

Real Time Application using multicast Ethernet in Power Substation Automation according to IEC61850

Julio Cezar de Oliveira¹, Walter Augusto Varella², Antônio Eduardo Marques³, Gustavo Forster⁴

^{1,4}ABB – Substation Automation
 Av. Monteiro Lobato, 3411, Vila são Roque – CEP 07190-904 – Guarulhos – SP – Brasil

²CEFET – Centro Federal de Educação Tecnológica de São Paulo
 Rua Maria Cristina 50 – CEP 11510-040 – Cubatão – SP – Brasil

³UNICSUL – Universidade Cruzeiro do Sul
 Av. Dr Ussiel Cirilo, 225 – CEP 08060-070 – São Paulo – SP – Brasil

{julio.zoliveira@gmail.com, varella@cefetsp.br, aesilva@unicsul.br, gustavo.forster@br.abb.com}

Abstract — *Abstract: The Ethernet technology has been accepted with strongly success since its creation in the 70s. That good acceptance also regards SCADA applications, in reference of electrical substation automation applications, displacing other process bus technologies such as Modbus and DNP, or forcing then to migrate to Ethernet based environments, originating “Modbus over TCP/IP” and “DNP TCP/IP”.*

However, Ethernet was not widely indicated to real time messages delivery because the CSMA/CD itself limitations. But, with the introduction of new aggregated standards like RSTP protocol, VLANs, speed increase and flow control in switched systems, the Ethernet become a reliable network technology for that application.

Recently the IEC61850 standard once has accomplished real time application under an Ethernet network.

Key words: IEC61850, GOOSE, multicast, Ethernet, Substation Automation, IED

INTRODUCTION

Substation automation actual scenery lead us to a SCADA systems, RTU (Remote Terminal Unit), control and protection relays (IED - Intelligent Electronic Devices) consolidation, and their functionalities into a single system [1], taking into account the digital communication and the interaction among themselves. A RTU can be a communication gateway, a control and protection relay (IED) can be used to acquire and distribute data and an SCADA system can have resources that before was only possible with dedicated equipment.

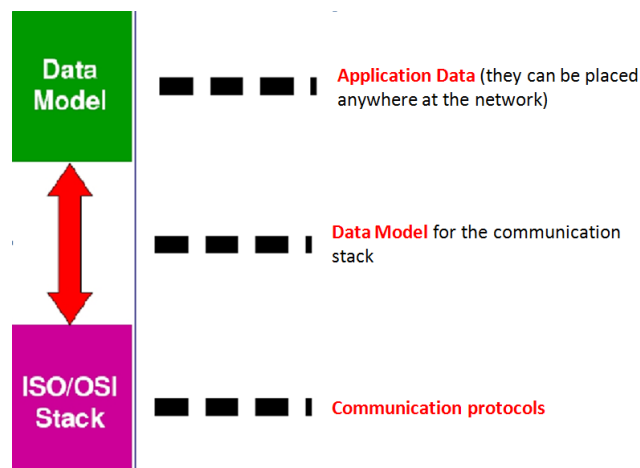
Even though a physical integration of these functions can occur in a near future, a new concept was introduced through the IEC61850 standard. This standard separate

the application model, data transfer service and communication protocols, treating them individually. This way, these functions can be grouped or distributed in a network among devices, being them IEDs, RTUs or SCADA systems

The IEC61850, as it is Ethernet based, allows that real time applications to coexist with another services also *Ethernet* based as FTP, phone application or video etc.

Concerning to real time services the IEC61850 has 4 mechanisms: **GOOSE** (generic object oriented substation event), **GSSE** (generic substation state event), **IEC61850-9-1** (Sampled Analog Values over serial point to point link) e **IEC61850-9-2** (Sampled Analog Values over ISO/IEC8802-3). This article treats exclusively of GOOSE mechanism [2].

The following picture shows the separation of the modeling / mapping of data and communication protocols:



PICTURE 1
 IEC61850 function division

REAL TIME APPLICATION

In a power substation, a few milliseconds is the time that the equipment has to turn off a circuit that presents a fail, be it an overload, short circuit, over voltage or any other motive that places in danger the system stability or the physical integrity of the switchgear like transformers, breakers disconnectors or a transmission line.

The substation system protection is done by protection relays actually named IEDs Intelligent Electronic Devices. This name reflects the devices characteristic of having multiple functions in one equipment.

An IED has resources to make bay control in a substation (open or close a breaker and disconnectors) besides the protection function. May also execute complex logics for decision take of system recomposition or multiple automatism.

The IEDs also has communication interfaces (serial or optical transceivers) to transmit and receive data to or from an SCADA system. There are many communication protocols like for example MODBUS, LON, PROFIBUS and DNP.

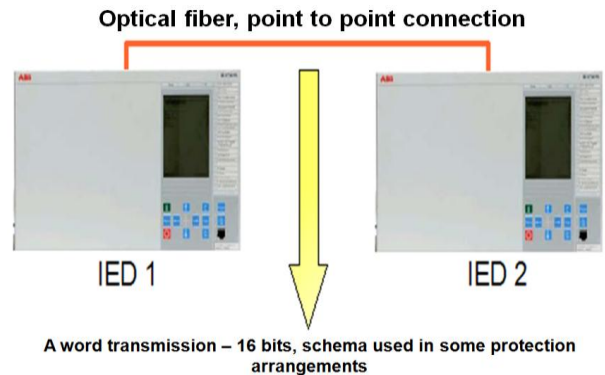
With the arrival of IEC61850 standard, there is an strong trend of convergence of main manufacturers in the substation automation area to adopt this standard [3], due to the possibility of Ethernet and TCP/IP use, among other important characteristics that emphasizes it's use.

One of the great benefits taken by IEC61850 standard was the possibility of real time communication for the high priority messages. Application where traditionally the interconnection between IEDs were made by wire by means an electrical circuit using communication interfaces and proprietary communication protocols, now can have a very good solutions using an optical Ethernet network. Is just here that the **GOOSE** takes action

The IEDs normally have digital I/O boards. When one of the digital I/O change it's state (0 to 1 for example), a high speed multicast message (GOOSE) is reported by a publish x subscriber mechanism (approached later in this article). That messages in their first transmission, takes approximately 5 milliseconds. The message is also repeated in 2, 4, 8 to 60,000 milliseconds interval. In this way the possibility of message delivery is very high.

When making protection and control systems for power substations, it's very common to have situations where we have to send signals from one digital input from one IED to another.

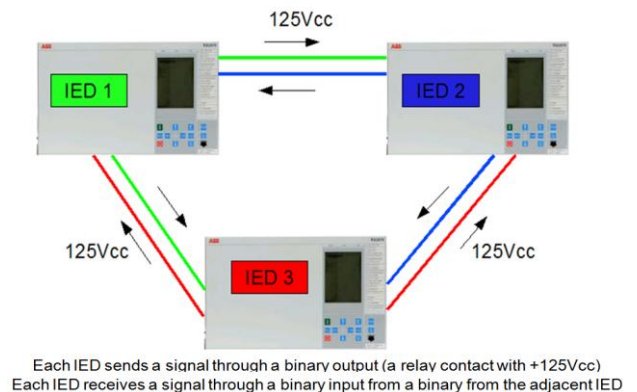
Traditionally, signals that are of high priority (as those which are used to open a breaker in 500kV substation, just as na example) are transmitted by a direct electrical connection (copper wire) or in a optical peer to peer link, as shown in picture 2:



PICTURE 2

Information transmission from one IED to another in a peer to peer link

Picture 3 shows na interlocking scheme (typical automatism of a power substation) implemented via an electrical circuit, exclusively for digital signal exchange among IEDs that needs to be acquired in real time (critical mission application):



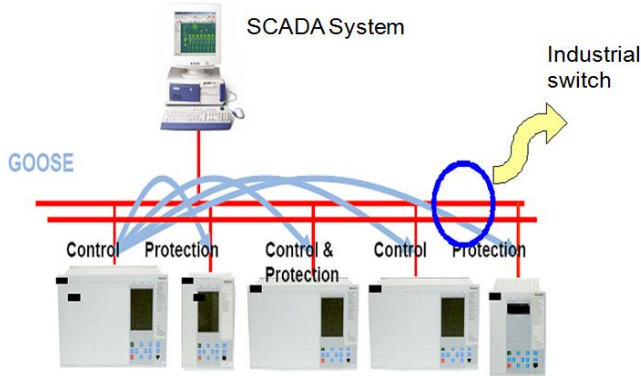
PICTURE 3

Send and receive signal among IEDs via electrical circuit

Both arrangements can be implemented with the use of sending multicast messages like GOOSE.

Considering a large power substation, with a large number of circuits with complex automatism logics configured in many IEDs and suppose that these IEDs has to exchange information among themselves in real time because of the severity of the application, here GOOSE is a relevant solution.

Engineering can be highly simplified due to the possibility of elimination of the conventional copper wiring, being not necessary, thus, foreseen this wiring in the project developing phase, or even not consider the installation costs of dedicated interfaces to achieve this function. Picture 4 sows a topology that is an alternative to the solution shown in picture 3, with the IEC61850 standard and GOOSE messages:



PICTURE 4

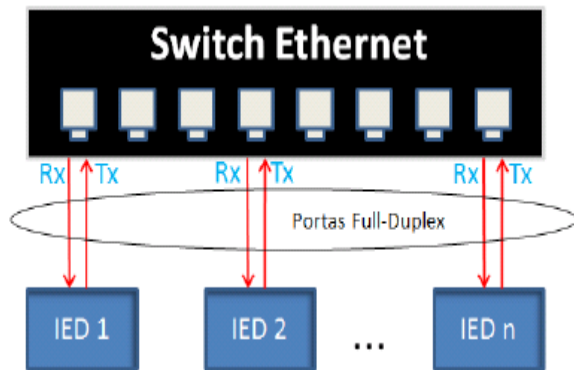
Sending GOOSE messages in an Ethernet Network

A series of factors are essential for a real time application considering an Ethernet network, as follow [1] [4]

a) IEEE 802.3x standard: Full duplex operation

A full duplex interface present at modern switches, practically double the available bandwidth. This means that in the fast Ethernet pattern this bandwidth probably takes 200 Mbps, as the interface can receive and transmit data simultaneously.

Moreover the bandwidth, a significantly characteristic for real time application over Ethernet is a medium without collisions due to the establishment of a direct connection between the emitter and receiver of the messages. These connections are possible via MAC address. In this way the traffic between a port and another do not overflow the others, not interfering in their performance. This is a relevant fact, as when we have a collision, data may not be transmitted immediately, compromising the application.



PICTURE 5

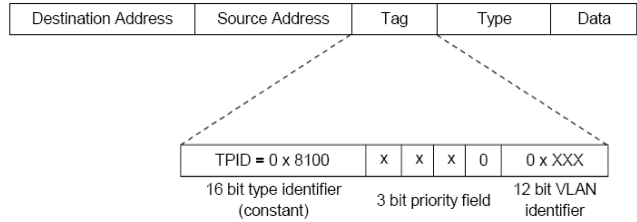
Switches with full duplex ports do not have collisions.

b) IEEE 802.1p standard: Critical messages transmission through de QoS

Ethernet is multiple application network. Each application that pass through the medium can have variable data reception time. Because of that without the advent of 802.1p

standard wouldn't be possible that a critical message like GOOSE to coexist with other application in a network.

The 802.1p standard inserts a 4 bytes tag in the original Ethernet header as can be observed in the Bellow Picture.:



PICTURE 6

Ethernet Frame with priority tag

The priority row number can vary according to manufacturer / model f the switch. At least 2 rows are needed to use 802.1p. In this way no low priority message is transmitted until all high priority messages have been sending.

With the segregation of messages by priority, it's possible to preserve the bandwidth to what is considered "critical mission", assuring its delivery. Meanwhile there are latency cases that need be considered, for example:

- Assume that a GOOSE message arrives to one of the switch port while a non priority message is in transmission what would happen?

GOOSE has to wait. However it's possible to calculate this delay. The calculation basis is over the bigger Ethernet frame – 1518 bytes.

A maximum size frame takes 122µs to be send. A 300 byte GOOSE message takes approximately 24µs. Considering, for example that there are more 5 GOOSE messages to be send, the total latency is calculated in the following manner:

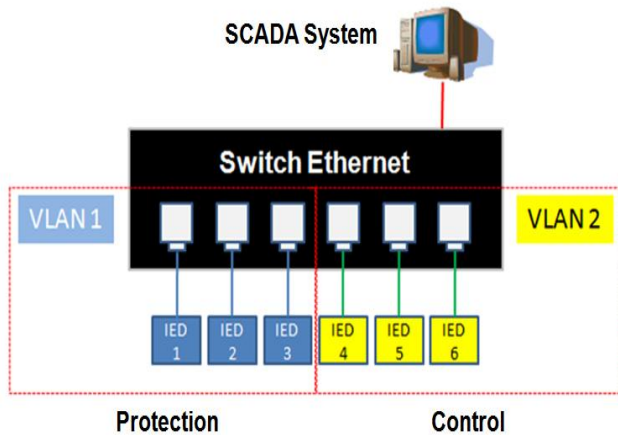
- Leading frame time according to switch processing, normally around 10µs;
- Transmission time for a maximum Ethernet frame size is 122µs;
- GOOSE waiting to be transmitted 24µs;
- 5 GOOSE messages in a row to be transmitted 120µs;

Total time for a complete transmission is **276µs**.

As can be observed, it's possible to assembly and manager a **deterministic** real time network with the 802.1p resource.

c) IEEE 802.1q standard: Virtual LAN (VLAN)

The possibility of segmentation of broadcast domain (picture 7) according to the application purpose can increase the flexibility and contribute to an easier management of the network, as well to reflect in performance gain.



PICTURE 7
VLANs segmenting IED groups

A practical example can be illustrated, a power substation with 230kV and 500kV sections. It's feasible to aggregate the IEDs by control and protection of 500kV section in one VLAN and the 230kV in another.

○ 802.1q adds 12 bits identification for the VLANs in the header of original Ethernet frame with the standards 802.1p e 802.1q implemented:

SYNCH	MAC	MAC	802.1p	802.1Q	LENGTH		
PREAMBLE	DESTINATION	SOURCE	PRIORITY	VLAN	OR	DATA	FCS
	ADDRESS	ADDRESS		ID	TYPE		

PICTURE 8
Ethernet header with VLAN and QoS

d) RSTP – Rapid spanning tree Protocol

The network in a protection and control substation must have a very high performance and very high reliability. Thus is highly stimulated the use of redundant way among the switches to prevent the lost of a link. However, this artifice can be a problem to the enlacement layer.

In conventional switches when ring connected to provide multiple ways to the traffic, may occur loops due to the constant flood of broadcast to find where a host is located. Em switches convencionais, quando ligados em anel para prover múltiplos caminhos para o trafego, pode ocorrer loops devido à constante inundaçao de broadcasts para descobrir onde um host se encontra. As both ways announce the presence (reach) of the host, routing of the messages happens infinitely among the equipment, up to the complete stop of the network.

To avoid this behavior the spanning tree protocol is available. The RSTP is a "fine tuning" of the STP protocol.

The recovery time of the network functions in a STP switch can vary some seconds while in a RSTP switch this time is between 40 to some hundreds of milliseconds.

e) IGMP – Internet group multicast protocol

Basic switching mode treats in very efficient way the unicast frames, but isn't enough to lead the broadcast and multicast traffic.

In a substation automation application, the multicast concept [5] is essential concerning to the necessity of a signal transmission (or a bunch of them) at the same time to many hosts in the network. In this way, instead of sending unicast messages, only one multicast frame is sent to all the switch ports (what is not recommended as may compromise the bandwidth, but occurs in switches without IGMP support) The hosts (in this case the IEDs) are configured to listen to one particular multicast address, not considering the unwanted traffic.

However the excessive use of multicast messages may cause problems in very large networks compromising drastically its performance. A way to avoid this problem is to create dynamically managed domains. This functionality is offered by IGMP.

The IGMP was originally designed to execute jobs on the network layer, but was enhanced to the link layer listening to an IGMP message exchange, auto configuring itself as the multicast packages are subscribed to the evolved communication groups. If the switch listen to an IGMP report type it adds a host port in the IGMP table. If the switch listens to an IGMP leave type, the port is excluded from the table.

Besides of the IGMP power, it's necessary an IP address (network layer) for the protocol to work. It is not possible to apply it directly to the GOOSE or GSSE messages.

From the above exposed, the IGMP functionality can be described as follow:

1. The hosts (IEDs) announce to the medium (switch) its multicast groups addresses which in they are inserted.
2. The routers or switches that are connected to these hosts, made an IGMP table, associating hosts to groups.
3. When a switch receives a frame with multicast address of a determined group, the frame is sent only to the ports where the hosts that belong to that multicast group are. The other ports will not be flood with the multicast which does not contain a host subscribed in the group.

The IGMP also has a query mechanism to make the switches ask to the hosts the ID address of the multicast groups to where they are subscribed. This way, if a host is removed from the switch port and relocated to another, the new host will not receive the multicast frame that was destined to the original host.

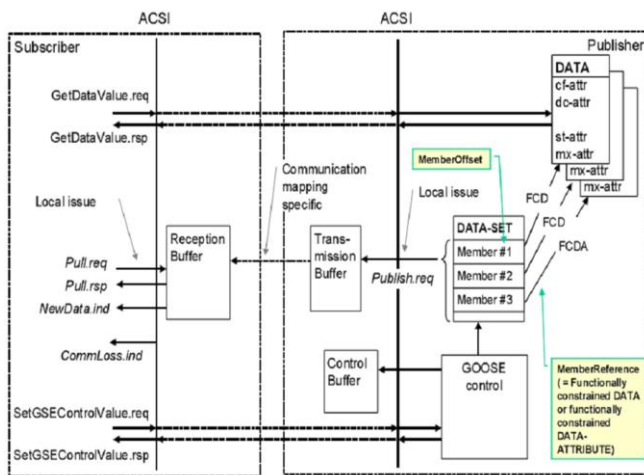
The query messages are typically executed every 125 seconds, but can be manipulated to change this time. When the IGMP is applied to a link layer, it turns into a powerful tool to make GOOSE have a good propagation, as we will show ahead.

GOOSE MESSAGE FUNCTIONAL DESCRIPTION

The GSE messages (Generic Substation Event), model which GOOSE belongs to, and allow the registered values distribution in the inputs and outputs of the process by the network in a fast and reliable way.

The GSE model was constructed under the concept of decentralized and autonomous distribution, that means, any equipment, independently of its location can provide a message delivery simultaneously to more than one host of this network, using multicast.

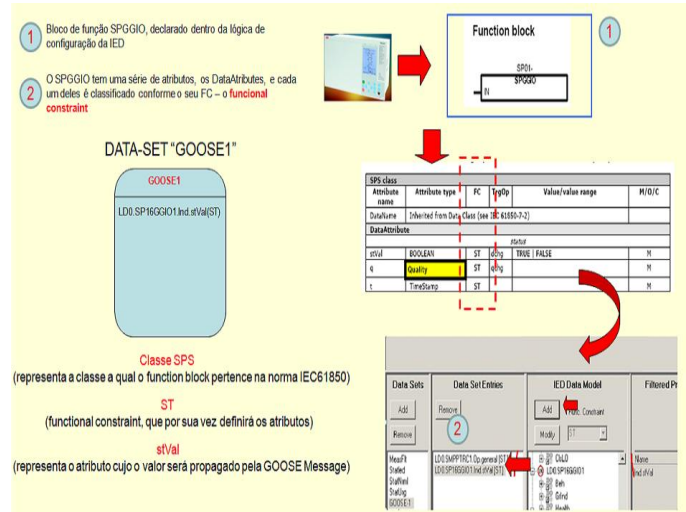
The information exchange is base don the *publisher* and *subscriber* [2] mechanism. The entity described as publisher writes the values in a register (buffer) on the frame sender side. The subscriber entity makes a reading in its respective buffer (located in the frame receptor side). The communication system is responsible for keeping the subscriber buffer actualized, and all the process is controlled by an instance control of GOOSE in the publisher side, as shown in picture 9.



PICTURE 9
Publisher x Subscriber scheme

The process of message exchange is done by specific attribute values declared of the signals that are send and received to or from an IED group. However, for this process to be approached, it's needed to relate the entities evolved in the GOOSE composition.

Initially the signals that will take part in the GOOSE message must be grouped in a DATA SET [6]. The DATA SET is by analogy, a data sheet similar to a data base. Inside the DATA SET are included the *DataAttributes* of an specific function (IEC61850 standard determined as FC – Functional Constraint). This way if any value of a *Data Attribute* of a FC subscribed in the DATA SET is changed in the publisher side, the transmission buffer is actualized by the *publish* service. As a consequence, new values are transmitted with a GOOSE message. Picture 10 shows an example that describes the way the components that will take part of a DATA SET are allocated:



PICTURE 10
Relationship between DATA SET and GOOSE

As defined in the 61850-7-2 document of IEC61850 standard, the data structure of GOOSE messages follow an order established by the GOOSE Control Block as shown in table I.

Table I
GOOSE Control Block

GoCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
GoCBName	ObjectName	GO	-	Instance name of an instance of GoCB
GoCBRef	ObjectReference	GO	-	Path-name of an instance of GoCB
GoEna	BOOLEAN	GO	dchg	Enabled (TRUE) disabled (FALSE)
AppID	VISIBLE STRING65	GO		Attribute that allows a user to assign a system unique identification for the application that is issuing the GOOSE. DEFAULT GoCBRef
DatSet	ObjectReference	GO	dchg	
ConfRev	INT32U	GO	dchg	
NdsCom	BOOLEAN	GO	dchg	
Services				
SendGOOSEMessage				
GetGoReference				
GetGOOSEElementNumber				
GetGoCBValues				
SetGoCBValues				

The attributes of this GOOSE Control Block are responsible to provide the necessary information to the relation publisher x subscriber works correctly. These attributes are described as follows:

- GoCBName:** identification of the only instance of the **GoCB** (GOOSE Control Block) inside the logical node LLN0. LLN0 is as detrimned by IEC61850 standard, the logical node which contains all information inherent to an IED as for example, manufacturer data and operational state of the equipment. Other functions, being them system protection or control have their own logical nodes (LNs). And in cases that the names were not determined by the standard, are created logical devices (LDs), an entity that may embody many logical nodes.



PICTURE 11

GoCB Name declaration inside the logical node LLN0

- **GoCBRef:** Is the location reference of the GoCB inside LLN0. It must be interpreted in the following way:

LDName/LLN0.GoCBName

This way, as in the picture 11 example, the GoCB inside

LD0/LLN0.NewGSEControl

LLN0 is composed as:

- **GoEna:** This attribute inform if the GoCB is qualified (true value) or unqualified (false value) to send GOOSE messages.
- **AppID:** Represents the logical device where the GoCB is located. In this case, as picture 10 example, its value is 0 (as the GoCB is located inside LDO).
- **Dataset:** Is the DATA SET reference that contains the values to be transmitted in the multicast message. Each member of the DATA SET is identified by a number known as MemberOffset and each of these items has an element called MemberReference that corresponds to the data type to be transmitted.
- **ConfRev:** Corresponds to the number of times the DATA SET configuration was modified. These modifications include:
 - delete DATA SET member;
 - reorganize DATA SET;
 - change DATA SET name;
- **NdsCom:** Indicates, when the value is true, that GoCB requires additional configuration to operate correctly.

Moreover the described attributes, theGoCB is also responsible to execute some services to propagate of the GOOSE messages in the network as follows:

- **SendGOOSEMessage:** Is the service designed by GoCB to send multicast messages.
- **GetGoReference:** Used to reference member of a DATA SET specific from the GoCB.

- **GetGOOSEElementNumber:** Returns the member position which the attribute is asked for inside a DATA SET of a GoCB.
- **GetGoCBValues:** Recovers the GoCB attribute values.
- **SetGoCBValues:** Rights values to the GoCB attributes.

GOOSE MESSAGE SINTAXE

The model specified by IEC61850 standard for GOOSE messages defines its structure (the way the message is transmitted) in the way it is shown in the table II:

Table II
GOOSE message architecture

GOOSE message		
Parameter name	Parameter type	Value/value range/explanation
DataSet	ObjectReference	Value from the instance of GoCB
AppID	VISIBLE STRING65	Value from the instance of GoCB
GoCBRef	ObjectReference	Value from the instance of GoCB
T	EntryTime	
StNum	INT32U	
SqNum	INT32U	
Test	BOOLEAN	(TRUE) test (FALSE) no-test
ConfRev	INT32U	Value from the instance of GoCB
NdsCom	BOOLEAN	Value from the instance of GoCB
GOOSEData [1..n]		
Value	(*)	(*) type depends on the common data classes defined in IEC 61850-7-3. The parameter shall be derived from GOOSE control

- **DatSet:** Contains ObjectReference attribute value of the DATA SET (informed by GoCB) which values of its respective members will be transmitted in the frame.
- **AppID:** Contains the identifier of the Logical Device where the GoCB is located
- **GoCBRef:** Contains the GoCB reference (as shown in the GoCB attributes description).
- **T (time):** It's a time stamp that indicates the last change in the DATA SET value and, by consequence when the last GOOSE message was transmitted.
- **StNum (state number):** It's a counter that indicates the number of times a GOOSE message was transmitted.
- **Test:** Signalizes true and false Boolean values. Case is true indicates test mode where the GOOSE messages will not be used for operational purposes.
- **ConfRev:** This parameter contains the number of times the DATA SET configuration was changed. Its value comes from ConfRev GoCB attribute.
- **NdsCom:** The NdsCom attribute of the GOOSE message inherits the value from GoCB.
- **GOOSEData:** Contains the information previously defined in the DATA SET members that will be send by GOOSE.
The **Value** parameter will have the value of each part of the DATA SET referenced in the GoCB.

DIFFERENCES BETWEEN CLIENT / SERVER AND MULTICAST ETHERNET ARCHITECTURE

The main cause of using multicast mechanism for high priority messages sending is the speed factor. However, in the IEC61850 standard, also the client server with reporting communication is used [3] [7] for message exchange which has no severe needs relating to the transmission speed as GOOSE and GSSE have.

This topic has the objective of showing the most relevant differences between client/server with reporting and multicast Ethernet schemes.

a) Client/server with reporting model

The client / server architecture uses generally speaking the seven layers of the OSI model. It has transmission confirmation in the transport layer, function which is given by TCP protocol. By this reason is highly reliable.

Nevertheless, a client / server architecture can not be used by a time critical (real time) application due to its relative high time consumption. Its main characteristic is to be very efficient concerning to communication between the client and the data supplier, but the answers can have typical times up to 1 second, even more depending on the network traffic. This fact by itself compromises the needs of what GOOSE messages are made for.

In this model the server is the supplier of all data. The client by its time has the function of ask to the server the information, send commands to change server behavior and configure in the server conditions that allow the sending of selected information in an automated way (reporting function). This is basically the responsibility of a DATA SET.

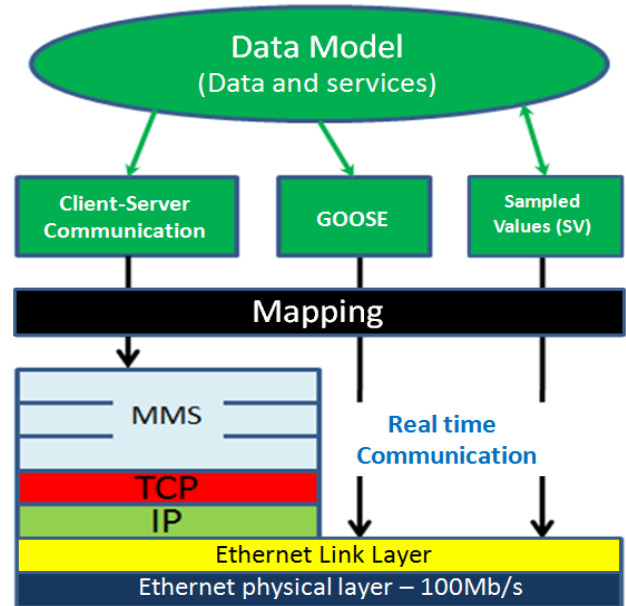
On the contrary to what happens to many protocols designed for SCADA systems that use master / slave scheme, the IEC61850 standard allows multiple links using the client / server model.

b) Multicast Ethernet model

In the multicast Ethernet model don't occur message processing at all communication layers as in the client / server model. Such action would consume a large time to the need of this application where there is no time available to detect a loose information after its sending.

The multicast Ethernet consists in information allocation directly to the link layer, adding only the header of this layer to its multicast reserved address.

Picture 12 represents messages acting in the client / server model and the multicast Ethernet related to communication layers [7] [8] :

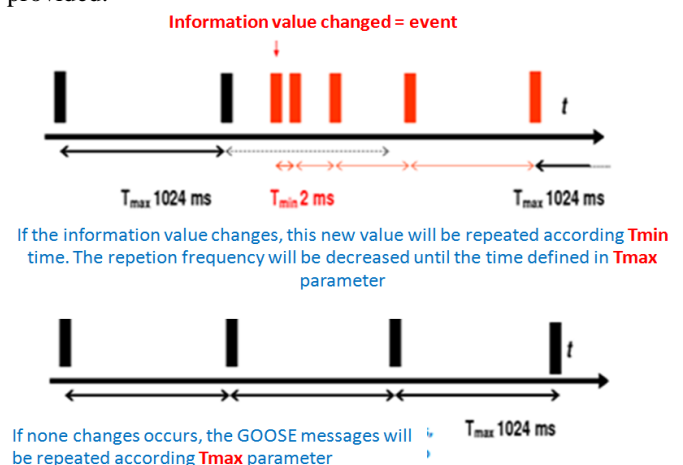


PICTURE 12

Client/server and multicast Ethernet

To by-pass the lack of a mechanism of delivery message confirmation (as the one made by TCP in the transport layer) a GOOSE message is transmitted with a blast (with configurable interval) to assure that IED transmitted data are delivered to their goal. These blasts consist in message repetition around 2ms interval typically.

After successive message transmission, the GOOSE changes its behavior to send data in bigger intervals (analogous to the hell packets of certain routing protocols) defined in its configuration. This behavior stays while there is no change in values defined in its DATA SET. If a change occurs in the members attributes of DATA SET, a new blast is provided.

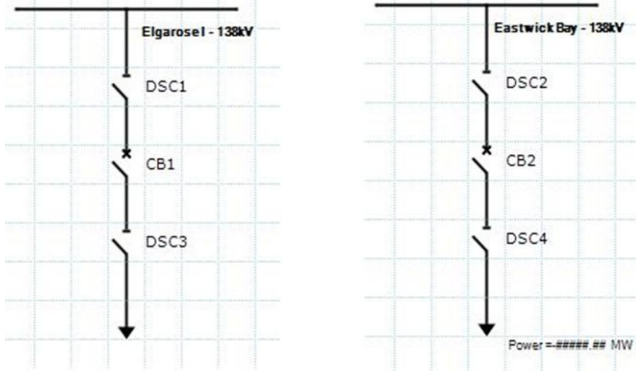


PICTURE 13

GOOSE message transmission scheme

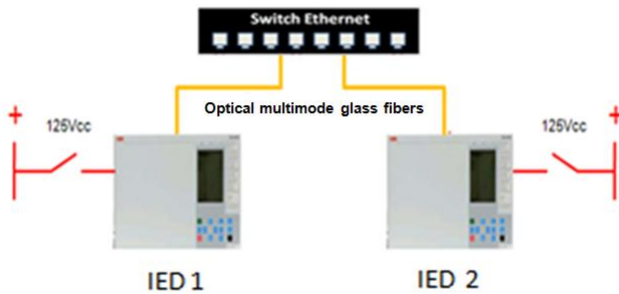
GOOSE MESSAGE TRANSMISSION AND RECEPTION BETWEEN TWO IEDS TESTING

For this test two IEDs were configured to realize the control of two breakers with the tag CB1 and CB2. Located, respectively in the IED1 and IED2. Moreover the breakers, two disconnectors (one for busbar e another for line) were added to each IED, as shown in the synoptic in the equipment.



PICTURE 14
Synoptic of IEDs 1 and 2

The implemented logic was made by the command blocks of the breakers and disconnectors, a logic block that executes a trip order of the breaker obliging it to open immediately when an external signal is received from a digital input, for example. In this application, the signal that sends the opening order of the breaker to the TRIP block will be the logical function responsible by the reception of the GOOSE message provided by an adjacent IED.



PICTURE 15
Test architecture

In picture 15 the contacts represented with 125 Vcc energize a digital input in the IEDs. These inputs are connected logically to the block that processes the sending of the GOOSE message to the other IED.

In the engineering application process, specifically in the GOOSE configuration, it's needed to inform the multicast address to which the GOOSE message will propagate. This way each GoCB of each unity will have an address by which the message can be identified.

To form these addresses, the 6 bytes that compound

the MAC address has to be allocated like this:

- The first three octets design by IEEE, must always be represented by the values 01-0C-CD;
- Fourth octet must assume value 01 when the message is a GOOSE, 02 when the message is a GSSE and 04 when the message is a SV- which is a multicast message containing the analog sampled values (current, voltage measurement etc...);
- The last two octets are freely admitted addresses in the range indicated by the table III as follows:

Table III
Multicast Addresses

Service	Recommended address range assignments	
	Starting address (hexadecimal)	Ending address (hexadecimal)
GOOSE	01-0C-CD-01-00-00	01-0C-CD-01-01-FF
GSSE	01-0C-CD-02-00-00	01-0C-CD-02-01-FF
Multicast sampled values	01-0C-CD-04-00-00	01-0C-CD-04-01-FF

To this test the addresses were 01- 0C-CD-00-01 and 01-0C-CD-00-02 for the IEDs 1 and 2 respectively.

Once the architecture assembled a computer with the *sniffer* Ethernet was connected to the switch with the objective of monitoring the GOOSE message when the digital input connected to the logic block of GOOSE processing change to level 1. This action change the value of one of the members configured in the DATA SET of the message sending it to the network, as can be observed in Picture 16:

No.	Time.	Source	Destination	Protocol	Info
583	2007-02-13 21:16:04.377253	00:80:82:59:16:58	01:0c:cd:01:00:02	IECGOOSE	GOOSE Request
584	2007-02-13 21:16:04.379569	00:80:82:59:16:58	01:0c:cd:01:00:02	IECGOOSE	GOOSE Request
585	2007-02-13 21:16:04.387566	00:80:82:59:16:58	01:0c:cd:01:00:02	IECGOOSE	GOOSE Request
586	2007-02-13 21:16:04.419618	00:80:82:59:16:58	01:0c:cd:01:00:02	IECGOOSE	GOOSE Request

PICTURE 16
GOOSE message send monitoring

The *sniffer* Ethernet used to demonstrate the GOOSE message sending received a plug-in to correctly show the frames and packages of IEC 61850.

In Picture 16 may be noticed a blast of GOOSE messages sent by IED2 (send multicast address 01-0C-CD-00-02). The messages were configured with the parameters Tmin of 2ms and Tmax of 10s, initially are transmitted with intervals equal to Tmin and this intervals increase successively up to Tmax. It's possible to observe in picture 17 that the time between the first and second transmission is approximately 2ms.

No	Time	Source	Destination
583	2007-02-13 21:16:04.377253	00:80:82:59:16:58	01:0c:cd:01:00:02
584	2007-02-13 21:16:04.379569	00:80:82:59:16:58	01:0c:cd:01:00:02

```

... 0000 0000 0001 = ID: 1
Type: IEC 61850/GOOSE (0x88b8)
IEC 61850 GOOSE
AppID*: 0
PDU Length*: 160
Reserved1*: 0x0000
Reserved2*: 0x0000
PDU
{
  IEC GOOSE
  {
    Control Block Reference*: REC670_T2LD0/LLN0$GO$NewSEControl
    Time Allowed to Live (msec): 11000
    DataSetReference*: REC670_T2LD0/LLN0$GOOSE-2
    GOOSEID*: REC670_T2LD0/LLN0$GO$NewSEControl
    Event Timestamp: 2006-11-12 04:47:36,327000 TimeQuality: 6a
    StateNumber*: 9
    Sequence Number: 0
    Test*: FALSE
    Config Revision*: 1
    Needs Commissioning*: FALSE
    Number Dataset Entries: 4
    Data
    {
      BOOLEAN: FALSE
      BOOLEAN: TRUE
      BITSTRING:
      BITS 0000 - 0015: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      BITSTRING:
      BITS 0000 - 0015: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    }
  }
}

```

PICTURE 17
GOOSE message PDU

The GOOSE parameters can be visualized via message PDU capture, as well the values that are transmitted.

FUTURE PERSPECTIVES

Although the IEC61850 standard already defines the multicast Ethernet structure for transmission of analog measurement – the SV, this function represents a challenge due to a very fine synchronism needed for the phasors of values like voltage and current.

This synchronism must be around one microsecond. The mechanisms to feasible this accuracy must use the pattern IEEE 1588 (Standard for a Precision Clock Synchronization for Networked Measurement and Control Systems).

The SV is similar to GOOSE concerning to its functionality. However there is a relevant difference: on contrary of GOOSE some samples of the SV data can be lost without compromising completely its functionality. For the GOOSE this would be fatal.

An important using of SV could be some automatism among IEDs where is needed sending data referring to the measurement of the electrical system for decision take, for example, to shed a load or to allow a circuit closing. An application where this functionality can contribute is the ECS (emergency control scheme) used in some Brazilian power utilities.

CONCLUSION

This article shows an alternative to the use of multicast Ethernet as a technology of real time messages delivery in an substation automation application.

Due to the speed and robustness of the GOOSE mechanism, they can be used in the many control schemes (interlocking or automatism activation proposed by an Project logic) or protection (trip signal between IEDs) contributing to reduce the wiring in the substations, as this functions before made by electrical connections (copper wire)

may now be made by Ethernet network.

Another relevant factor is that Ethernet is largely used, this way the techniques developed as enhancement of this technology made the multicast use for real time possible, and may be modified and perfected to make this kind of application still more efficient in a near future.

REFERENCES

- [1] Skendzic, Veselin, "Enhancing Power System Automation", *Schweitzer Engineering Laboratories USA*
- [2] "IEC61850 Standards", *Primeira edição*, Agosto - 2003
- [3] Brand, Klaus-Peter, "IEC61850 Short Tutorial", CIGRÉ Paris 2004
- [4] Pozzuoli, Marzio, "Ethernet in Substation Automation Applications – Issues and Requirements", *Western Power Delivery Automation conference*, Abril – 2003
- [5] "Media Access Control", www.firewall.cx/mac_addresses, último acesso em 14/05/2007
- [6] GE, "IEC61850 Presentation", Julho -2005
- [7] Brand, Klaus-Peter, "Communication Architectures with IEC61850", *CIGRÉ Brasil*, 2006
- [8] Baigent, Drew, "IEC61850 Communication Networks and Systems in Substations", *SIPSEP 2004 México*, Novembro – 2004

BIOGRAFY

Julio Cezar de Oliveira was Born at Guarulhos, SP, Brazil in 1978. Is graduated in System analysis by the Guarulhos University and is doing a post graduation course in Computer Network Engineering by UNICSUL University. Works on ABB Substation Automation area since 1994, Participating of elaboration and testing of system in many power utilities in Brazil. In 2005 worked with the software development and IED integration at ABB Sweden.

Walter Augusto Varella, Born in Santos, SP, Brazil, 1959. Graduated as Electronic Engineer in 1982 by FEI - Faculdade de Engenharia Industrial. In 2003 got the master degree in Electronics Engineering by POLI/USP - Universidade de São Paulo. Actually doing a PhD in Electronics Engineering and Computer by ITA - Instituto Tecnológico de Aeronáutica. Since 1992 is a teacher for CEFET Technical School and since 2006 is a teacher and professor for Computer Network do Course of Unicsul University and professor for UNIBAN Bandeirantes University.

Antonio Eduardo Marques da Silva Born in São Paulo, SP, Brazil, 1971. In 2000 got a degree of Mechanical Engineer (system automation) by UPM UPM - Universidade Presbiteriana Mackenzie. In 2006 got the degree of Especialista em Educação do Ensino Superior pela PUC/MG. Made some extension courses at POLI/USP - Universidade de São Paulo e INPE – Instituto

Nacional de Pesquisas Espaciais. Actually doing his Master in Eletronic Engineering at ITA - Instituto Tecnológico de Aeronáutica. Since 2003 he is teacher and coordinator of computers network group of UNICSUL - Universidade Cruzeiro do Sul and coordinator of post graduation course of the same university. He's also na academy instructor and manager of CTT Brasil - Cisco Learning Solution Partner, obtained several certifications regarding data communication and network infrastructure.

Gustavo Forster, Born in Pouso Alegre MG Brazil in 1956. In 1980 got a degree in Eletrotechnics by Getulio Vargas school. In 2004 got degree in Information Administration Business by Faculdade Radial. Works in Protection, Control in Substation Automation for the last 27 years.