# Speed and Reliability of Ethernet Networks for Teleprotection and Control

Gary W. Scheer and Darold A. Woodward
*Schweitzer Engineering Laboratories, Inc.*

Presented at the
3rd Annual Western Power Delivery Automation Conference
Spokane, Washington
April 10–12, 2001

# SPEED AND RELIABILITY OF ETHERNET NETWORKS FOR TELEPROTECTION AND CONTROL

Gary W. Scheer, PE and Darold A. Woodward, PE
Schweitzer Engineering Laboratories, Inc.
Pullman, WA USA

## INTRODUCTION

A growing number of electric utilities are applying or evaluating Ethernet networks for substation instrumentation and control (I&C). Interest in Ethernet is driven in part by the work of the Electric Power Research Institute in North America, as they work with utilities, equipment vendors, and standards organizations to define the Utility Communications Architecture (UCA). The primary reason for UCA is to achieve faster and lower cost substation integration, by reducing the labor required to integrate monitoring and control devices for substation and feeder equipment. This expected labor reduction results from standard object models and self-description of the data items and their characteristics through the network. If devices from multiple vendors comply with the rules (interoperability), special integration labor for each device is reduced or eliminated. To realize these goals, software tools will need to exploit this self-description to automate portions of the system integration process.

Protection engineers benefit from understanding the appropriate application of Ethernet networks for protection applications. Automation engineers benefit from understanding application of GOOSE messages for real-time station control.

In this paper we apply familiar tools to evaluate Ethernet protection and control applications. Important criteria for evaluating protection and control applications include transmission time, dependability, security, and availability. We analyze transmission time, dependability, and security for UCA event messages using portions of the IEC-834 Standard, *Performance and Testing of Teleprotection Equipment Of Power Systems* [1]. System availability based on equipment reliability is addressed in the paper, "Comparing The Reliability Of Ethernet Network Topologies In Substation Control And Monitoring Networks"[2].

## UTILITY COMMUNICATIONS ARCHITECTURE AND GOOSE MESSAGES

### UCA2

The Utility Communications Architecture consists of definitions of generic object models and the instructions to build extensions to models and new models. UCA2 is defined in the document *UCA2 Generic Object Models for Substation and Feeder Equipment (GOMSFE) Version 0.91* [3]. The goals of interoperability and improved integration efficiency depend on successful incorporation of GOMSFE into substation networks. The architects of UCA2 provided an additional mechanism, defined as a part of the GLOBE model, which is named the Generic Object-Oriented Substation Event (GOOSE). While not required to achieve the primary goals of UCA, the GOOSE mechanism provides high-speed communications that may interest protection and automation engineers.

## GOOSE Transmission and Reception

UCA2 devices can use GOOSE messages for event-driven, peer-to-peer communication. Each UCA2 device sends a GOOSE message when a monitored point changes state, for example from logical zero to logical one. After sending the initial event-driven message, the sending device waits for a calculated delay time, and then transmits the same message again. The calculation of this delay time is defined by the UCA2 Standard to increase with each re-transmission [3]. The delay between the first event-driven message and the first retransmission is five milliseconds. The message is sent with decreasing repetition frequency, until the maximum delay is capped at one minute. The delay time, in milliseconds, is specified by Equation 1:

$$t = 4 + (1.n)^{R-1} \tag{1}$$

Where n is a setting between two and nine, inclusive, and R is a sequential repeat-number of the message.

For fastest transport by the network, GOOSE messages are broadcast (multicast) messages. A publisher-subscriber model applies to GOOSE message usage. The originator of the message publishes (sends) the message, and each device that uses data from the message subscribes to the data. This subscription is accomplished by settings in the receiving device that indirectly or directly specify the multicast address used by each sender of interest. The receiving device ignores GOOSE messages with addresses that are not in the local list of addresses.

Another attribute of a GOOSE message is that it contains an expiration time, designated the Hold Time. When a message is aged beyond its Hold Time, it is ignored.

The combination of the multicast address and the expiration time reflect the designers' intention that the realm of GOOSE messages is limited to subscribing devices that share the same network segment as the publishing devices. Because GOOSE messages are not routable, their intended application is in local-area networks or segments rather than in inter-station wide area networks (WANs). An engineer could consider using GOOSE messages through a WAN. This would require using a non-standard application, with a non-multicast destination address and a longer expiration (hold) time.

An example application discussed in the definition of the GOOSE mechanism is to isolate an electrical bus by sending a trip command to the protective relays for lines connected to the bus. By broadcasting the messages, all of the devices receive and act upon the same message. In contrast, a non-broadcast network mechanism would send a message to each receiver.

## GOOSE MESSAGE CONTENT

The monitored points in GOOSE messages are each represented by a two-bit value, or "bit-pair." Each bit-pair conveys the status of a single binary point, as shown in Table 1.

**Table 1** GOOSE Message Bit-Pair Usage

| Binary Value | Meaning |
|---|---|
| 00 | In transition or unknown |
| 01 | Clear (0) |
| 10 | Set (1) |
| 11 | Invalid |

The primary content of the GOOSE Message is 32 bit-pairs named DNA and 128 user-defined bit-pairs, for a total of 160 bit-pairs. Fields in the GOOSE message also provide identification, the message sequence number and hold-time. For a GOOSE message with 128 user-defined bit-pairs transported over a network, the total packet size is approximately 320 bytes long. The GOOSE receiving device determines the actions to perform on receipt of new GOOSE data.

## IEC-834 STANDARD DEFINITIONS

The IEC-834 Standard is commonly used to evaluate point-to-point teleprotection. The following definitions are paraphrased from the IEC-834 Standard [1], including attachments:

**Transmission time of a teleprotection channel**: the time elapsed between the moment of change of state at the transmitter input and the moment of the corresponding change of state at the receiver output. Propagation time is excluded.

- **Nominal transmission time ($T_0$):** transmission time under noise-free conditions.
- **Actual transmission time ($T_{ac}$):** maximum transmission time measured under noisy conditions for a defined dependability and signal-to-noise ratio or bit-error rate.

**Security:** the ability to prevent interference and noise from generating a command state at the receiving end when no command signal is transmitted. $P_{uc}$ is the probability of an unwanted command.

**Dependability**: the ability to issue a valid command in the presence of interference and/or noise. $P_{mc}$ is the probability of missing a command. $1–P_{mc}$ is normally applied as the measure of dependability. A command is considered missing if for a valid transmitted command, no valid command is received before an excessive delay time.

## IEC-834 STANDARD REQUIREMENTS APPLIED TO UCA2 GOOSE MESSAGES

For meaningful comparison to point-to-point connections, consider two protection or automation devices that communicate with each other using GOOSE messages over a substation network. In this case, these two devices are the only users of GOOSE messages; the balance of the network traffic is to support a distributed instrumentation and control system, using GOMSFE models and

other data packets. One requirement of the IEC-834 Standard is that the probability that a command is not successfully transferred within 20 milliseconds must be less than $10^{-3}$.

## GOOSE Message Nominal Transmission Time

The nominal transmission time, $T_0$ is the time it takes to transmit the initial GOOSE message, and to have the subscribing device able to act on the received data. In a theoretical unloaded network, where the probability of a collision is 0%, the nominal transmission time is comprised of the processing time in each device, the transmission time of the message, plus any delays introduced by network components. For two devices connected to the same hub, the nominal delay of the GOOSE message is shown in milliseconds in Table 2 for network speeds of 10 Mbps and 100 Mbps.

**Table 2** GOOSE Message Nominal Transmission Time

| Description | Time at 10 Mbps (msec) | Time at 100 Mbps (msec) |
|---|---|---|
| IED Preprocessing | 5.4 | 5.4 |
| Transmission | 0.242 | 0.024 |
| IED Post Processing | 4.426 | 4.426 |
| Total | 10.068 | 9.85 |

## Ethernet Network Loading, Delays and Dependability

Ethernet networks are subject to delays due to network traffic when a message is ready to send, and when more than one node attempts to start transmitting at the same time (collision). Include these delays in calculations of $T_0$.

The probability of a delay due to traffic ($P_{BUSY}$) is a function of the loading of the network ($P_{LOAD}$). For an n-node network, with loading between 0 and 1, Equation 2 gives the probability of the network being busy due to other nodes when a given node is ready with a new message.

$$P_{BUSY} = \frac{n-1}{n} \bullet (P_{LOAD}) \tag{2}$$

However, if $P_{LOAD}$ represents the network loading excluding the incremental loading due to a rare protection event, it is more accurate to assume that $P_{BUSY}$ equals $P_{LOAD}$. If the network is busy, the average delay is one-half of the average message time, plus the interframe gap.

It is at this time that collisions are likely. A considerable body of work exists [5] that addresses the statistical characterization of networks using the collision detection method employed in Ethernet networks, CSMA/CD [5]. This paper does not attempt to reproduce this work to generate a complete statistical prediction model of collision resolution. Instead, we focus on the collision likelihood in the particular case of two devices using GOOSE messages to exchange data to protect the same element of electrical apparatus. Assume that the protection scheme requires both devices to generate a message resulting from the same event. Generally, one device senses the event and has a message ready to send before the second device. If the network is not busy, the first device successfully sends the message. The second device detects that the network is busy due to the successful message from the other device and waits until the network is available. If the network is busy when both devices are ready to transmit, both wait for the

network to clear, both wait the interframe gap time, and then collide as they both attempt to transmit.

The Ethernet collision detection and retry process attempts to retransmit for up to 16 collisions.

Equation 3 is the CSMA/CD equation for the back-off slot time multiplier, for collision-retry number n:

$$0 \le r \le 2^k \tag{3}$$

where r is an integer random number, and k is the minimum of the retry attempts (n), or 10.

Multiply r of Equation 3 by the slot time of 512 bit-times. For a 10 Mbps network, the slot time is 51.2 microseconds. Multiply r by the slot-time to yield the back-off (delay) time in microseconds, for a given retry attempt (n).

$$T_0 \equiv r \bullet 51.2 \tag{4}$$

The most time-critical specification for the IEC–834 Standard is for blocking schemes. The probability of missing a command ($P_{mc}$) must be less than $10^{-3}$ within a time of 20 milliseconds.

We applied the following assumptions to simplify the predictions for comparison against the standard:

- The non-GOOSE network loading is comprised of messages of an average time length, evenly distributed in time. The local segment traffic is visible to all nodes connected to shared hubs.
- The idle time between the messages of length $T_{AVG}$ is:

$$T_{IDLE} \equiv \left( \frac{1}{P_{LOAD}} - 1 \right) \bullet T_{AVG} \tag{5}$$

- Two devices exchange GOOSE messages to coordinate protection of connected apparatus. Both devices sense the event and attempt to send a message. If this event occurs when the network is idle, it is very likely that that one device will be ready to send a message more than 9.6 microseconds earlier than the other; an initial collision is unlikely and can be neglected.
- If the network is busy when a device is ready to send a GOOSE message, the message is delayed at most by one message time ($T_{AVG}$) and an interframe gap. Assume that the two devices determine the need to send a GOOSE message within this time. So, at the end of the interframe gap, assume that a collision is inevitable.
- The hold-time is set longer than 15 milliseconds, to eliminate its impact on the calculations.
- Resolution of each collision step results in one of the following:
  - After a calculated random delay, the device under analysis wins the collision resolution and is allowed to transmit.
  - After a calculated random delay, the other device wins the collision resolution and is allowed to transmit. The device under analysis senses that the other device has the network busy, waits for the message to complete plus an interframe gap time, and then sends the message.

  - After a calculated random delay a collision occurs, and the next collision resolution step takes place.

- The percentage-resolved in each step is the sum of the probabilities of the cases where the devices have non-equal delay times, multiplied by the percentage of cases remaining to resolve at the beginning of the step.
- The worst-case time to successfully send a message in a step is when the other device is allowed to send a message after one slot-time less than the maximum delay. (See equations 3 and 4). Then, the device under test is able to send a message.
- At the end of each step, add the worst-case time of resolved cases to the delay time, and the additional percentage of resolved cases to the total-cases resolved. When this percentage solved exceeds 99.9%, there is less than a $10^{-3}$ probability that the message will be lost due to collisions, within the accumulated delay time.
- CSMA/CD inherently favors newer message requests than older ones, a phenomenon, documented as "packet starvation effect" [6]. At each step where the accumulated delay exceeds a step in the idle time, add a delay of an average message time plus inter frame gap to account for delays from the other loading.

Apply equations 2 through 4 and the simplifying assumptions to loadings from 10% to 60%. A summary of the results is shown in Table 3. These data indicate that in a noise-free network, base-processing time plus collision delays is within the IEC specification for blocking schemes at approximately 60% loading.

**Table 3** Predicted Collision Delay Times in Noise-Free Network for Selected Loading Levels

| Loading (%) | Time With $P_{mc} < 10^{-3}$ (msec) |
|---|---|
| 10 | 12.33 |
| 20 | 12.33 |
| 30 | 12.97 |
| 40 | 13.61 |
| 50 | 14.25 |
| 60 | 15.53 |

At 70% loading, the available time between messages is comparable to the GOOSE message time, so the timing predicted by this mechanism becomes indeterminate. Independent tests conducted by Hydro One for a similar example case [7] show consistent GOOSE performance of less than 20 milliseconds for loadings below 70%, with rapid degradation of time at higher loadings.

## Noise and GOOSE Messages on Ethernet Networks

Include the effect of noise in $T_{ac,}$ the actual transmission time. Noise in digital networks is typically expressed as a bit-error rate. Noise has different impacts depending upon the portion of the data-stream that it affects.

- Error bursts during idle times that are less than the minimum packet length do not introduce false data nor corrupt valid data, but appear as small bursts of loading. If the burst takes

place when a node is sensing network status to initiate a message, it could introduce a delay of the balance of the noise burst plus one interframe gap.

- Errors that occur in non-GOOSE messages do not directly delay nor corrupt the GOOSE message sequence under test. They may cause slightly increased network loading, depending on the nature of the network.
- Errors that occur in a GOOSE packet and are detected by the CRC cause the entire message to be ignored. This instance of the GOOSE packet is lost, and the dependability depends on subsequent GOOSE retransmissions.
- Errors that are not detected by the CRC depend on where in the packet they occur:
  - Errors in the addressing or control data, or in the identification, time, or hold-time fields cause the packet to be dropped or ignored.
  - Errors in the data bit-pairs that change a value from 01 to 10, or from 10 to 01 are unwanted commands, to be included in determining probability $P_{uc}$.
  - Errors in the data bit-pairs that change a value from 01 or 10 to either 00 or 11 do not cause false operation, but cause the default value to be used. Undetected errors in the first GOOSE message cause a delay. In subsequent messages, these changes do not impact the dependability or the security.
- Errors in the CRC bits generally cause rejection of the packet, as an error detected by the CRC comparison.

Almost all errors cause rejection of the packet. When a packet is rejected, the dependability relies on receiving the subsequently re-transmitted messages. From Table 1 the first message starts transmitting 5.4 milliseconds after the event is detected. From Equation 1, five milliseconds later the second message starts transmitting (elapsed time = 10.4 milliseconds). The longest time until the next message starts is when n is set to 9 in Equation 1, yielding an additional 5.9 milliseconds for an elapsed time of 16.3 milliseconds. Add this to the 4.668 milliseconds remaining from Table 1, for a time of 20.968 milliseconds. So, if the first message is not successfully received, then there is one more message that may be decoded successfully within the 20-millisecond time window. Each of these messages are subject to additional delay due to collisions. The elapsed times for subsequent transmissions of the initial GOOSE message are summarized in Table 4.

**Table 4** Elapsed Time for Repeated GOOSE Messages

| Message Number | Elapsed Time to Start Message | Elapsed Time to Complete Processing |
|---|---|---|
| Initial | 5.4 | 10.07 |
| Repeat 1 | 10.4 | 15.07 |
| Repeat 2 | 16.3 | 20.97 |
| Repeat 3 | 23.9 | 28.58 |
| Repeat 4 | 34.8 | 39.44 |

To prevent successful receipt of the message in under 20 milliseconds, two messages would need to be corrupted and rejected. To prevent successful receipt of the message in under 40 milliseconds, five messages must be corrupted and rejected.

## Measured Loading and Noise Tests

We performed tests to measure the effects of loading and noise on two devices. We simulated loading by injecting a continuous pulse-train into the network, with a pulse width of one average message length (640 μsec), and a repetition frequency adjusted to provide the desired loading percentage. This is ill-behaved loading in the sense that it starts at an arbitrary time; if a message is in progress when the pulse starts, the message collides with the pulse. All receivers detect it as a jam signal, and wait for it to clear. If the rising edge of the pulse takes place after the early collision period of the message, then the impact is the same as that of a detected bit-error; the packet is rejected. We used automated test equipment to introduce a GOOSE event every second, and to record the time until the corresponding GOOSE message was properly decoded and processed by the receiving device. The noise-like impact of the loading is modeled by an equivalent channel bit-error rate, as it impacts the GOOSE messages. No non-GOOSE message load is caused by this equivalent BER. The results of the measured loading tests are summarized in Table 5.

**Table 5** Measured Loading Tests

| Loading % | Equivalent Bit-Error Rate($\bullet 10^{-5}$) | Approx. Time (msec) | | | $P_{mc} \bullet 10^{-3}$ for $T_{ac} < 20$ msec |
|---|---|---|---|---|---|
| | | Min. | Max. | Typ. | |
| 10 | 1.56 | 7 | 20 | 13 | 2.32 |
| 20 | 3.13 | 8 | 25 | 13 | 6.21 |
| 30 | 4.69 | 7 | 32 | 13 | 9.45 |
| 40 | 6.25 | 7 | 32 | 13 | 12.20 |
| 50 | 7.81 | 8 | 37 | 15 | 16.60 |
| 60 | 9.38 | 7 | 44 | 15 | 20.70 |
| 70 | 10.94 | 8 | 29 | 16 | 21.00 |

## Security

$P_{uc}$ is the probability of an unwanted command. The IEC-834 Standard recommends applying a square wave of noise to the channel, consisting of 200 milliseconds bursts of white noise and separated by 200 milliseconds of noise-free periods. It is not very probable that this particular test will create a stream of 2592 bits that will be interpreted as a complete, properly framed 320 byte GOOSE message. For noise to simulate a properly framed message that will be accepted by the receiving device, there are 320 bits for bit-pairs and 64 bits of sequence numbers and state numbers, yielding $2^{384}$ legal combinations. So, of $2^{2592}$ total combinations, $2^{384}$ are valid, or one in $2^{2208}$ (approximately one in $4.7 \bullet 10^{729}$)

The more likely mechanism for a false command is that a legitimate GOOSE message will be corrupted to show wrong data. Failure to frame correctly, errors detected by the CRC, errors in the identification, or errors in the framing cause the packet to be rejected. Errors that are undetected by the checksum and change the value of a bit-pair from 01 to 10, or from 10 to 01 cause unwanted commands.

In the tests summarized in Table 5, we assigned 31 user-bits in the message to static known values, and monitored for any GOOSE messages that indicated a different value for the known

points. Throughout all testing, the device under test received no false events. These tests confirm that for systems using GOOSE, the security is very good; dependability is the critical evaluation factor. We also performed the IEC-834 Standard 200 msec square wave noise-burst test; the receiving device detected no unwanted commands, and missed approximately 50% of the commands: $P_{uc} < 10^{-5}$, $P_{mc} = 0.50$.

### Summary of GOOSE Message Comparison to Selected Requirements of the IEC-834 Standard

Table 6 summarizes the comparison of parameters from the IEC-834 Standard for a system using GOOSE messages in the calculated example case, loaded at 30% or less. The last column of Table 6 shows the Bit Error Rate (BER) range needed to meet the stated $P_{mc}$ and $T_{ac}$ requirements. Each BER is derived from the probability of bit errors that cause consecutive GOOSE messages to be rejected within the corresponding $T_{ac}$ limit.

**Table 6** IEC-834 Parameters and GOOSE Messages

| Application | IEC Requirement | GOOSE Over Ethernet |
|---|---|---|
| Blocking | $T_{ac}$<20 msec, $P_{mc}<10^{-3}$, and $P_{uc}<10^{-1}\ldots10^{-2}$ | BER $<1.24 \bullet 10^{-5}$ |
| Permissive Tripping (underreaching) | $T_{ac}$<40 msec, $P_{mc}<10^{-2}$, and $P_{uc}<10^{-3}\ldots10^{-4}$ | BER $<1.24 \bullet 10^{-4}$ |
| Permissive Tripping (overreaching) | $T_{ac}$<40 msec, $P_{mc}<10^{-2}\ldots10^{-3}$, and $P_{uc}<10^{-1}\ldots10^{-2}$ | BER $<6.95 \bullet 10^{-5}$ |
| Direct Tripping | $T_{ac}$<60 msec, $P_{mc}<10^{-3}\ldots10^{-4}$, and $P_{uc}<10^{-5}\ldots10^{-6}$ | BER $< 8.24 \bullet 10^{-5}$ |

## Discussion of Other Cases

The example case is for two devices that use GOOSE messages over a substation network with other devices that are not using GOOSE messages. Varying the assumptions and input parameters varies the results, as described in the following sections:

### Number of GOOSE Messages

The example case has GOOSE Messages from two devices due to the same event. If there are other devices in the station that use GOOSE messages, but the probability is low that a single event will cause GOOSE messages from more than two devices, the system will behave as predicted in the example case. If the same event causes GOOSE messages from more than two devices, then collisions and loading increase due to the additional messages.

If a change in data in one GOOSE message is used to initiate other GOOSE messages, then the additional GOOSE messages increase loading and the probability of collision delays during the period of time that is critical for command dependability.

### Network Loading

The example case uses average loading that is equally distributed in time. Different distributions of the loading change which events are delayed by network traffic, but generally do not significantly impact the average time in which GOOSE messages are successfully received.

### Network Data-Rate

The example case uses a 10 Mbps Ethernet network. The most significant impact of using a 100 Mbps rate is to drop the loading by a factor of 10. For a given network, loading is due to the bits in the packets needed to accomplish all I&C system tasks. The number and frequency of packets does not need to change due to an increase in the network data rate, so the higher speed network effectively reduces the loading by a factor of 10. The time impacts of collisions are also reduced by a factor of 10, because all of the interframe gaps and slot times are reduced by a factor of 10.

### Adjusting GOOSE Repeat Rate

The example case uses the UCA2 Standard repeat rate. The impact of GOOSE traffic on short-term network loading is reduced if the repeat-rate is lower. However, successfully receiving a GOOSE message when noise corrupts an initial message depends on the repeated messages. Using repeat rates that are longer than the UCA2 specification reduces the probability that a command will be received within 20 milliseconds, thus reducing the dependability.

## EQUIPMENT AVAILABILITY

Availability is the percentage of time that a system is capable of performing an action when commanded to act. Reference [5] uses fault trees and contrasts the availability of local point-to-point serial communications with Ethernet networks, for relay-to-relay communications. Table 7 is replicated from Reference [2] to illustrate the relative availability of several example topologies. These availabilities are based on the probabilities of equipment failure, using representative MTBF and MTTR data. Serial point-to-point communications are more available than single and redundant Ethernet LANS.

**Table 7** Relay-to-Relay Communications in a Substation

| Communications Topology | Availability % | Predicted Annual Hours Out of Service |
|---|---|---|
| Ethernet Switches | 99.7138 | 25.0 |
| Ethernet Shared Hubs | 99.8778 | 10.7 |
| Ethernet Redundant Switches | 99.9991 | .07 |
| Ethernet Redundant Servers, Routers, Switches | 99.9995 | .04 |
| Ethernet Redundant Shared Hubs | 99.9998 | .01 |
| Serial Point-to-Point | 99.9999 | .00014 |

## UNAVAILABILITY DUE TO INTEGRATION PROBLEMS

Beyond equipment failures, there are other factors that can reduce availability. When only two intelligent configurable nodes are involved in a data transaction there is a finite probability that an error will be made configuring one or both of the devices so that it will not operate correctly. Generally, these types of errors can be detected and eliminated with proper factory acceptance testing and commissioning testing.

As you increase the number of intelligent devices in a network, the number of opportunities to make configuration errors increases, as does the number of possible interactions. When enough devices and transactions share a network, there is a point where it is not feasible to fully simulate the network and all data transactions that might occur. At this point an additional unavailability can occur due to a combination of configuration and untested interactions between devices. The probability of a configuration error is characterized by Equation 6:

$$P_{CE} \equiv [\sum_{1}^{d} P_d(C_d, U_d, T_d)] + P_{IE}(d, T_{IE}) \quad (6)$$

where:

- d is the number of devices.
- $P_{CE}$ is the probability of a configuration error or integration error that is not detected before live system deployment.
- $P_d$ is the probability of a configuration error in device d that is not detected before live system deployment.
- $C_d$ is the complexity of device d.
- $U_d$ is the understanding of device d by the person doing the configuration.
- $T_d$ is the percentage of actual cases for device d that can occur that are tested prior to the measurement period.
- $P_{IE}$ is the probability of an interaction problem that is not detected before live system deployment.
- $T_{IE}$ is the percentage of actual cases for interaction problems that can occur that are tested prior to the measurement period.

We identified the major independent variables influencing integration related problems, to identify the issues involved. The quantification of these variables and details of these functions are beyond the scope of this paper. For a simple two-device, point-to-point system, it is generally feasible to validate all interactions prior to live deployment. As the number of devices and interaction modes increases in a multi-device system, the analysis is increasingly complex, and the probability of an interaction error undetected by verification testing increases.

## CONCLUSIONS

1. Characterizing GOOSE message delays over an Ethernet network is a complex task with many independent variables. Two-node point-to-point communications, as generally addressed by IEC-834, are much easier to model.

2. GOOSE messages on a reasonably loaded, noise-free Ethernet network can operate with sufficient speed for many high-speed automation and teleprotection applications.

3. Network loading includes an independent and dependent component, both critical to determining delay times. Exercise care to analyze, control, and understand the loading on the network, especially during events when the communication is most important.

4. When noise is injected into an Ethernet network that uses GOOSE messaging, the system may not meet the IEC-834 Standard dependability specifications for teleprotection. You should determine if noise in your Ethernet network is probable. Even if you design the network to minimize electrical noise, you must determine the degree to which you require compliance with IEC-834 Standard for your application.

5. Serial point-to-point connections are more reliable (i.e., the equipment has higher availability) than Ethernet networks. For each application, analyze the acceptable availability for mission-critical high-speed control.

6. The complexity of multiple-device networks can decrease the system availability due to integration interactions that are not adequately tested in feasible-duration commissioning tests.

## REFERENCES


[1] CEI IEC-834 International Standard, Performance and Testing of Teleprotection Equipment of Power Systems, International Electrotechnical Commission, 1988.

[2] G.W. Scheer and D.J. Dolezilek, "Comparing The Reliability Of Ethernet Network Topologies In Substation Control And Monitoring Networks," Western Power Delivery Automation Conference, Spokane, Washington, April 4–6, 2000.

[3] KC Associates, UCA 2.0: Generic Object Models For Substation And Feeder Equipment, Version 0.91, Electric Power Research Institute, February 5, 2000.

[4] G.W. Scheer, "Answering Substation Automation Questions Through Fault Tree Analysis," Proceedings of the 4th Annual Texas A&M Substation Automation Conference, College Station Texas, April 8–9, 1998.

[5] D.R. Boggs, J.C. Mogul, C.A. Kent, "Measured Capacity of an Ethernet: Myths and Reality," Western Research Laboratory.

[6] B. Whetten, S. Steinberg, D. Ferrari, "The Packet Starvation Effect in CSMA/CD LANS and a Solution," University of California at Berkeley.

[7] J.A. Whatley, "Hydro One Networks, Inc. UCA 2.0 Laboratory Demonstration," UCA Substation Communications Demonstration Meeting, Andover, MA, September, 2000.

## BIOGRAPHIES

**Gary W. Scheer** received his B.S. in Electrical Engineering from Montana State University in 1977. He worked for the Montana Power Company and the MPC subsidiary, The Tetragenics Company, before joining Schweitzer Engineering Laboratories, Inc. in 1990 as a development engineer. He has served as Vice President of the Research and Development Division, and of the Automation and Engineering Services Division of SEL. Mr. Scheer now serves in the Marketing and Customer Services Division as Product Manager for automation and communications products. His biography appears in Who's Who in America. He holds two patents related to teleprotection. He is a registered professional engineer and member of the IEEE, NSPE, and the ISA.

**Darold Woodward** has a B.S. in Electrical Engineering from Washington State University. He is a member of the Instrument Society of America (ISA). He joined Schweitzer Engineering Laboratories in 1998 in the position of System Integration Engineer. He was with the consulting firm HDR Inc., for six years where he participated in design and commissioning projects for electrical, automation, and instrumentation systems in water, wastewater, and hydroelectric facilities. Before joining HDR Inc., he was with R. W. Beck and Associates assisting with the design of electrical and instrumentation systems for substations, wastewater, and hydroelectric facilities.