The Real-Time Publisher/Subscriber Communication Model for Distributed Substation Systems

Cagil R. Ozansoy, Aladin Zayegh, and Akhtar Kalam

Abstract—**In the past decade, new communication schemes have been designed and retrofitted into substations by utilities to integrate data from relays and Intelligent Electronic Devices (IEDs) and capitalize on the protection, control, metering, fault recording and communication functions available in digital devices. The introduction of IEC 61850 has made it possible and justifiable to integrate station IEDs on a high-speed peer-to-peer communication network (ethernet) through standardization. However, more advances are needed in order to establish an open and standard working environment allowing for more functions to be developed. In this paper, the authors propose a real-time publisher/subscriber communication model as a means of satisfying the unique behavior and communications needs of the IEC 61850 protocol. The authors provide a detailed design and implementation detail of this model along with interesting performance results. The target audience for this paper includes power system protection and automation engineers and technicians as well as research personnel who have at least a basic understanding of the IEC 61850 international standard and other technology mechanisms addressed in this paper.**

Index Terms—**Abstract Communication Services Interface (ACSI), communication systems, IEC 61850, IED, mapping, middleware, protocol, publisher/subscriber, relays, standardization, substation.**

I. INTRODUCTION

Communication systems have been used for decades to enhance the performance of power systems. Power providers are focused on increasing productivity and making electric power safer, more reliable and more economical by providing innovative, simple to use, robust technologies for power system protection, automation, control and monitoring. Development of appropriate communication technologies and protocols is at the heart of this strategy [1], [2].

For the last few years, the advancements in microprocessor-based IEDs networked over high-speed communication networks using standardized communication protocols is leading the evolution of power system control technology. The use of existing communication standards and commonly accepted communication principles jointly with the new standards such as IEC 61850 provides a solid base for interoperability between IEDs in a substation leading to more flexible and powerful protection and control systems [3].

Even though IEC 61850 allows discrete devices to share data and services, it is only an abstract application layer protocol outlining two main groups of communication models in [4], [5], the client/server and publisher/subscriber models, that provide mechanisms for sending and receiving data. However, the services that use these models have only been abstractly modelled. Abstract means that the standard focuses on describing what the services are indented to provide rather than how they are build. Hence, they can only be usable when mapped to specific communication services, such as the manufacturing message specification (MMS) and ISO/IEC 8802-3, as described in IEC 61850-8-1 [6]. Although not addressed in IEC 61850, it is possible to implement the standard's object models and services by mapping to other different communication stacks such as the distributed component object model (DCOM) or common object request broker architecture (CORBA).

The first stage of the IEC 61850 related research carried out by the authors involved the transformation of the IEC 61850 into a solid protocol by the implementation of its services and information models as concrete programs making use of the powerful techniques of object-oriented-programming (OOP). The work carried out describes how the information models and services can be build based on their IEC 61850 descriptions. It intends to provide an alternative to the current adopted approach that is the implementation of the standard's models and services by making use of the existing models and services of an underlying communication service simply called the mapping process. This mapping process can easily get extremely complex and hard and in some cases, there is always the possibility that "some ACSI services may not be supported directly in all mappings but equivalent services shall be provided" [4]. The proposed research provides a standard universal implementation of the IEC 61850 standard removing its dependency on any specific communication service such as the MMS.

The second stage of the research undertaken, which is to be discussed in detail in this paper, involved the design and implementation of a standard universal data delivery service middleware, designed specifically to cater for the special communication needs of the IEC 61850 protocol such as the need to support

1) client/Server communication model;
2) publisher/subscriber communication model;
3) fine grained time synchronization [7];
4) ability to tradeoff delivery reliability against delivery delay [8];
5) ability to identify differing quality of service (QoS) requirements of the different message types supported by the application layer.

Manuscript received March 23, 2005; revised December 1, 2005. Paper no. TPWRD-00153–2005.

C. R. Ozansoy and A. Zayegh are with the School of Electrical Engineering, Victoria University, Melbourne 3047, Australia (e-mail: cagil.ozansoy@research.vu.edu.au).

A. Kalam is with the Faculty of Science, Engineering and Technology, Victoria University, Melbourne 3047, Australia.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRD.2007.893939

Middleware software is a layer between the networking and application code provided by the communication processor. The function of the middleware is to insulate the application programmer from the raw networking code thus providing an easier way to communicate. The required data delivery service middleware needs to cope well with the abstract communication services interface (ACSI) object and service models as well as supporting different message types specified in the protocol.

The designed middleware needs to be small and fast adding only minimal overhead to the underlying network communication stack. Moreover, it should be much more efficient than MMS or CORBA. Although MMS preserves many technical advantages, it has not been completely successful. Main criticism to the MMS architecture includes the complexity, poor performance and lack of any explicit support for publisher/subscriber architectures. The latter item explains the reason why a second communication stack, the ISO/IEC 8802-3, had to be proposed in [6] for the mappings of one-to-many communication capability requiring models such as the GOOSE.

Clearly, there are many challenges such as aforesaid that have to be solved. However, the need to design a real-time communication model to run with a communication processor is evident. The term "real-time" means that an application should respond to events within a prescribed range of time even under failure and extreme load conditions. Overall, the formal communication model (middleware) has to support the following features:

- client/Server communication mechanisms;
- publisher/subscriber communication mechanism;
- modeling time and time-stamping each transaction [8];
- allowing the application software to tradeoff timing against reliable delivery [8];
- allowing for a adaptive synchronization scheme [8];
- working in a real-time communication processor environment [8];
- allowing the application software to distinguish between different message types and their corresponding QoS requirements.

This paper mainly focuses on the publisher/subscriber communication component of the designed middleware architecture. A publisher/subscriber system, shown in Fig. 1, is a middleware communication service supporting an asynchronous style of many-to-many communication in contrast to the Request/Response type of synchronous approach of object invocation. It relies on the preferences expressed by the subscribers to deliver messages from one publisher to one or many subscribers instead of the publisher relying on specific destination addresses [9]. A publisher can also be referred as a producer or sender. Similarly, subscribers are most often referred to as consumers or receivers.

II. IEC 61850 Standard

IEC 61850 is an abstract application layer protocol aimed at providing interoperability between a variety of substation and feeder devices. IEC 61850 is based on the need and opportunity for developing standard communication protocols to permit interoperability of IEDs from different manufacturers. IEC 61850 makes use of existing standards and commonly accepted communication principles, which allows for the free exchange of information between IEDs. It considers the operational requirements in view of the fact that any communication standard must consider the substation operational functions. However, the communication standard IEC 61850 focuses on neither standardizing the functions involved in substation operation nor their allocation within the substation automation systems. It identifies and describes the operational functions used to define the impact of the operational functions on the communication protocol requirements [3].

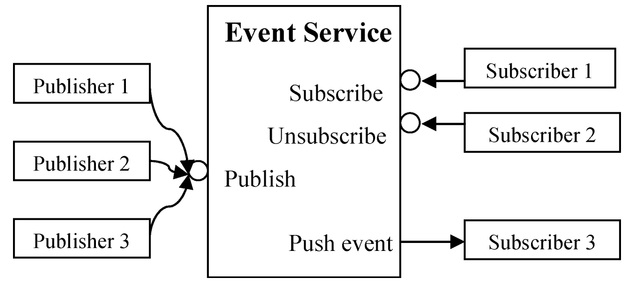

Fig. 1. Publisher/subscriber communication model.

IEC 61850 application, device and communication views are discussed in detail in the following subsections explaining the process of standardization within substation automation systems.

A. ACSI

The IEC 61850-7 ACSI models are abstract definitions of common utility communication functions in field devices mainly describing communication between clients and remote servers. It aims to ensure that the semantics of the data exchanged are understood by all field devices provided that standardized mappings of these abstract services to the underlying communication protocol are defined [4], [5]. Accordingly, ACSI defines

- substation-specific information models such as
  - DATA;
  - SERVER;
  - LOGICAL–DEVICE;
  - LOGICAL–NODE.
- substation-specific information exchange service models such as
  - setting group control block (SGCB);
  - generic substation event control block (GSECB);
  - sampled values control block (SVCB).

ACSI specifies the basic layout for the information models and information exchange service models. Nevertheless, the implementation of objects and issues related to modeling are left to the user. The representation of a physical object can be referred to as an object model. For instance, the measurements of voltage, current and power in a relay can easily be grouped together to form the "measurement model." Why do we need to use object-oriented models? The primary answer to this question is the fact that object modeling standardizes information exchange between models and devices by creating an independent representation of the data not linked to physical location or storage medium inside the physical device.

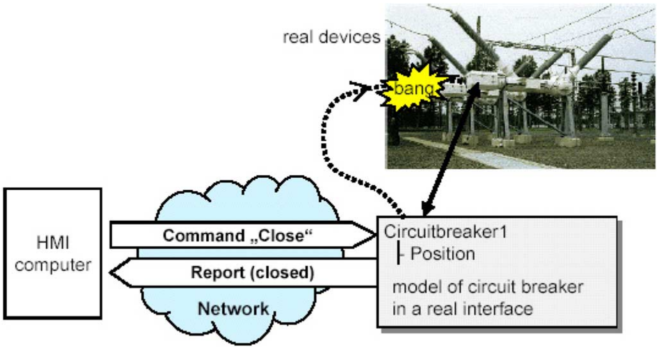

Fig. 2. Substation application view example [4].

LOGICAL_NODE
+ LNName: CosNaming::NameComponent + LNRef: char [1..255] ([255]) + Data: DATA* + DataSet: DATA_SET* + BufferedReportControlBlock: BRCB_Class* + UnbufferedReportControlBlock: URCB_Class* + Log ControlBlock: LCB_Class*
+ GetLogicalNodeDirectory() : void + GetAllDatavalues() : void**

Fig. 3. UML class notation for Logical_Node class.

Once standardized, it is possible to request some information from devices without having to know any information about the manufacturer of the device. On the other hand, additional manufacturer implementation specific information stored in extended data objects can only be retrieved on condition that the node requesting the information knows about the details of the manufacturer. Object-oriented modeling (OOM) and OOP techniques can easily be used for modeling and implementing ACSI models alongside services. However, it should also be noted that some vendor specific objects within the IED will be left unstandardized and will take some part of the total object space [4], [5].

B. IEC 61850 Application View

A simple example of an interoperable function within a substation is to switch a circuit breaker via a computer. Such a case is depicted in Fig. 2. The task of the human–machine interface (HMI) in this example is to send control commands to an IED, which implements the tasks of a circuit breaker, requesting the IED to switch the position of the breaker [4].

Once the request has been processed by changing the position of the breaker, the IED may send a reply signal back to the HMI computer indicating the new position of the breaker. In addition to sending control commands, the HMI might also query about the information content of the IED, which causes IED to forward data about its information contents such as the nameplate and ratings. To be able to successfully send its command and receive replies, the HMI computer needs to know [4]:

- name of the circuit breaker implemented in the IED;
- how to express its request of changing the position of the breaker;
- how to read reply data.

From this application point of view, IEC 61850 aims to assist substation devices and their communications amongst them by

- standardizing abbreviated names for substation functions and equipment;
- by naming and describing functions and information;
- by describing how to access functions and how to exchange information.

Therefore, IEC 61850 identifies all the known functions in a substation automation system and splits them into subfunctions or so called logical nodes. A logical node (LN) is a subfunction located in a physical node, which exchanges data with other separate logical entities.

IEC 61850 defines 13 different groupings of data each subdivided further into LNs. All LNs have been grouped according to their most common application area, a short textual description of the functionality, a device function number if applicable and the relationship between logical nodes and functions. IEC 61850 decouples applications to design them independent from communication so that they will be able to communicate through the use of different communication protocols. Hence, LNs are simply functional models of real devices [4], [5], and [10]. Each LN can be thought of as an object with attributes and methods. Fig. 3 shows the Logical_Node Class diagram, which is simply a unified modeling language (UML) template for the creation of LN objects.

The Logical_Node class is a composition of a number of attributes that describe the characteristics of LN objects. These attributes not only include data that contain the information required by a specific function but also various control blocks, data sets, and others as shown in Fig. 3. There are two specific infrastructure LNs defined in IEC 61850-7, which are the physical device logical node (LPHD) and the logical node zero (LLNO). LPHD is used for accessing hardware related data of an IED whereas LLNO is used for accessing Logical Device (LD) related data of an IED. In addition to inheriting all the attributes and operations of the Logical_Node class, LLNO can also include:

- Setting–group–control–block (SGCB);
- Log;
- GOOSE–control–blocks (GoCBs);
- GSSE-Control-Blocks (GsCBs);
- multicast–sampled–value–control–blocks (MSVCBs);
- unicast–sampled–value–control–blocks (USVCBs).

In accordance with Fig. 3, Fig. 4 shows the basic building blocks of the circuit switch (XSWI) LN object from the "Switchgear Logical Node" group, which can be used for modeling switches without short-circuit breaking capability.

Logical_Node classes are specialization of common data classes (CDCs) [11], which are entities specifying class structures by including one or more DataAttributes. Each CDC is a subclass of the DATA class, which is one of the most important building blocks of the IEC 61850 representing meaningful and domain specific information located in field devices within a substation automation system. The various building blocks of the DATA class are illustrated in Fig. 5.

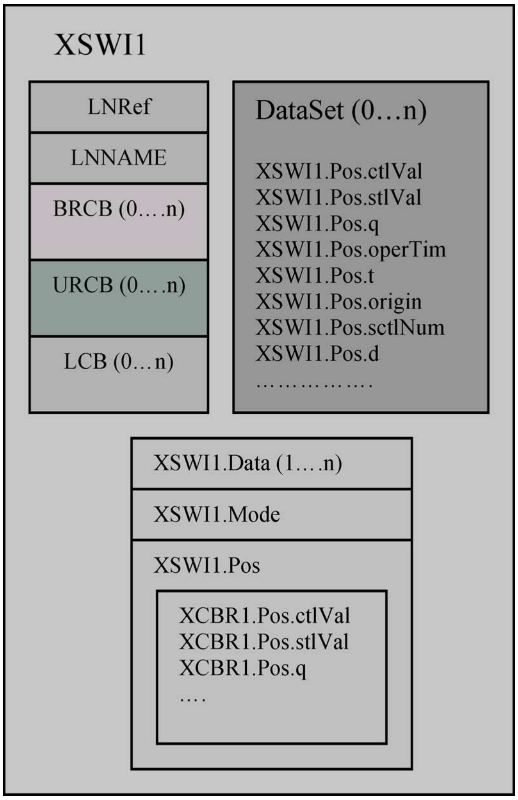

Fig. 4. Building blocks of the LN class model.

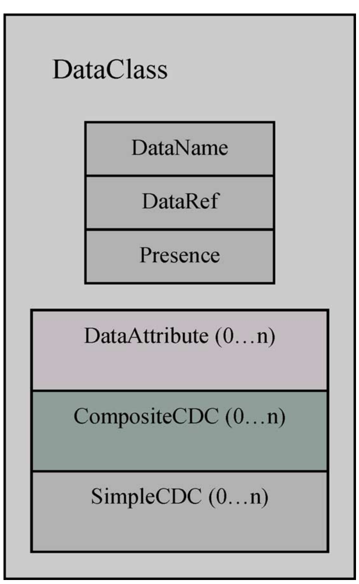

Fig. 5. Building blocks of the DATA class model.

Other than the DataName, DataRef and the Presence attributes, each instance of a DATA class is a composition of SimpleCDCs, CompositeCDCs and DataAttributes. Each DATA object may be a composition of zero or more instances of CompositeCDCs, SimpleCDCs and DataAttributes. However, it must contain at least one of the elements listed above [4], [5], [10]. CompositeCDCs are also of type DATA class thus making the structure of the DATA class recursive [5]. SimpleCDCs are subclasses of the DATA class since they are of type CDC [11]. Table I shows the list of CDCs that make up the XSWI LN object.

TABLE I
LIST OF THE COMMON DATA CLASSES MAKING UP THE CIRCUIT SWITCH (XSWI) LOGICAL NODE OBJECT

DataClass	DataName	CDC
Mode	Mode	ISC
Behavior	Beh	ISI
Health	Health	ISI
Name plate	Name	PLATE
Local operation	Loc	SPS
External equipment health	EEHealth	ISI
External equipment name plate	EEName	PLATE
Operation counter	OperCnt	ISI
Switch position	Pos	DPC
Block opening	BlkOpen	SPC
Block closing	BlkClos	SPC
Charger motor enabled	ChMotEna	SPC
Switch type	SwTyp	ISI
Switch operating capability	SwOpCap	ISI

TABLE II
LIST OF ATTRIBUTES FORMING THE CONTROLLABLE DOUBLE POINT CDC

AttributeName	Type	Value
ctlVal	BOOLEAN	FALSE | TRUE
stVal	ENUMERATED	0|1|2|3
pulseConfig	PulseConfig	
operTim	TimeStamp	
q	Quality	
t	TimeStamp	
origin	Originator	
ctlNum	INTEGER	0…255
d	Description	Text
ctlModel	ControlModel	
sboTimeout	INTEGER	
sboClass	ENUMERATED	Operate once | Operate many
tag	Tag	

Each CDC is composed of a number of attributes each having its own Name, Type, Value, etc. For example, the switch position DataClass is a specialization of the controllable double point (DPC) CDC. Table II shows the list of the attributes that compose the DPC CDC [11].

In addition to being composed of a number of attributes, all the class models described above (Logical_Node, Common Data and DATA) also allow a number of services to be performed by the objects created with the use of these class models. Classes not only define attributes (state information) but also services of the collection of objects having same characteristics. For example, the Logical_Node class model shown in Fig. 3 supports two services as follows.

1) GetLogicalNodeDirectory: Clients use this service to get the ObjectReferences of all the instances contained within a LN. The input/output arguments for this function are shown in Table III;
2) GetAllDataValues: Clients use this service to get all the DataAttributes of all the data contained within a LN. The input/output arguments for this function are shown in Table IV.

TABLE III
ARGUMENT TABLE FOR GETLOGICALNODEDIRECTORY SERVICE

Input Arguments	Output Arguments (successful)	Output Arguments (unsuccessful)
LNReference	InstanceName[0..n]	Response-
ACSIClass	Response+	

TABLE IV
ARGUMENT TABLE FOR GETALLDATAVALUES SERVICE

Input Arguments	Output Arguments (successful)	Output Arguments (unsuccessful)
LNReference	LNReference	Response-
Functional Constraint	DataAttributeReference	
	DataAttributeValue	

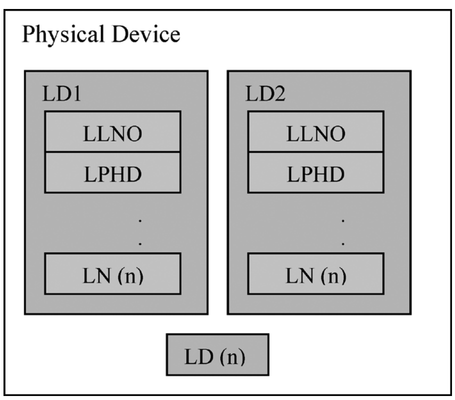

Fig. 6. Logical device building blocks.

C. IEC 61850 Device View

So far, the authors have described LNs and data that represent information related to real application functions within substations. However, these are not by themselves sufficient to express all the necessary details and issues concerning substations [4], [5]. This resulted in the need for further components to be defined and modeled. For example, there was a clear need for a specific component to represent information about the resources of the host itself including the real equipment connected to the host device as well as the common communication aspects applicable to a number of LNs. Hence, the Logical_Device class model was introduced with the intention of filling these missing gaps [4], [5].

Each LD, which is mainly a composition of LNs, can be defined as a "virtual device that exists to enable aggregation of related LNs and data sets". As shown in Fig. 6, each LD must be composed of a single LLNO (logical node zero), a single LPHD (logical node physical device) and at least one other LN. The LLNO contains data related to LD whereas the LPHD includes common data related to the physical device excluding any data from the common Logical_Node class model [5], [10]. The physical device shown in Fig. 6 is usually defined and modelled as a server within the IEC 61850 perspective. The server, containing all the communications visible and accessible models, represents the communications visible behavior of an IED usually modelled with a single LD.

D. ACSI Service Models

As discussed in Section II-B, ACSI comprises a number of service models. The two most important of these to be also discussed in detail in this section are the

1) GOOSEControlBlock (GoCB);
2) MulticastSampledValueControlBlock (MSVCB).

The GOOSEControlBlock is mainly engaged in controlling and regulating the exchange of Generic Object Oriented Substation Event (GOOSE) messages. GOOSE provides fast and reliable exchange of values of a collection of data attributes organized by a data set in a simultaneous fashion to more than one physical device through the use of multicast/broadcast services [5], [6].

Each GOOSE message, having the format shown in Fig. 7, expresses all required protection scheme information regarding an individual protection IED. The status of the functional elements in the IED is reported in the form of a state machine. IEDs are responsible for capturing the effects of abnormal system conditions within substations. Once they capture the effects, they express the details in the form of a GOOSE message.

The power quality monitoring and recording devices are the type of devices that are usually interested in such GOOSE messages. However, in order to receive such messages, they need to have a mechanism for subscribing to the publishing devices. Once they enrol themselves in a particular physical device, they will be added to the publishing device's subscriber list. The transmission of a GOOSE message will be initiated by changes in the values of one or more data attributes referenced by a data set.

The generic object-oriented substation event (GOOSE) replaces the mechanism of exchanging control signals between IEDs using a fixed, hardwired and sequential data acquisition infrastructure, a mechanism incapable of meeting the requirements of real time substation communication systems. These messages are time-critical and mission-sensitive needing high reliability. Subscribing IEDs receiving the GOOSE messages use the information contained within these messages to decide on the suitable protection responses to be taken in response to a particular state change described by the GOOSE message.

The fact that protection responses rely on the successful and timely exchange of GOOSE messages requires very strict performance requirements, such as the need for a worst case 4 ms application-to-application delivery delay requirement [12]. This implies that the total delivery delay between the communicating devices, which not just includes the delay on the wire but also the delay message encounters while travelling through the protocol stack [13] from the application layer all the way to the hardware, should not exceed 4 ms.

IEC 61850 standard specifies a real-time publisher/subscriber communication model as illustrated in Fig. 8 as a solution for meeting the 4 ms total delivery delay requirement [4], [5].

The MSVCB is conversely responsible for controlling and regulating the exchange of sampled values over the network. MSVCB [5] provides organized and time controlled exchange of values of a collection of data attributes organized by a data

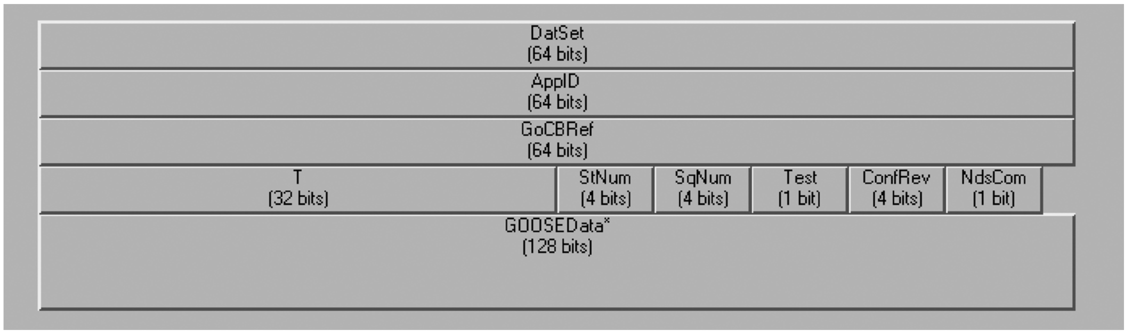

Fig. 7. GOOSE message format.

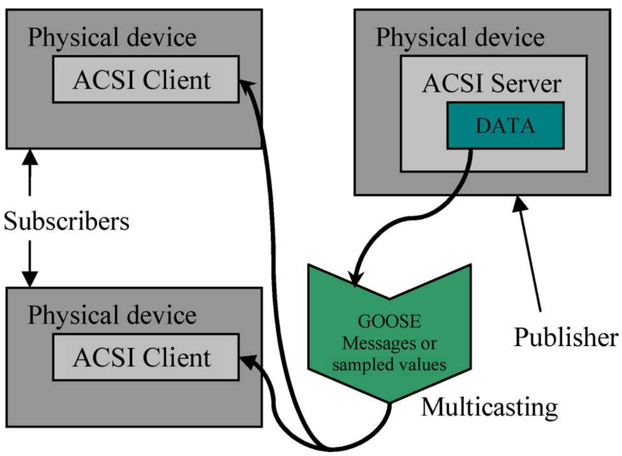

Fig. 8. Publisher/subscriber communication mode.

set minimizing the combined jitter of sampling and transmissions to a degree where an unambiguous allocation of the samples, times and sequence is possible. Similar to the GOOSE, the information exchange is also based on a publisher/subscriber mechanism depicted in Fig. 8.

E. Communication Mechanisms for ACSI

There are four main communication mechanisms used in ACSI. These are [4]

- request/response;
- request/no response;
- GOOSE message;
- sampled value.

Services such as GetDataSetValues and SetDataSetValues make use of client/server communication mechanisms, where the communication between the requesting client and the replying server exhibits a synchronous type of messaging with the client being blocked once it makes the request until the corresponding reply arrives.

Such a communication architecture works very well when clients need to issue service requests expecting confirmations of the service that has been processed in the server. However, as it is in the case of GOOSE messages, the client/server architecture is not useful when nodes within the communication system need to talk to many other nodes simultaneously. Within distributed systems, there is a need for a communication pattern, which will serve for asynchronous, loosely coupled and one-to-many communication scheme. publisher/subscriber architectures are well suited to serve such needs.

III. Real-Time Publisher/Subscriber Model

This section discusses the design constraints behind the design of a suitable real-time publisher/subscriber middleware for substation communication systems. Design concerns are described for developing a communication system responsible for distributing mission and time critical information within substations to legitimate parties in a timely, reliable and accurate manner. A number of issues need to be considered concurrently when building appropriate publisher/subscriber communication models. These issues involve the choice of a suitable architecture and a number of communication techniques. Some of the specific requirements of real time substation applications such as the need for a time synchronization mechanism also need to be discussed and solutions proposed. The first subsection discusses the routing problem.

A. Routing Problem

The main approaches for solving the routing problem are

1) sending a number of points to point messages;
2) sending a multicast message;
3) sending a broadcast message.

Within the IEC 61850 framework, GOOSE messages and messages transmitting sampled values (SV) are the main types of messages that require indirect asynchronous delivery. The IEC 61850 protocol has specified the use of the multicast alternative, shown in Fig. 9, for the indirect delivery of GOOSE state change messages. It is possible to multicast GOOSE messages due to the fact that they need to be repeated a number of times until their time-to-live expires to achieve high reliability and they do not need to be acknowledged.

Multicast messaging allows the sender to send a single copy to the data stream, which will then be replicated and forwarded to consumers that have previously signalled their interest. Therefore, instead of sending thousands of copies, the sender sends a single copy directed by routers on the network to consumers that have indicated their interest in the message. Consumers usually indicate their interest by joining a particular multicast session group.

Thus, multicast messaging reduces the amount of traffic over the network yielding an increased efficiency for both the sender and network with a number of other performance improvements

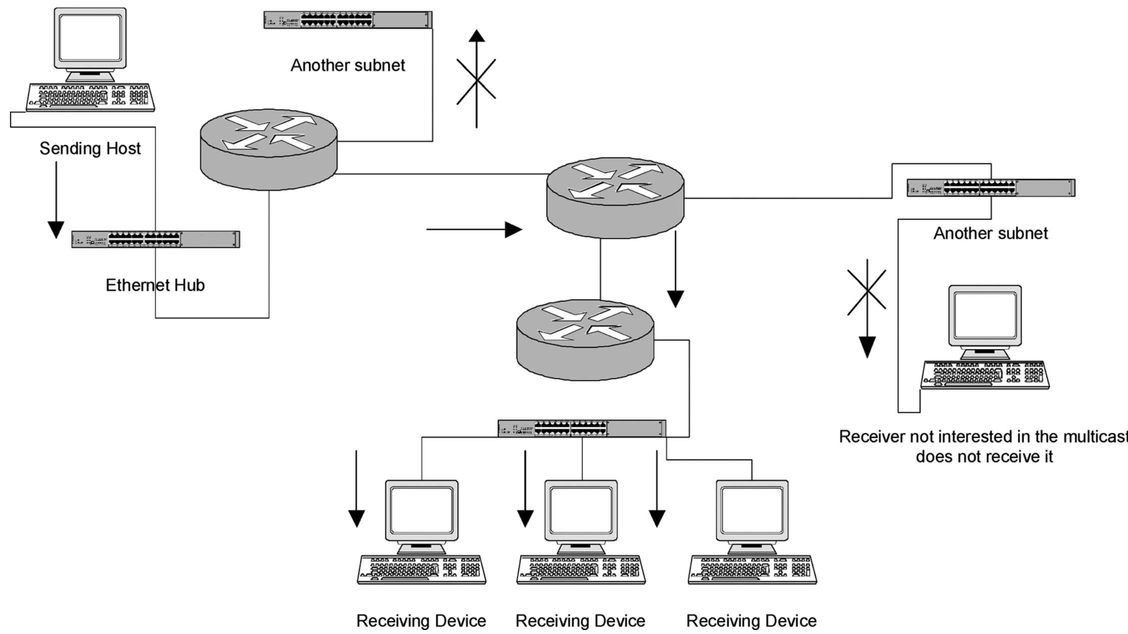

Fig. 9. Multicast transmission.

as well. Multicast oriented communication enables nodes to join or leave groups as a local activity unambiguously creating group membership and group wide awareness. In this research, the decision has also been to stick with the multicast alternative since it provides numerous advantages such as reduced traffic over the network and minimized delivery delays.

B. Subscription Mechanism

In IEC 61850 ACSI, GOOSE messages are proposed as a means of expressing all required protection scheme information of an individual protection IED. The status of the functional elements in the IED is reported in the form of a state machine. Once IEDs capture the effects of abnormal system conditions within a substation, they express the details in the form of a GOOSE message. The power quality monitoring and recording devices are the type of devices that are usually interested in receiving such GOOSE messages [10]. However, in order to receive such messages, they need to have a mechanism for subscribing to the publishing devices. This suggests that each subscribing device needs to be aware of its publishers and their relative IP group addresses. Fig. 10 shows a distribution feeder protection relay feeder PIED publishing to a subscribing device, which is a power quality, monitoring and recording (PQMR) IED. When the feeder detects a fault, it will trigger the operation of the power quality monitoring and recording IED by sending a multicast message to various destinations including the PQMR IED.

Reference [8] discusses that each publishing node must not only be aware of its own subscribers but also a complete list of publishers each one of its subscribers subscribes to. Similarly, [8] also states that in the case of a subscribing node, each subscriber needs to be aware of its own publishers and a complete list of the subscribers each one of its publishers publishes to.

The designed real-time publisher/subscriber model associates a logical handle to each publishing device. The logical handle

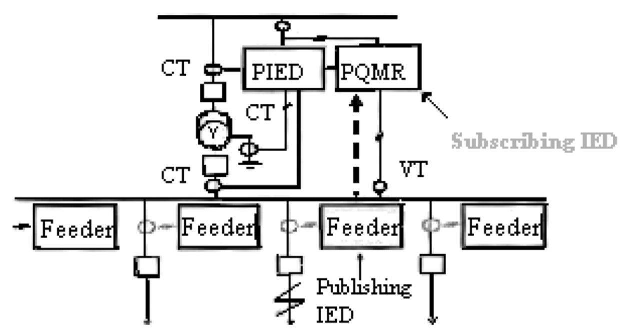

Fig. 10. Feeder IED publishing to the subscribing IED [10].

can simply be a variable length ASCII string containing the address and name of each publishing device and names of its subscribers. Each subscriber interested in receiving GOOSE messages from a particular publisher will need to register itself in the subscriber list of the publishing device with a subscription message containing its name, address and the names of the other devices it has subscribed to. Thus, each publisher will have a list of its own subscribers and the other publishers they have subscribed to. Each published GOOSE message will be tagged with the logical handle information of its publisher. Subscribers upon receiving GOOSE messages will be able to filter the logical handle to acquire the name and address of the publisher and the list of subscribers it publishes to. The subscribers will then be able to keep a record of not just their publishers but also a list of the other subscribers of each particular publisher.

C. Binding and Filtering

The problem of binding can be overcome very easily by using the publisher-based subscription mechanism. Publisher-based subscription mechanism requires subscribers to subscribe to publishers providing their details such as node name, node address and protocol etc. Once a subscription is processed, the

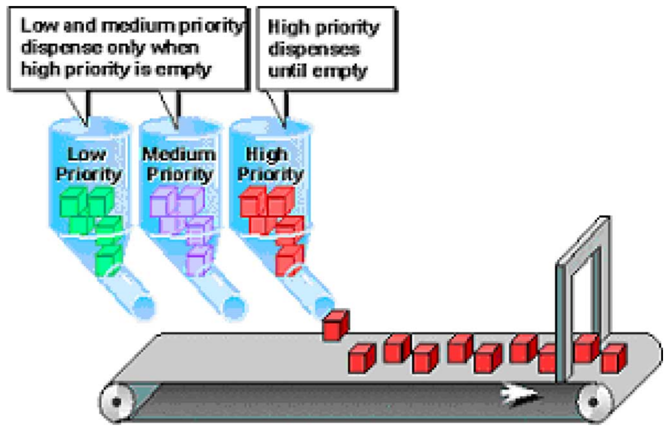

Fig. 11. Priority queuing from [14].

publishers will add the relative subscriber address into their subscriber lists. When a GOOSE message becomes available at the publisher, it can be multicast to all the subscribers by the source making use of the IP group address instead of individual subscriber addresses. However, since the GOOSE message is to be tagged with a logical handle, the task of binding includes the processing of the subscriptions in order to fetch the subscriber addresses forming a subscribers list to accompany the GOOSE message. This has to be repeated before the delivery of every multicast GOOSE message updating the list of subscribers taking into account the possibility of new subscriptions and unsubscriptions.

While binding takes place at the publisher, filtering needs to be carried out at each subscriber to filter out the unwanted messages. Although the possibility of receiving unsubscribed GOOSE messages at the subscriber is quite low, it would still be desirable to include a filtering mechanism. The relative overhead of the filtering is quite small when the publisher-based subscription mechanism is used. Each GOOSE message is tagged with a logical handle, which includes information about the publisher. The complexity of the filter and hence the overhead will be reduced since the filter only needs to evaluate the logical handle rather than the whole message contents. The evaluation of the logical handle includes matching the publisher's name with one of the names in the subscriber's list of publishers. If a match is found, then the message will be accepted and processed. Otherwise, the message will be rejected and destroyed.

D. QoS

Publisher/subscriber systems usually address mechanisms for message ordering and reliability of message delivery. One such example is "Priority Queuing". Priority queuing uses multiple queues as shown in Fig. 11, which are serviced with different priority levels.

The highest priority queue containing the highest priority messages is usually serviced first. In the case of any congestion, packets residing in the lower priority queues will be dropped [14]. This kind of queuing is perfectly suitable for the delivery of GOOSE state change messages, which certainly have the highest priority level on the network. Therefore, with the use of an appropriate scheduling mechanism, it is beneficial to make use of priority queuing in the publisher/subscriber communication model exclusively designed for substation communications. Besides the use of congestion management mechanisms such as priority queuing, the use of congestion prevention mechanisms such as the Weighted Random Early Detection (WRED) [14] for congestion avoidance is also beneficial. WRED prevents congestion by starting to drop low priority packets only in the case of a future congestion detection to ensure the delivery of mission critical messages such as GOOSE messages.

In this research, the authors propose the synchronized use of the user datagram protocol (UDP) and Resource Reservation Protocol (RSVP) to satisfy the most important QoS parameter concerned with the delivery of GOOSE messages, which is the maximum application-to-application delay requirement of 4 ms. The UDP is a connectionless transport layer protocol, which has a very fast response time and a very low overhead. Using the RSVP protocol, each publisher can easily specify the upper bound of the delay, which in the case of GOOSE messages will be 4 ms. Once the specifications are given, then the delivery of GOOSE messages takes place taking the traffic specification into consideration at every step along the network.

However, the mechanisms described above are not adequate when addressing some other issues concerned with substation communication systems. For example, with the use of the UDP and RSVP, it is quite possible to satisfy the 4 ms delay condition. However, the reliability of messages becomes a major concern since UDP cannot provide reliable messaging at all. On the other hand there is a different interpretation in substations for the relationship between reliability and delivery delay. Timely and reliable transmission of messages implies that GOOSE messages need to be repeated in the case of failures until their hold time expires, while not exceeding 4 ms application-to-application delay criteria. This can be achieved by a mechanism, which trades off delivery delay against delivery reliability. What is needed is a guaranteed delivery mechanism operating above the level of the UDP transport protocol. Such a mechanism running above UDP will be superior to the transmission control protocol (TCP) since it will prevent the uncontrollable communication latency that results in the case of TCP. Moreover, by limiting the number of retransmissions the necessary tradeoff can be achieved since UDP will not get stuck trying to retransmit the messages forever destroying the timing determinism completely.

IV. Design and Implementation of the Real-Time (RT) Publisher/Subscriber Model

The key characteristic behind the design of the real-time publisher/subscriber communication model developed for substation applications can be briefly summarized as follows. Non-real time activities such as getting publication or subscription rights happen outside the real time loop ideally at the startup. On the other hand, the generation, transfer and reception of messages are real-time activities happening in the real-time loop requiring very fast response times. There are two main types of nodes. Subscriber nodes can only subscribe to messages whereas publisher/subscriber nodes can publish messages as well as subscribe to messages published by other nodes.

A. Registering

Ideally at the startup, each publishing node must be associated to a multicast group each having a unique label, which is referred to as the IP multicast group address or in some cases as the multicast membership address. The Multicast Membership Service (MltcMS) is responsible for storing the IP group addresses. Entries within the MltcMS contain publishers' IP host names and multicast group addresses. Various nodes on the network communicate with the MltcMS to create/ delete entries and to add/delete their publication/subscription rights.

A node interested in publishing messages gets its publication right by creating an entry in the MltcMS's IP group addresses table. On the other hand, those nodes interested in receiving messages from a particular publisher, have to make use of the IP group address in order to join themselves to the multicast group associated with that publisher. This is referred to as registering. Any node interested in subscribing to messages gets its subscription rights by retrieving the IP group address for the multicast group it seeks to join from the MltcMS. Once the subscriber obtains an IP group address, it can complete its registration and join a multicast group by following the subsequent sequence:

1) The registry manager process running within the subscriber joins a multicast group, Group 1, by sending a join request to its IP module using a remote interrupt.
2) The IP module forwards an IGMP membership report message to the neighboring multicast router.
3) The multicast router sets up a distribution tree for Group 1 adding the interface details of the joining subscriber so that it can receive packets sent to Group 1.

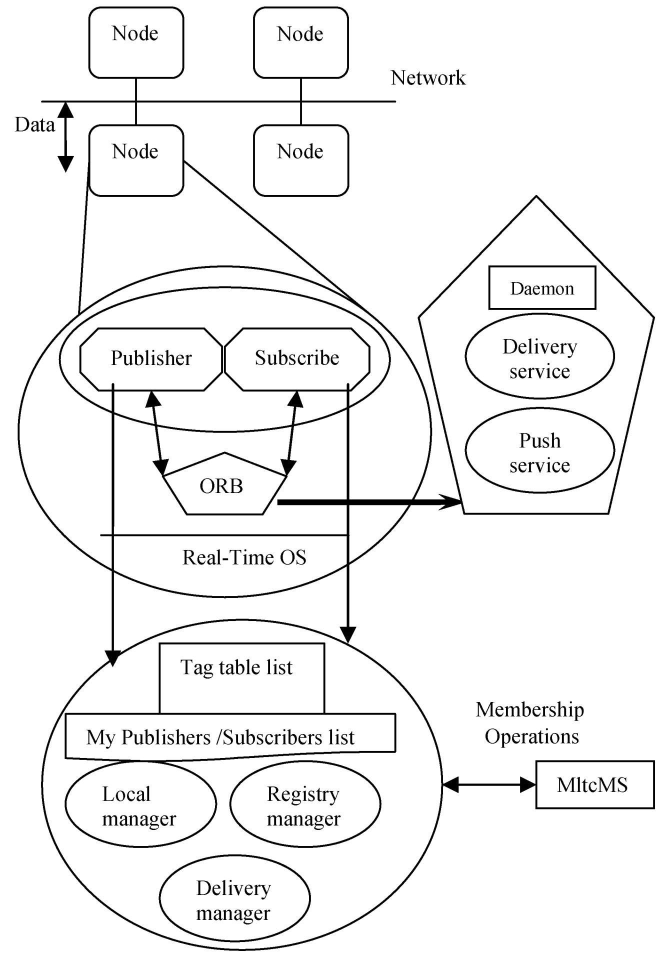

Fig. 12. Architectural components of the push communication model.

B. Subscription, Binding, and Filtering

Registering has to be followed with a subscription request sent to the relevant publisher. It should be kept in mind that registering basically serves the purpose of setting up a distribution tree for each multicast group within the multicast router. In applications where publishers need not to be aware of their subscribers, the process of registering is by itself adequate. However as indicated in Section III-B, in substation applications each publisher is required to hold detailed information related to their subscribers. Thus, the mechanism of sending subscription requests and receiving confirmation messages in return has been designed and implemented in order to solve this problem. Subscription requests are sent immediately after the subscriber registers for multicast groups. One subscription request has to be sent to each one of the publishers. A subscription request is no more than a packet containing a number of fields.

By setting the fields appropriately, a subscriber can inform any publisher over the network about its own local details such as node name and IP address in addition to information specifying whether it wants to subscribe for GOOSE and/or MSVCB messages produced by that node. Hence, a combined approach of publisher and subject based subscription mechanism has been adopted, which has a number of advantages when used in conjunction with the registering process.

Strictly speaking, the task of binding in publishers is unnecessary since each publisher uses a multicast group address to publish its messages. However, each publisher still produces a list of its own publishers to be tagged onto the outgoing multicast message. This serves two purposes. First of all, such a list can be cross-examined by the event service (router) against its own registry list reducing the possibility of unwanted messages from being sent to nodes showing no interest in them. Secondly when a multicast message reaches its destination, the subscriber finds the opportunity of evaluating and storing a list of subscribers subscribing to the same publisher as required in substation applications. Similarly, the task of filtering can also be fully avoided since after all the measures taken, the chances of an unwanted message reaching at any node is fairly low. However, with the intention of being on the safe side, a fairly simple filtering process has been implemented that checks the source of messages comparing it to the list of publishers. In cases where a match can not be found, the packet will simply be destroyed.

C. Architectural Components

Fig. 12 shows the architectural components of the real-time communication model, which can also be considered to represent "push communication" where data is pushed by information sources to information sinks. Such a model is appropriate and efficient for periodic and synchronous updates between sources and sinks as it is in the case of substation applications. The following sequence of events occurs when

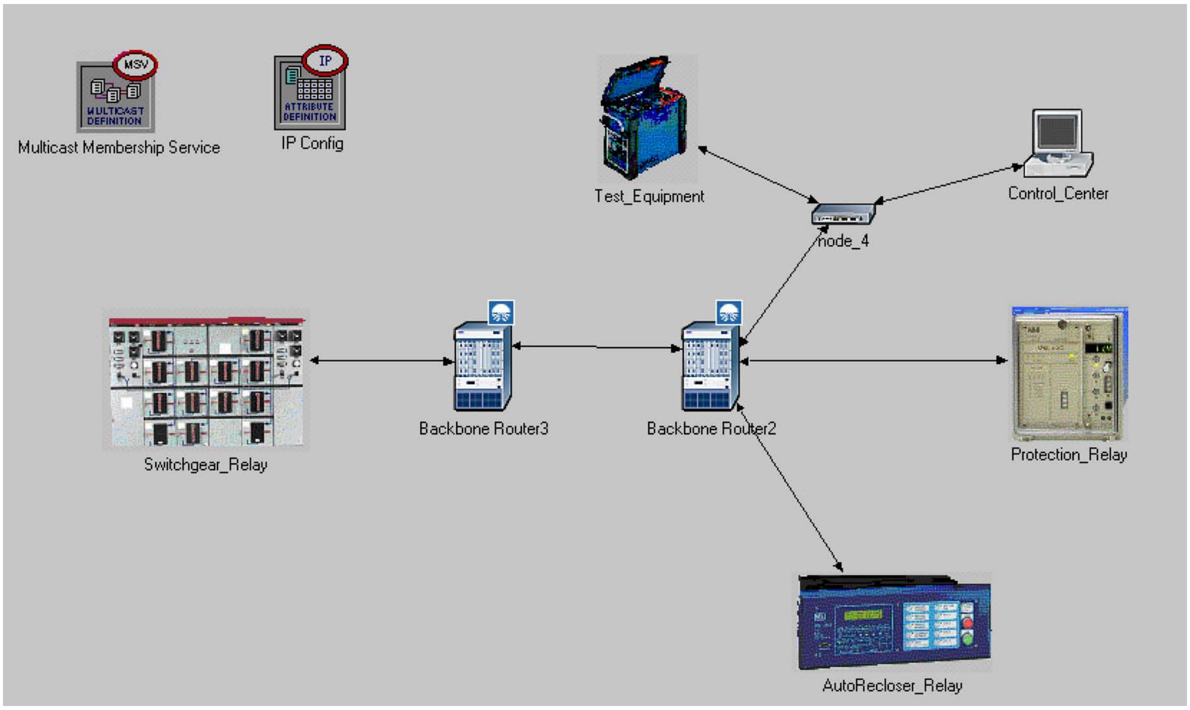

Fig. 13. Simulation test setup.

a source (publisher) pushes packets (data) out of its output interface destined for a particular multicast group.

1) The application running within the source multicasts a packet using the multicast membership address.
2) Once the packet reaches its rendezvous point (the IP address of the router responsible for distributing multicast application traffic to the specified multicast group), the router's IP process forwards the multicast packet to the "ip_pim_sm" process, which is one of the child processes of the IP module.
3) The ip_pim_sm process makes multiple copies of the multicast data and sends one copy to each of the subscribers listed in the multicast route table.

All the architectural components shown in Fig. 12 assist in the successful operation of the push communication model in one way or another. An individual discussion for each one of the architectural components is provided below:

Local Manager: Local manager running in every source and sink is responsible for the creation/deletion of entries in the MltcMS and addition/deletion of publication or subscription rights. One of the tasks of the local manager is to periodically update the multicast membership addresses list of a node based on the periodic info received from the MltcMS.

Registry Manager: Subscribers interested in joining any multicast group do this with the help of the registry manager, which simply contacts the IP module expressing subscriber's interest in joining a group. It also handles the process of sending subscription requests and evaluating confirmation messages received. On the publisher side, registry manager is mainly used for assessing subscription requests on their arrival building the "subscribers" component of the MyPublishersSubscribers list. In cases where publishers subscribe to other publishers multicast groups, registry manager also performs the tasks described above for a subscriber node including the update of the "Publishers" component of the MyPublishersSubscribers list.

Push Service: Push service is in charge of pushing messages out of an ethernet interface destined for their rendezvous point.

Delivery Manager: Delivery manager has the task of encapsulating GOOSE or MSVCB messages into a packet along with the publisher's Publishers/Subscribers list. It then forwards the packet to the push service running in each ORB.

Delivery Service: Delivery service in an ORB transfers application packets, received from other nodes within the network, to the remote application layer.

V. Performance of the Real Time Publisher/Subscriber Model

This section mainly discusses the simulation of the combined IEC 61850 and publisher/subscriber communication model in a switch-router-based multilevel network as shown in Fig. 13 in order to measure several statistics such as the application-to-application delay for various packets transmitted, UDP session delay and ethernet delay. Although UDP is not used by IEC 61850 but an equivalent specification, the authors propose the use of the UDP protocol in this project. The real-time publisher/subscriber communication model described in the proceeding sections has been implemented using the OPNET Modeler software, which is an object-oriented simulator enabling the authors to model data nodes and network traffic.

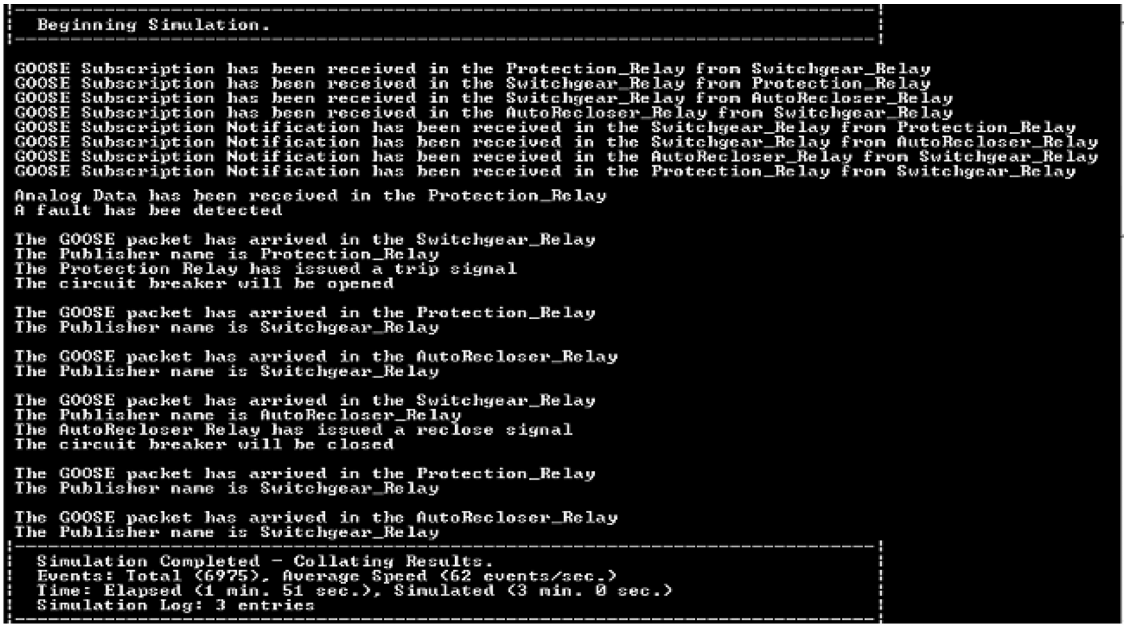

Fig. 14. Simulation console output.

Three main IEDs (Switchgear_Relay, Protection Relay and AutoRecloser_Relay) were configured according to the rules specified by the IEC 61850. The Protection_Relay consists of the logical nodes LLNO, LPHD, and PSCH [15] whereas the AutoRecloser consists of the logical nodes LLNO, LPHD, and RREC [15]. The Switchgear_Relay, on the other hand, is a composition of the LLNO, LPHD, XCBR, and XSWI logical nodes [15]. Device parameter configuration tools are used to create DataSets and set up the attribute values of GoCB in each one of the three IEDs.

The Test_Equipment simulates a short circuit and feeds the corresponding current and voltage values into the Protection_Relay. On receiving a fault current, the Protection_Relay issues a Trip Signal to the Switchgear_Relay in the form of a GOOSE message. When the Switchgear_Relay receives the GOOSE message from the Protection_Relay, it decides to open the circuit breaker in response to the GOOSE message. It then sends a GOOSE message containing the status of the circuit breaker and also the switches both to the Protection_Relay and the AutoRecloser_Relay. Once the AutoRecloser_Relay receives the status of the circuit breaker, it issues a Close Signal to the Switchgear_Relay, which acts against the Close Signal by reclosing the circuit breaker before sending the new status of the circuit breaker and the switches back to the AutoRecloser_Relay.

The event-by-event simulation summary received on the simulation console is shown in Fig. 14. Fig. 15 shows the application-to-application delay statistic of the GOOSE packets received in each one of the IEDs. The measured delay includes the transmission delay as well as the delay through the protocol stack. This has been achieved by time stamping the messages at the application layer. We can clearly identify the delay to be in the vicinity of 0.8 ms which not only satisfies the 4 ms maximum delay within substations but also the 1 ms extra high performance requirement.

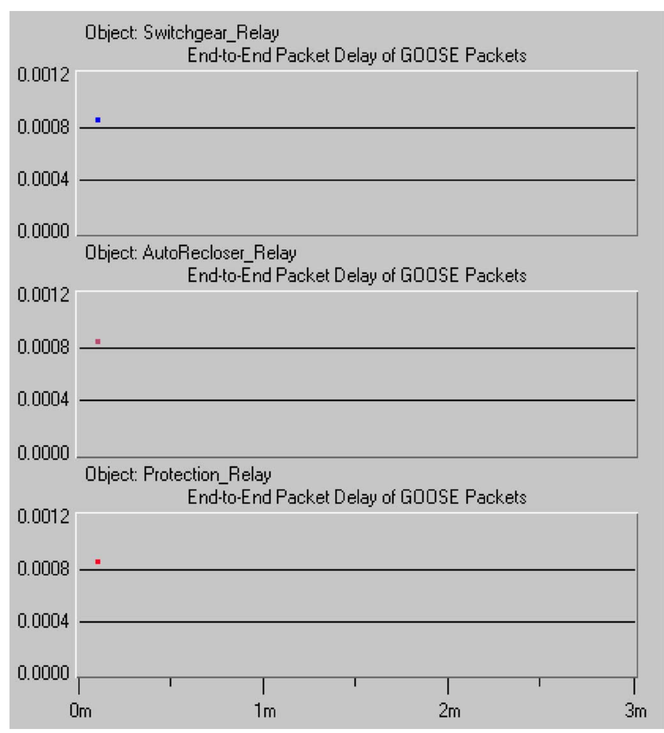

Fig. 15. Packet delay of GOOSE messages.

Fig. 16 shows the UDP session delay of the Switchgear_Relay for packets received/sent over UDP including GOOSE packets, and GOOSE Subscription packets. Session delay varies in between 1.20-ms and 0.70-ms still satisfying the 4 ms maximum requirement. Fig. 17 shows the ethernet delay of the Test_Equipment device for packets received/sent over TCP mainly including the analogue values of the currents and voltages.

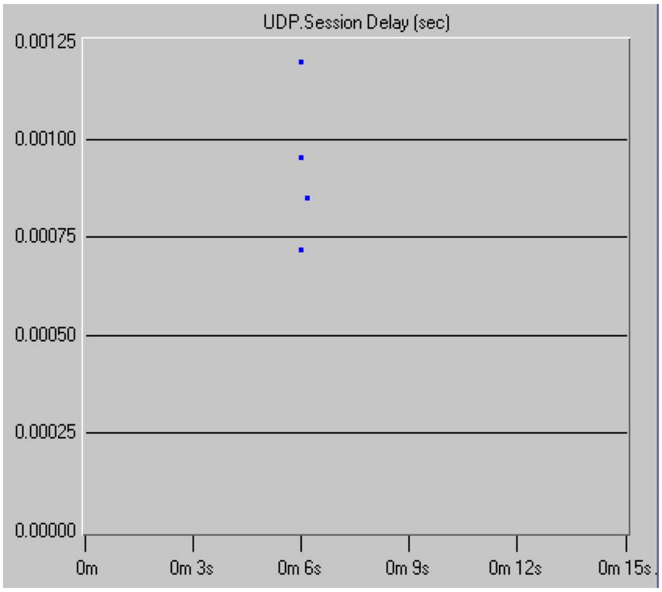

Fig. 16. UDP session delay of the Switchgear_Relay.

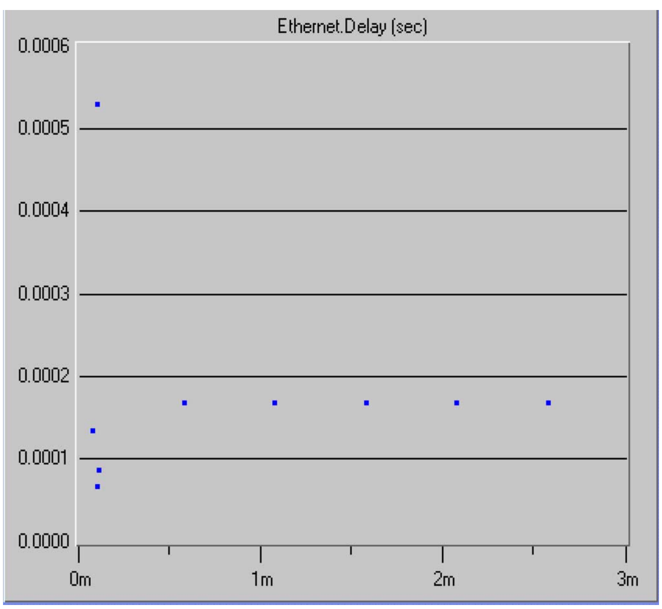

Fig. 17. Ethernet delay of the Test_Equipment device.

VI. Cyber Security

The use of automated control systems and microprocessor based protection systems in the electric power industry along with the need to provide remote access into these systems has increased the risk of these systems being vulnerable to unauthorized hostile access. The extension of communication networks out to the substation has magnified the possibility of cyber attacks ranging from espionage to sabotage via electronic intrusion and computer hacking [16]. However, the electric power industry has long been aware of this threat and taken the necessary steps to reduce risk and mitigate vulnerabilities. IEC report 62210 [17] “Data and Communications Security”, a report developed and circulated throughout the IEC in 1999 and published in 2003, has lead to the establishment of the IEC Working Group (WG) 15 titled “Power System Control and Associated Communications—Data and Communications Security” to look across the other working groups to address end-to-end security recommending or supplying standardized security enhancements as needed [18], [19]. Securing application-to-application information exchange through supplying strong authentication, message integrity and confidentiality (e.g., encryption) enhancements as well as Spoof/Replay protection to the IEC TC 57 protocols is the main focus of the WG 15.

A new document known as IEC 62351 [20] incorporates many of the new work items currently under development by IEC TC 57 WG 15. The IEC 62351 document consists of seven main sections and includes security considerations for profiles including TCP/IP, MMS, IEC 60870-5, and IEC 61850.

The fact that many TC 57 communication profiles including IEC 61850 ACSI over TCP/IP and IEC 60870-5-104 are based on the TCP/IP has resulted in the need for a common security solution to be investigated for profiles using TCP/IP. IEC 62351-3 specifies the use of Transport Layer Security (TLS) in order to secure IEC TC 57 protocols over the Internet. Likewise, IEC 62351-4 includes security considerations for profiles that include the MMS. The addition of application-level authentication through the use of TLS’s authentication measures is the main suggestion.

Security enhancements are also required when implementing and using communication profiles of IEC 61850 in non-secure environments. The basic design principles are that secure and non-secure profiles must be able to unambiguously co-exist with a single set of identity management policies for all profiles using mainstream IT methodologies. The main security objective is to prevent eavesdropping and spoofing/playback of captured data from non-trusted entities as well as assuring authorized access even within a closed private network. Security for five IEC 61850 profiles: Client/server, GOOSE, GSSE, GSSE management and Sample Measured values (SMV) have been developed and packaged into IEC 62351-6. TLS encryption is used for client/server profiles where the data is encrypted so that only the two communicating entities can understand the data. For the peer-to-peer multicast profiles such as GOOSE, encryption is not acceptable since it affects the strict transmission rates required for multicast datagrams. Hence, authentication is the only security measure included for those where digital signatures are used ensuring that the entity at the other end is known and trusted [21], [22].

In the project described in this paper, no security measures have yet been considered. However, the incorporation of security measures in accordance with the security measures described in IEC 62351-6 is being considered and any future work is to reflect on such changes.

VII. Conclusion

In this paper, the authors described the IEC 61850 standard and the communication middleware architecture designed as a means of satisfying the unique behavior and communication needs of the IEC 61850 protocol. After discussing IEC 61850 ACSI models and services, the authors identified the design constraints for effectively designing a suitable real-time publisher/subscriber data delivery middleware for substation communication systems. A detailed insight into the model was provided including the architectural components of the publisher/

subscriber model on top of discussing the processes of registering, subscription, binding and filtering.

The designed communication service does not include any object or service models. It only supports a variety of communication mechanisms needed for the successful operation of ACSI clients and servers. Therefore, it is small, fast and relatively efficient. The publisher/subscriber communication model provides an identical programming model for unicast or multicast, and an identical programming model for sources and sinks thus achieving ease of programming. The physical network is not bound to any protocol meaning that it does not need to concern itself with the type of protocols being used, a concept referred to as "protocol independence". The design achieves portability by marinating protocol independence and constant interface. Efficiency has been achieved by making sure that the real-time communications make minimal expenditure of computer resources by using locally stored information and duplication of messages going to multiple receivers.

The remainder of the paper evaluated the performance of the publisher/subscriber communication model with the help of simulations. The simulations demonstrated that the designed middleware architecture can be used effectively to provide the necessary communication services for the IEC 61850 protocol. It only adds minimal overhead to the underlying protocol stack maintaining the overall delay within acceptable bounds.

References

[1] D. J. Dolezilek, Power System Automation Schweitzer Eng. Labs., 2000. Pullman, WA [Online]. Available: http://www.selinc.com/techpprs/6091.pdf.

[2] M. Adamiak and M. Redfern, "Communication systems for protective relaying," *IEEE Comput. Appl. Power*, vol. 3, no. 3, pp. 14–22, Jul. 1998.

[3] M. C. Janssen and C. G. A. Koreman, Substation Components Plug and Play Instead of Plug and Pray: The impact of IEC 61850 Kema T&D Power. Amsterdam, The Netherlands [Online]. Available: http://www.nettedautomation.com/download/Substation_Components_Plug_and_Play_instead_of_Plug_and_Pray.PDF.

[4] *IEC TC-57. Communication Networks and Systems in Substations—Part 7-1: Basic Communication Structure for Substation and Feeder Equipment—Principles and Models*, Int. Std. IEC 61850-7-1, Int. Electrotech. Comm., 2003, Geneva, Switzerland.

[5] *IEC TC-57. Communication Networks and Systems in Substations—Part 7-2: Basic Communication Structure for Substation and Feeder Equipment—Abstract Communication Service Interface (ACSI)*, Int. Std. IEC 61850-7-2, Int. Electrotech. Comm., 2003, Geneva, Switzerland.

[6] *IEC TC-57. Communication Networks and Systems in Substations—Part 8-1: Specific Communication Service Mapping (SCSM)—Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*, Int. Std. IEC 61850-8-2, Int. Electrotech. Comm., 2004, Geneva, Switzerland.

[7] F. Olken, H. A. Jacobsen, and C. McParland, "Middleware requirements for remote monitoring and control," in *OMG-DARPA-MCC Workshop on Compositional Software Architectures 98* [Online]. Available: http://www.objs.com/workshops/ws9801/papers/paper097.html.

[8] D. Holstein, "Communication architectures for substation automation," in *WiseOwl Newsletter*, 1999 [Online]. Available: http://www.opusss.com/CommunicationArchitectures.htm.

[9] Ö. K. Sahingoz and N. Erdogan, "Agvent agent systems," in *Proc. 4th Int. Symp. Engineering of Intelligent Systems* , Madeira, Portugal, 2004, pp. 548–554 [Online]. Available: http://www.cs.itu.edu.tr/~erdogan/sahingoz-erdogan-SOFSEM2004.pdf.

[10] A. Apostolov, C. Brunner, and K. Clinard, "Use of IEC 61850 object models for power system quality/security data exchange," in *CIGRE/IEEE Power Eng. Soc. Int. Symp.*, Oct. 8–10, 2003, pp. 155–164.

[11] *IEC TC-57. Communication Networks and Systems in Substations—Part 7-3: Basic Communication Structure for Substation and Feeder Equipment—Common Data Classes*, Draft Std. 61850-7-3, Int. Electrotech. Comm., 2001, Geneva, Switzerland, IEC.

[12] *Communication Delivery Time Performance Requirements for Electric Power Substation Automation*, Std. 1646-2004, IEEE Power Eng. Soc. (Power Eng. Soc.), Feb. 2005, IEEE Std., NY, USA.

[13] J. T. Tengrid, The SCADA Four Millisecond Myth OPUS Publishing, 2005 [Online]. Available: http://www.utc.org/file_depot/0-10000000/0-10000/1013/conman/The+SCADA+Four+Millisecond+Myth.pdf.

[14] Cisco Systems, 1999, Quality of Service End-to-End With CISCO IOS [Online]. Available: http://www.scd.ucar.edu/hps/CISCO/sld001.htm.

[15] *IEC TC-57. Communication Networks and Systems in Substations—Part 7-4: Basic Communication Structure for Substation and Feeder Equipment—Compatible Logical Node Classes and Data Classes*, Draft Std. 61850-7-4, Int. Electrotech. Comm., 2001, Geneva, Switzerland, IEC.

[16] *Recommended Practice for Network Communication in Electric Power Substations*, IEEE Std. P1615, IEEE Power Eng. Soc. (PES), 2005, IEEE Std. Draft. IEEE Substation Committee, Draft 0.5.

[17] IEC, IEC TC-57. Power System Control and Associated Communications—Data and Communication Security Geneva, Switzerland, Tech. Rep. IEC TR 62210, 2003, 1st Ed.

[18] Office Elect. Del. Energy Rel., National SCADA Test Bed: A Summary of Control System Security Standards, Activities in the Energy Sector U.S. Dept. Energy, Oct. 2005.

[19] J. Gillerman, H. Falk, and R. Mackiewicz, "Focus on IEC TC 57: Power system reliability and profitability," *IEEE Power Energy Mag.*, vol. 3, no. 4, pp. 66–67, Jul./Aug. 2005.

[20] *IEC TC-57. Data and Communication Security*, Int. Std. IEC 62351 TS, IEC, 2005, Geneva, Switzerland.

[21] F. Cleveland, IEC TC 57 Security Standards for the Power System's Information Infrastructure—Beyond Simple Encryption Xanthus Consulting Int., Oct. 2005.

[22] SISCO Systems, 2005, IEC 61850 Security—Requirements, Solutions, and Future Trends [Online]. Available: http://www.66.102.7.104/search?q=cache:EztSI-rDr0sJ:sharepoint.ucausersgroup.org/helpdesk/Lists/Help%2520Desk%2520Issues/Attachments/56/Security%2520for%2520IEC%252061850.ppt+IEC+61850+Security+sisco&hl=en.

Cagil R. Ozansoy was born in Cyprus in 1980. He received the B.Eng. degree in electrical and electronic engineering (Hons.) from Victoria University, Melbourne, Australia, in 2002, where he is currently pursuing the Ph.D. degree in the area of power system communications.

He has several publications in power system communications.

Aladin Zayegh received the B.E. degree in electrical engineering from Aleppo University, Syria, in 1970 and the Ph.D. degree from Claude Bernard University, Lyon, France, in 1970.

In 1980, he joined the Faculty of Engineering, Tripoli, Libya. Since 1984, he has held lecturing positions with Footscray Institute of Technology, Melbourne, Australia, and Victoria University of Technology, Melbourne. He is currently Associate Professor and the Head of the School of Electrical Engineering, Faculty of Health, Victoria University.

Akhtar Kalam received the B.Sc. degree from Calcutta University, Calcutta, India, in 1969 and the B.Sc. degree in engineering from Aligarh Muslim University, India, in 1973. He received the M.S. degree from the University of Oklahoma, Norman, and the Ph.D. degree from the University of Bath, Bath, U.K., for his work on the application of distance protection to series-compensated extra high-voltage line.

He has been actively engaged in the teaching of power systems for over 20 years, both in the School of Electrical Engineering, Victoria University of Technology, and overseas. He has conducted research, provided consultancy, and has more than 280 publications on power system protection and independent power generation. His major interests are power system analysis, power system protection, and expert system application in power system, cogeneration, and renewable energy.