be passed through a debounce counter to require more than one message with the direct trip bit asserted before it is routed to the local trip output. It can also be supervised with a disturbance detector, such as an undervoltage element and/or an overcurrent element, for even more security against a false trip.

Three timers (carrier coordination, a loss-of-channel tripping window, and a reverse-blocking timer) are used in this hybrid scheme. The hybrid scheme also requires that three bits be communicated end to end: a block trip, permissive trip, and direct trip. One microprocessor-based relay includes communications capabilities that make this scheme very feasible and practical.

X. Fallback Considerations

For the sake of maximizing the Hamming distance of a certain error detecting code, its potential error correcting capability is almost always abandoned. Therefore, whenever the received packet is deemed invalid, its content is discarded rather than fixed. In other words, the security counter, which has been recognized in the previous sections as a powerful tool to improve security against undetected errors, would have no input data from a corrupted packet. How it behaves in the event of data corruption needs to be carefully calculated.

Regardless of the protocol, three settings are commonly assigned to each digital status bit to be received: pickup time, dropout time, and default fail-safe state. Because a security counter is part of the design, it is possible for packet corruption to occur when the counter is counting up or down towards its state-change threshold. Completely restarting the counter for one or two invalid packets is not ideal and can inadvertently affect the speed of operation by a great deal. Therefore, a more adequate solution, as shown in Fig. 19, is to put the counting process on hold and freeze the count value when no valid packet is received. Without losing the security gained by requiring confirmation of successive packets, the impact of invalid packets is limited to the duration of the noise burst instead of the entire pickup or dropout time.

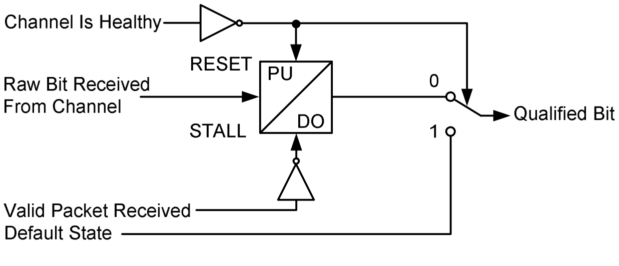


Fig. 19. Fail-safe logic for user-programmable bits.

The security counter, including its output and memorized counts, is cleared only when the relay is unable to receive any healthy packets for an extended period of time. At the same time, the logic passes the user-defined fail-safe value of that bit to the downstream function that uses it as an input. The advantage of this approach becomes more evident when the channel is relatively noisy. The qualified bit will eventually be driven to and stay on its intended state even if channel noise causes packet dropping to happen on a regular basis.

In addition to user-programmable bits, a high-performance 87L relay typically defaults to the DTT bit and external fault detection (EFD) bit, among a few others, in its packet payload [9].

The processing of the DTT bit is unique in that the relay needs to be absolutely sure that the received DTT command is legitimate. A validated DTT bit directly results in tripping one or multiple circuit breakers. Restricted by the message rate, debouncing with multiple consecutive packets is a good solution as long as there is enough margin in the critical fault clearing time. If extra delay is not acceptable, the disturbance detector supervision, as described in Section V, is a better candidate because DTT is normally the result of a system disturbance. However, there is still a chance of undesired operation if the receiving logic latches the DTT bit to the wrong state.

For example, a packet with the DTT bit inverted by channel noise might slip through the data integrity check and be treated as valid. The received raw DTT bit thus transitions to one because a new valid packet declares it so. A very unfortunate scenario is that all subsequent packets are lost or unidentifiable due to a noise burst, which will latch the DTT bit to one indefinitely if the receiving logic chooses to latch this bit to its last known state. Such behavior will eventually defeat the security created by the three-packet qualifying path in the disturbance detector supervisory logic. For this reason, the raw DTT bit should be reset to zero right away when no packet or an invalid packet is received.

The need for exchanging the EFD bit among 87L relay terminals is explained in [9]. The 87L element is usually driven into a more secure operating mode when the EFD bit is picked up [10]. In other words, temporary assertion of the EFD bit as a result of channel noise can delay the fault clearing. However, the relay will be able to operate as soon as its correct value is received and validated. Therefore, it is acceptable to latch this bit to its last known good value if the present packet is invalid or did not arrive. By not resetting the EFD bit immediately to zero, the security of the 87L system is preserved. To prevent unnecessary overrestraining, a good practice is to clear the already asserted EFD bit when the channel is completely out of service. This treatment of keeping the present value until a new valid packet says otherwise is also commonly adopted in handling the GOOSE reception where the heartbeat interval is set in a remote device [11].

In summary, the emphasis is definitely on security when managing the DTT bit while the fallback strategy of the EFD bit is more biased towards dependability.

XI. Conclusion

Status bit exchange within or between substations through various digital communications protocols has become an inseparable part of modern protection design. The integration of digital communication into the existing system brings many challenges to both protection and communications personnel. Several mature protocol options are available, and one of the many differences among them is the exclusiveness of channel usage. A peer-to-peer protocol takes up all the resources of a given channel, digital bits as part of an 87L frame share the channel with analog data, and GOOSE messaging implemented over Ethernet has to share the bandwidth with many other types of data.

Under the circumstances that additional delay is not acceptable and the number of spare bits that can be combined to achieve the desired security is also limited, disturbance detector supervision, which works on the local analog data that are not affected by channel activities, provides an excellent security boost.

Regardless of the choice of protocol, transmit and receive address verification helps to prevent accidental cross-connection from endangering the protection system.

As pointed out by the field examples and the protection system operation survey, undesired DTT operation triggered by channel noise can occur under both faulted and nonfaulted power system conditions.

Real-world experience reported by SWTC and other practitioners in the field demonstrates the exceptional value of the communications monitoring report provided by modern digital relays. The monitoring report makes it possible to detect channel problems early on, allowing them to be rectified before they cause trouble. The information from the communications report is also valuable in selecting the protection scheme to match the characteristics of the channel. Furthermore, the real-time measurement can also be used to adjust the security counter to optimize the operating speed when the channel is healthy and maintain the same level of security even if the channel becomes noisy.

When the packets are not correctly received due to temporary unavailability of the channel, both the user-programmable bits and predefined 87L system bits are handled in a way that best suits their respective natures of application. Certain bits can be latched to their previously known good values to ride through packet corruptions while other bits must be cleared without intentional delay.

XII. References

[1] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Pearson-Prentice Hall, Upper Saddle River, NJ, 2004.

[2] C. T. Ong and C. Leung, "On the Undetected Error Probability of Triple-Error-Correcting BCH Codes," *IEEE Transactions on Information Theory*, Vol. 37, Issue 3, May 1991, pp. 673–678.

[3] W. Chong, "On the Undetected Error Probability of BCH Codes," MASc thesis, Electrical and Computer Engineering, The University of British Columbia, 1992.

[4] J. Stone and C. Partridge, "When the CRC and TCP Checksum Disagree," proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Stockholm, Sweden, August 2000.

[5] P. Koopman, "32-Bit Cyclic Redundancy Codes for Internet Applications," proceedings of the International Conference on Dependable Systems and Networks, Bethesda, MD, June 2002.

[6] H. S. Warren, Jr., *Hacker's Delight*. Addison-Wesley Professional, Upper Saddle River, NJ, 2012.

[7] E. O. Schweitzer, III, K. Behrendt, and T. Lee, "Digital Communications for Power System Protection: Security, Availability, and Speed," proceedings of the 25th Annual Western Protective Relay Conference, Spokane, WA, October 1998.

[8] E. O. Schweitzer, III, and J. J. Kumm, "Statistical Comparison and Evaluation of Pilot Protection Schemes," proceedings of the 23rd Annual Western Protective Relay Conference, Spokane, WA, October 1996.

[9] H. Miller, J. Burger, N. Fischer, and B. Kasztenny, "Modern Line Current Differential Protection Solutions," proceedings of the 63rd Annual Conference for Protective Relay Engineers, College Station, TX, March 2010.

[10] B. Kasztenny, G. Benmouyal, H. J. Altuve, and N. Fischer, "Tutorial on Operating Characteristics of Microprocessor-Based Multiterminal Line Current Differential Relays," proceedings of the 38th Annual Western Protective Relay Conference, Spokane, WA, October 2011.

[11] IEC 61850-8-1, Communication Networks and Systems for Power Utility Automation – Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3, ed. 2.0, June 2011.

XIII. Biographies

Zachary Eyasu received his B.S. in electrical engineering from San Diego State University, California. He joined Southwest Transmission Cooperative, Inc. in 2008 as a protection and automation field engineer. His activities include performing power system protection studies and design, testing and commissioning new and upgraded substations, modernizing protection scheme and relay programming, and supervising electrical system fieldwork.

Bin Le received his B.S.E.E. from Shanghai Jiao Tong University in 2005 and an M.S.E.E. degree from the University of Texas at Austin in 2008. He has been employed by Schweitzer Engineering Laboratories, Inc. since 2008. Mr. Le currently holds the position of power engineer in the research and development division. He is a member of IEEE and a professional engineer registered in the state of Washington.

Tony Lee received his B.S. in electrical engineering from Washington State University in 1987. Mr. Lee then worked for Texas Instruments in Dallas, Texas, from 1987 through 1991, when he joined Schweitzer Engineering Laboratories, Inc. (SEL) as a hardware design engineer. He presently holds the position of research and development director at SEL. Mr. Lee holds twenty-one U.S. patents and several foreign patents and has several patents in process.

Ken Behrendt received his B.S.E.E. from Michigan Technological University in 1970. Upon graduating, he served nearly 24 years at Wisconsin Electric Power Company (now WE-Energies), where he worked in distribution planning, substation standards development, distribution protection, and transmission planning and protection. He joined Schweitzer Engineering Laboratories, Inc. in 1994, where he is a senior application engineer located in New Berlin, Wisconsin. He is a senior member of IEEE and a member of the Power System Relaying Main Committee and has authored and presented several papers on power system protection topics.

Veselin Skendzic is a principal research engineer at Schweitzer Engineering Laboratories, Inc. He earned his B.S. in electrical engineering from FESB, University of Split, Croatia; his M.S. from ETF, Zagreb, Croatia; and his Ph.D. from Texas A&M University. He has more than 25 years of experience in electronic circuit design and power system protection-related problems. He is a senior member of IEEE, has written multiple technical papers, and is actively contributing to IEEE and IEC standard development. He is a member of the IEEE Power Engineering Society (PES) and the IEEE Power System Relaying Committee (PSRC).

20130328 • TP6602-01