21, rue d'Artois, F-75008 PARIS
http : //www.cigre.org

CIGRE US National Committee
2022 Grid of the Future Symposium

T ComEd's 9 Box Method for Goose Messaging

J. BETTLER
Commonwealth Edison Company
USA

SUMMARY

ComEd recently embarked on a Digital Smart Substation (DSS) program. One of the program's main objectives is to implement technologies that could increase awareness of substation issues, improve performance, and reduce / eliminate preventive maintenance (PM) activities.

Our DSS sites utilize IEC 61850 GOOSE Messaging, advanced HMI deployment, highly secure networks and innovative SCADA. One site also incorporated Sampled Values (SV). Our DSS design can be applied at any voltage level or topology. And although we were implementing new technologies, standardization and simplicity were a core design objective.

Of the technologies applied in our DSS program, GOOSE Messaging has the largest footprint and greatest impact to the overall station design. But after beginning the conceptual design for our program, we quickly realized there is no clear way to lay out your GOOSE tripping. Additionally, there is no standard approach to documenting or testing a GOOSE system.

As a result, we developed the "9 Box Method". This method utilizes a systematic approach to developing all GOOSE messaging and the technique can be applied at any voltage level

The basic design premise resides in the following two facts for our GOOSE Messaging

- Most relays receive GOOSE messages from a limited number of relays (between 3-9)
- There are very few actual functions being sent between these relays at a substation (between - 10) and of these, there were core functions present at all station, like Breaker Failure (BF).

We then used digital variables 10-19 as our outputs and their Generic Input/Output Logical Node (GGIO) counterparts in our Substation Configuration Description (SCD) file and digital variables 20-29 as our inputs. Standards were implemented to lock in certain variables and provide consistency at different station voltage levels and types.

This ultimately assisted us in two critical aspects:

- The development of ComEd's 8500 Print Series
- Testing features built into the relay logic to verify GOOSE functionality

This paper describes how to systematically implement the 9 Box Method, breaking it down for different station and equipment types. It will also describe the 8500 prints and the approach we took to simplify testing a GOOSE network for our field forces.

KEYWORDS

Digital Smart Substation, GOOSE Messaging, 8500 Prints, IEC 61850

john.bettler@comed.com

ComEd recently embarked on a Digital Smart Substation (DSS) program. One of the program's main objectives was to implement technologies that could increase awareness of substation issues, improve performance, and reduce / eliminate preventive maintenance (PM) activities.

Our DSS sites utilize IEC 61850 GOOSE Messaging, advanced HMI deployment, highly secure networks and innovative SCADA. One site also incorporated Sampled Values (SV). Our DSS design can be applied at any voltage level or topology. And although we were implementing new technologies, standardization and simplicity were a core design objective.

Of the technologies applied in our DSS program, GOOSE Messaging has the largest footprint and greatest impact to the overall station design. But after beginning the conceptual design for our program, we quickly realized there is no clear way to lay out your GOOSE tripping. Additionally, there is no standard approach to documenting or testing a GOOSE system.

To solve this problem, we developed the 9 Box Method, which standardized and controlled our GOOSE traffic and allows for simple easy to utilize prints. We call these our 8500 prints.

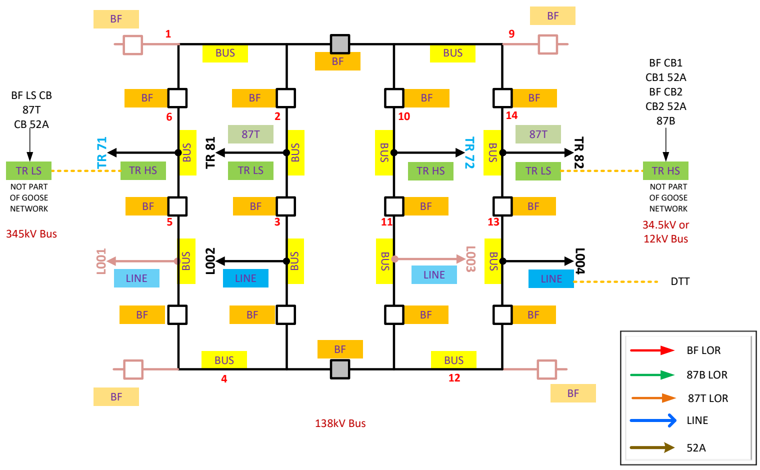
9 Box Method Development

The approach requires the following six steps to develop the substations GOOSE logic

1. Review station one line, the proposed relay schemes and their Zone Of Protection (ZOP's)
2. Identify contacts functions to be used if the system was a traditionally wired station.
    a. OUTPUTS: Contact Closures are considered Sent or Published functions
    b. INPUTS: Coils or inputs pick up are considered Receive or Subscription functions
3. Assign a digital variable for each function identified in step 2a and determine which relays perform that function. These OUTPUTS become Published GOOSE Messages in the SCD.
4. Determine what function each relay should receive and who is sending it to them. These INPUTS will be the Subscriptions in the SCD file. Note it is designed that the Publishing relay be given a device number relative to the Subscribing relay.
5. Create a template setting for each relay type, developing a standard Publisher / Subscriber relationship for the worst case
6. Map all Publisher / Subscriber functions between all relays in a spreadsheet.

Step 1 → General Layout

During this step, relays and their protective functions are identified. This should consider the ultimate station design. Also, during this step, all interface relays and merging units should be identified. For our GOOSE only stations, we call the Circuit Breaker (CB) BF relay the merging unit because it Trips / Closes the CB and also grabs CB status information. Below is a 138kV GIS Station for an example.



Functions

- BF (Merging Unit)
- LINE Relay
- BUS Relay
- 87T Relay
- TR HS Relay Interface
- TR LS Relay Interface

FIG 01

Step 2 → Contact Tripping

During this step, the objective is to determine what trips relays would be sending (Outputs) or receiving (Inputs) as if it was a wired scheme. Note this includes evaluating any Lock Out Relay (LOR) or other aux relay. This provides a road map for any GOOSE Trips sent or received.

A few observations:

- Keep System 1 (S1) and System 2 (S2) relays isolated from one another on their GOOSE LAN. This keeps the GOOSE network as small as possible. However, some S1 & S2 relays may need to communicate with each other like the BF/Reclosingelay. In this case, do the following:
    - Proprietary Protocol between relays. (This is what we did this using serial port)
    - Allow GOOSE Communication only between these two relays only
- Keep different station voltage levels isolated from one another and on separate LAN's.
    - If both voltage levels are operating a GOOSE network, establish interface relays that are part of their voltage's level network, but communicate via a proprietary protocol or via a line differential
    - If the different voltage area is wired, install an interface relay that converts your GOOSE to wired. Note if you do not want to run the IP cable to that area, the proprietary solution is also an option
- This evaluation needs to consider the ultimate station topology and worst case for any relay.
- Some function may only reside on S1 (or S2) as they are not required to be redundant
- Direct Transfer Trip (DTT) between stations is not a GOOSE function

So, let's look at a BF protection relay. Using colored arrows to indicate function, the picture below shows the GOOSE Publishing (Send) on the left and GOOSE Subscribing (Receive) on the right. Note the trip into the BF relays can initiates both BF and / or reclosing. In this manor, it is straight forward to evaluate all the relays in a system. Note that the BF relays have different possible tripping diagrams based on what CB they are assigned and every case must be checked (two cases shown)

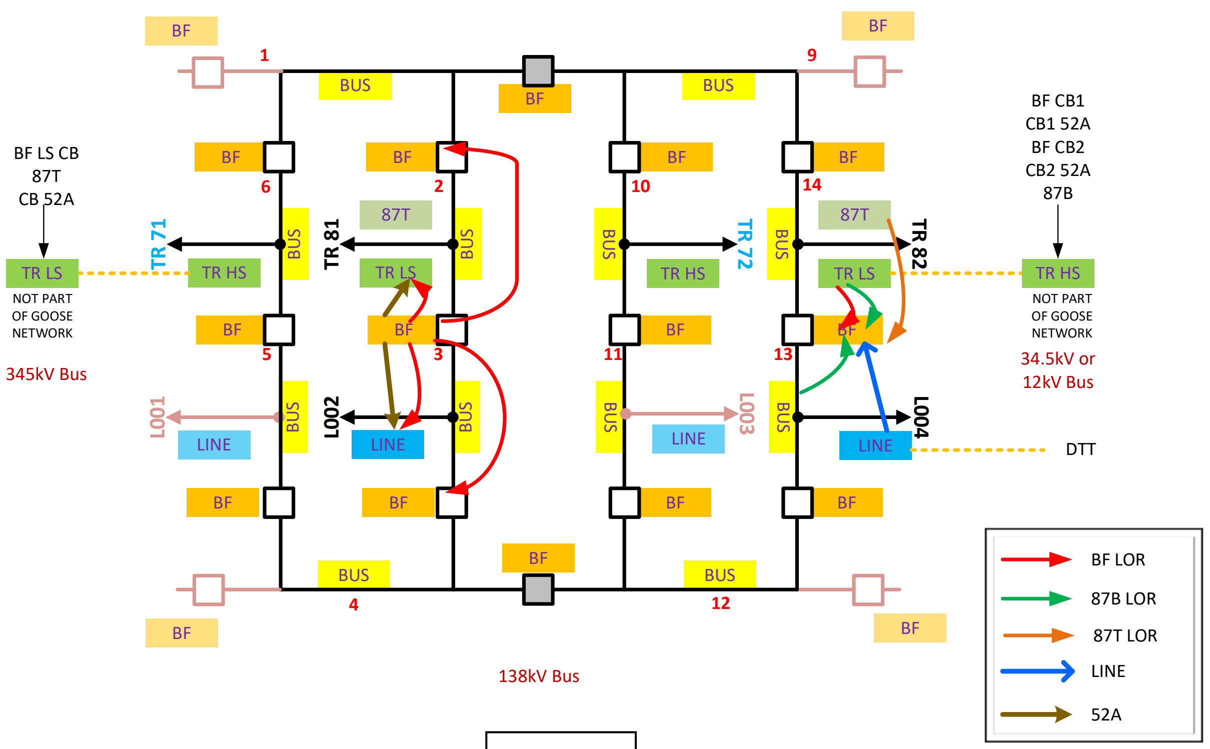


FIG 02

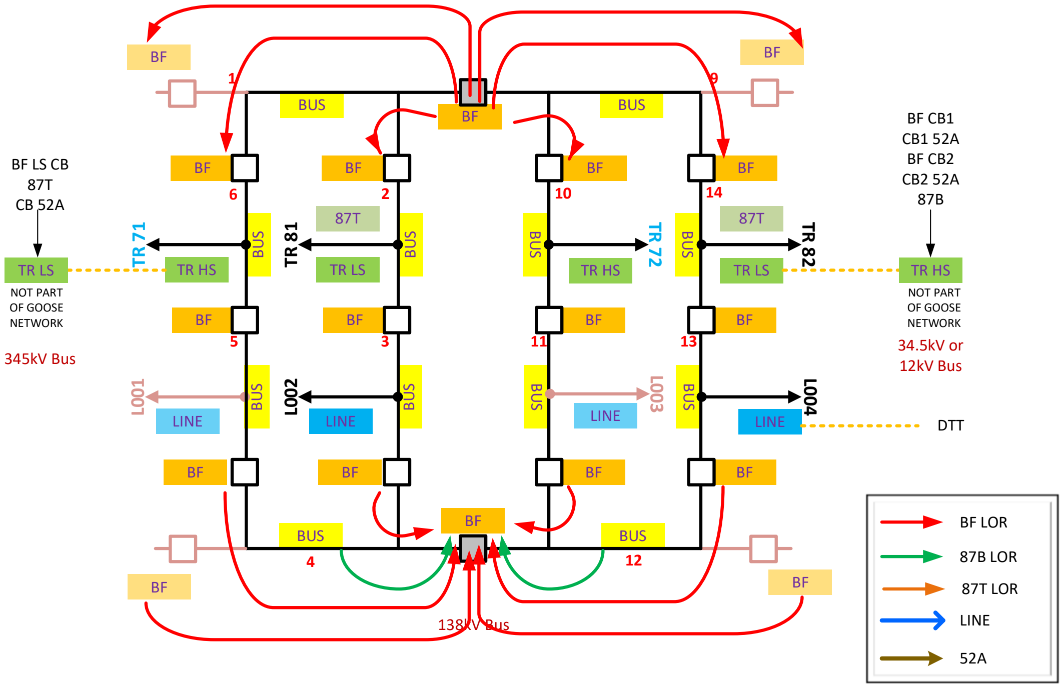


FIG 03

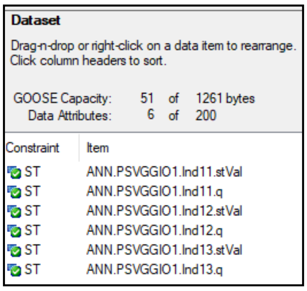
Step 3 → Subscribing

After reviewing all the expected interchanges between relays, we recognized that there were about ten (10) actual contact functions being shared between devices at any given station. And of these ten, that the following five functions were found at all stations: 1) CB Status 2) BF LOR (Lock Out Relay) 3) Bus Differential (87B) LOR 4) Transformer Differential (87T) LOR 5) General Trip. Since there are no 61850 objects for LORs, GGIOs were used in the Dataset for all published GOOSE Messages. Note modifying a relay's IED Capability Description (ICD) file was discussed but was considered a bad practice, as someone may fail to recognize the ICD change..

As a result, we developed a standard GOOSE Publishing table for each station type. This table has a column for each contact function and an assigned relay variable from PSV10 thru PSV19. Note, PSVxx's are digital variables inside the relay. Shown below is the table for a GIS station, with the underlined functions and their variable numbers common among all stations.

	Publisher (This Relay)	Block	G Check	52A	BF LOR	87B LOR	87T LOR	TOC	Overload	Line	Auto Close
Relay Px	PSV in Publisher Relay	PSV10	PSV11	PSV12	PSV13	PSV14	PSV15	PSV16	PSV17	PSV18	PSV19
Relay Px	Dataset ANN.PSVGGIO1	Ind10	Ind11	Ind12	Ind13	Ind14	Ind15	Ind16	Ind17	Ind18	Ind19

As such, each relay in the station would publish the appropriate variables from this tables based on that relays required functionality and unused function would be zero out. This method allows one to determine what a relay is publishing in the SCD file just by looking at the settings file.



For example, our GIS BF relay only publishes three variables: PSV11 = GOOSE Check, PSV12 = 52A, PSV13 = BF LOR. The other six would be zeroed out. Resulting in the GOOSE data set to the right and Publishing table below

	Publisher (This Relay)	Block	G Check	52A	BF LOR	87B LOR	87T LOR	TR TOC	Overload	Line	Auto Close
Relay Px	PSV in Publisher Relay	PSV10	PSV11	PSV12	PSV13	PSV14	PSV15	PSV16	PSV17	PSV18	PSV19
Relay Px	Dataset ANN.PSVGGIO1	Ind10	Ind11	Ind12	Ind13	Ind14	Ind15	Ind16	Ind17	Ind18	Ind19

FIG 04

Step 4 → Publishing

Since contact functional quantities were being published via PSV 10 – 19 by the sending relay, we decided to map subscribing quantity in the receiving relay to PSV 20 – 29, matching the last digit based on function. So, for example. BF would be sent as PSV13 and received as PSV23.

Additionally, intermediate variables used in the SCD also match this pattern. The relays we use have 256 Virtual Bits (VB001 thru VB256) per relay in the SCD file. These VB are used to link the Publishing Relay's Dataset variable to the Subscribing Relay. To do this, we reserved VB010-VB019 for publishing relay 1, VB020-VB029 for publishing relay 2 … to VB090-VB099 for relay 9. Again, the last digit from the virtual bit aligns with the contact function. Also, VB101 – VB109 were reserved for the quality bits from each publishing relay.

So based on this layout, nine devices can each publish any of the ten contact functions to our subscribing relay. Thus the name "9 Box". Note this could be expanded up to 18 devices.

The last step was to assign the required VB to the subscribing relays corresponding PSV. So continuing with our example, PVS23 = VB013 OR VB023 … OR VB073. Thus, this relay can receive a BF LOR (contact function 3) from up to seven devices. And as shown in the table below, this assessment was done for all of the functions.

So with this approach, the last digit provides the contact function and the first digit tells you the device number sending that function. The only thing left to sort out is how to establish the device numbering for a given subscriber. The most important aspect to this is that a consistent repeatable system be employed. As shown below, BT (i.e. BF) are assigned to devices 1 thru 7, with the lowest BT number going first. Lines and transformers were assigned to positions 4/5 and 6/7 respectively. Bus differential were assigned positions 8/9 based on voltage input connection. Note that for our topology, BT's on a vertical rung can only receive BF from at most 4 other BT, after which it is receiving something from a line or transformer.

This pattern works for our 138kV GIS example, but would not work for an Indoor SWGR topology. So each topology is different. The important step is that a constant approach be used that could be explained to another engineer (and repeated)

PUBLISHING RELAY		SUBSCRIBING Relay Virtual Bit Assignment. VB's are OR'ed together in the respective PSV if relay Subscribes to that function.										
Device #	Relay Type	Block	G Check	52A	BF LOR	87B LOR	87T LOR	TOC	Overload	Line	Auto Close	Goose Fail
		PSV20	PSV21	PSV22	PSV23	PSV24	PSV25	PSV26	PSV27	PSV28	PSV29	AST04Q
D01	BT01 (Smallest BT # 1st)	VB010	VB011	VB012	VB013	VB014	VB015	VB016	VB017	VB018	VB019	VB101
D02	BT02	VB020	VB021	VB022	VB023	VB024	VB025	VB026	VB027	VB028	VB029	VB102
D03	BT03	VB030	VB031	VB032	VB033	VB034	VB035	VB036	VB037	VB038	VB039	VB103
D04	BT04 or LINE02	VB040	VB041	VB042	VB043	VB044	VB045	VB046	VB047	VB048	VB049	VB104
D05	BT05 or LINE01	VB050	VB051	VB052	VB053	VB054	VB055	VB056	VB057	VB058	VB059	VB105
D06	BT06 or TR02	VB060	VB061	VB062	VB063	VB064	VB065	VB066	VB067	VB068	VB069	VB106
D07	BT07 or TR01	VB070	VB071	VB072	VB073	VB074	VB075	VB076	VB077	VB078	VB079	VB107
D08	BUS02	VB080	VB081	VB082	VB083	VB084	VB085	VB086	VB087	VB088	VB089	VB108
D09	BUS01	VB090	VB091	VB092	VB093	VB094	VB095	VB096	VB097	VB098	VB099	VB109

FIG 05

It should be acknowledged that interoperability was NOT a primary driver for our design, and that relays were selected based on our team's functional experience with the protection devices.

Step 5 → Template Settings

The backbone of a traditional template setting is the Inputs / Outputs. With GOOSE, this is replaced with Publisher / Subscriber relationships we developed in Step 3 and Step 4. Basically, we assign the appropriate function to its digital variable to PSV10-PSV19 to be Published. And then we assign the appropriate Virtual Bits to its digital variable to PSV20-PSV29 as Subscriptions. Again, once completed, his should match the tables developed in the above steps.

Below is the template setting logic for a BF Publisher and Subscriber relationships.

```
#___________PUBLISHER    ---SENT TO OTHER RELAYS
PSV10 := 0
PSV11 := PLT11 #   GOOSE CHECK SEND
PSV12 := 52AA1 #   CB STATUS SEND
PSV13 := (PLT12 AND PLT13) #   86BF LOR TRIP SEND
PSV14 := 0
PSV15 := 0
PSV16 := 0
PSV17 := 0
PSV18 := 0
PSV19 := 0
#
#___________SUBSCRIBER   --- RECEIVED FROM OTHER RELAYS
PSV20 := 0
PSV21 := VB011 OR VB021 OR VB031 OR VB041 OR VB051 OR VB061 OR VB071 OR VB081 OR VB091 #   GOOSE CHECK RCVD
PSV22 := VB012 OR VB022 OR VB062 OR VB072 #   CB STATUS RCVD
PSV23 := VB013 OR VB023 OR VB033 OR VB043 OR VB053 OR VB063 OR VB073 OR PCT13Q #   86BF LOR TRIP RCVD. LOCAL TEST VIA LB03
PSV24 := VB084 OR VB094 OR PCT14Q #   86B LOR TRIP RCVD; MAPPED TO TRIP EQ.  LOCAL TEST VIA LB04
PSV25 := VB065 OR VB075 OR PCT15Q #   86T LOR TRIP RVCD; MAPPED TO TRIP EQ. LOCAL TEST VIA LB05
PSV26 := 0
PSV27 := 0
PSV28 := VB048 OR VB058 OR PCT18Q #   TRIP (RELAY OR DTT) RCVD; MAPPED TO TRIP EQ. LOCAL TEST VIA LB08
PSV29 := 0
```

FIG 06

A few observations, since we are publishing digital variable, we can assign test bits to these the variable to simulate sending or receiving a GOOSE message. For instance, PSV23, receive a BF, has timer PCT13 mapped to it. This timer function will provide a 5 second BF trip when initiated. Similarly, 52A has a latch mapped to it to simulate a closed CB, which would then get published out.

These test features are controlled by Digital Test Switches inside the relay and are only allowed to be used if the relay is in Test Mode, which helps minimize Human Performance Errors. As an alternative, Virtual Test Switches could also be used, where you wire Positive – Test Switch – Input – Negative. Since this uses actual test switches, some employees may like this better

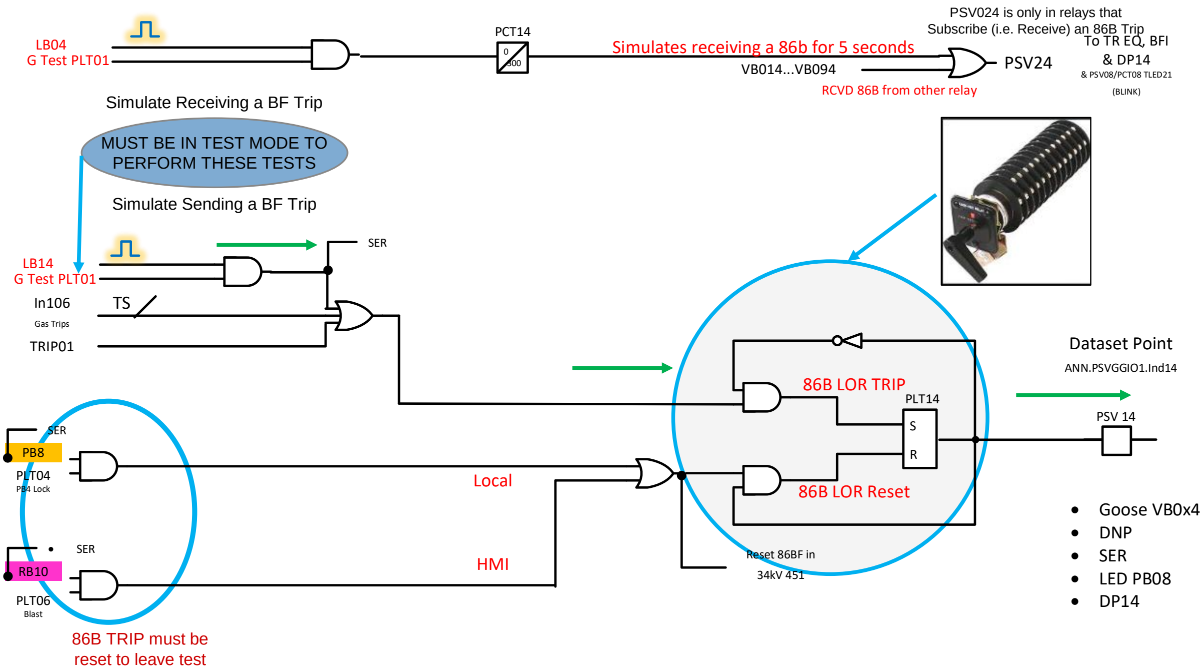


FIG 07

Step 6 → Virtual Bit Table

The final step is building out the Virtual Bit Table. Here subscribers get mapped to their respective publishers. Each publisher is assigned a relative device number based on the criteria established in Step 4, which establishes which row it will fill (1 thru 9). The columns in that row corresponding to the publishers functions are then populated with a Virtual Bit. The Virtual Bits are numbered so that the first number is the device and the second is the function. So if a BF relay is device #2 to a line relay, the line relay will receive VB021 (GC), VB022(52A), VB023 (BF) from the BF Trip.

This table is basically the SCD file relationships created in an Excel spreadsheet without any of the network information. Since it is a spreadsheet, it is simple to developing the Publisher / Subscriber relationships and then checking them against your topology using filters and sorting features. Additionally, when the final SCD file is completed, it can be exported as a spreadsheet and compared against the VB table. This is critical, as missing a connection is the same as forgetting a cable.

Subscriber	Publisher	D #	Publish Type	G Chec	52a	BF Trip	86B Tri	86T Tri	Trip	Quality
BT 10-11	BT 9-10	1	BF	VB011		VB013				VB101
BT 10-11	BT 11-12	2	BF	VB021		VB023				VB102
BT 10-11	TR83	7	TR	VB071		VB073		VB075		VB107
BT 10-11	B11	8	BUS - Y	VB081			VB084			VB108
BT 10-11	B10	9	BUS - Z	VB091			VB094			VB109
BT 4-12	BT 3-4	1	BF	VB011		VB013				VB101
BT 4-12	BT 4-5	2	BF	VB021		VB023				VB102
BT 4-12	BT 11-12	3	BF	VB031		VB033				VB103
BT 4-12	BT 12-13	4	BF	VB041		VB043				VB104
BT 4-12	B12	8	BUS - Y	VB081			VB084			VB108
BT 4-12	B4	9	BUS - Z	VB091			VB094			VB109
L004	BT 12-13	1	BF	VB011	VB012	VB013				VB101
L004	BT 13-14	2	BF	VB021	VB022	VB023				VB102
L004	B13	9	BUS	VB091			VB094			VB109
TR82	BT 1-2	1	BF	VB011	VB012	VB013				VB101
TR82	BT 2-3	2	BF	VB021	VB022	VB023				VB102
TR82	B2	9	BUS	VB091			VB094			VB109

FIG 08

ComEd’s 8500-series Prints

The VB table created in Step 6 is used to develop ComEd’s GOOSE prints. We refer to them as the 8500 Print Series.

The print replaces the DC Tripping print. The Top of the relay shows any GOOSE messages this relay is subscribing to (aka inputs), their function and where they are coming from. The bottom of the relay shows any GOOSE messages this relay is publishing (aka outputs), their function and where they are going. This is a graphical representation of the VB Table / SCD Connections. Also, any serial communication bits were indicated (right side).

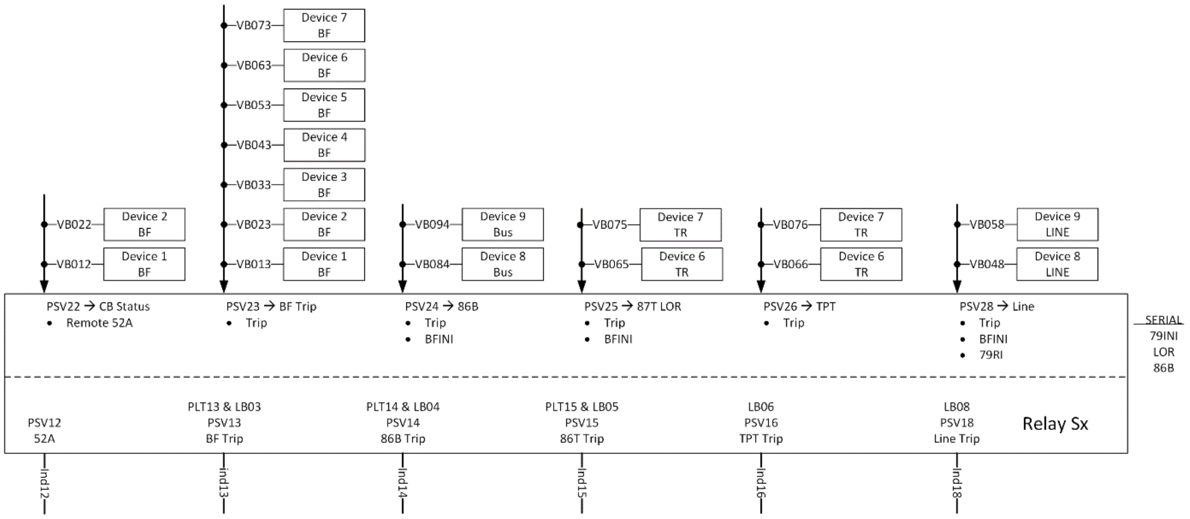


FIG 09

When commissioning the station, these prints are used to identify GOOSE signals that need to be sent and received by a given relay. Each function is highlighted once it is proven operational. Note these tests should include verification while in Op Mode (1) and Test Mode (3).

Conclusion

Overall, ComEd’s experience with the IEC61850 GOOSE Tripping at our DSS stations has been positive. The application of the 9 Box Method leverages a simple approach to our GOOSE Tripping. The development of standard functions and repeatable device numbering allowed up to easily build a Virtual Bit Table, which is basically the mapping relationships in the SCD file. The 8500 prints, which are just a visualization of the Virtual Bit Table, allowed for prints that clearly show field what is taking place for a relay. Finally, because the design used digital variable for publishing and subscribing, we were able to add in GOOSE Test Bits into our settings for advanced testability.