IBM Research GmbH

Saeumerstrasse 4
CH - 8803 Rüschlikon
Switzerland

Subject: Initial inputs to Ecogrid task 3.5 on DER and aggregator communication and standards contributions

Version: 1
Date: July 19, 2012

Authors: Aditya Mishra, Carl Binding, Dieter Gantenbein, Bernhard Jansen, Olle Sundstrom

1. Introduction

The document surveys a set of existing communication protocols and data models (from IEC 61850 standard) for transferring process data and supervisory control information between networked devices and/or computer applications. Suitability of the surveyed protocols for their application in real-time price exchange between DER (Distributed Energy Resources) and aggregator is analyzed. Data/information models and protocols covered in the document include IEC 61850 information model, MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Events), CIM (Common Information Model), FlexPrice, Olympic Peninsula [9] price exchange protocol, Hierarchical transactive control [12].

2. Requirements

Here we present a short summary of the data, security, and scalability requirements to a real-time price exchange protocol. Details of these requirements are available in [10]. Data model of such a protocol should allow for exchanging the real-time market price, currency of the market price, energy unit for the market price, start and end times for which the real-time market price is valid, and a forecast of the real-time prices consisting of any number of forecast values. The real-time price information is a public information therefore we don't need to encrypt the information. However, the protocol should be able to ensure the authenticity, integrity, non-repudiation, and non-replay-ability of the price messages. Additionally the real-time price exchange system should scale from a few thousands of real-time market price receivers to a very large number (potentially few million).

3. Survey

3.1 IEC 61850

IEC 61850 specifies the detailed information model required for communication by DER (Distributed Energy Resources) plants. The specification includes both, communication between DER units and plant management system, and also between the DER system and aggregator [3] [7]. A summary of IEC 61850 is presented in [16].

3.1.1 IEC 61850 Data Model

IEC 61850 uses a hierarchical data model. Server is the top most node in this hierarchy, an IED (Intelligent Electronic Device) can (theoretically) consist of multiple servers. Each server hosts multiple files or logical devices. A logical device is a logical representation of a physical device. Each logical device has a set of logical nodes. These logical nodes represent the functions supported by an IED. IEC 61850-7-4 defines a collection of template logical nodes. The standard requires every logical device to also have two specific logical nodes, viz., Logical Node Zero (LN0) and LPHD, which represent the logical device and the physical device. A logical node may contain several data objects. Data exchanged between logical nodes are modeled as these data objects. Each data object in turn consists of several data attributes. Additionally, IEC 61850 provides data set as another way of managing and exchanging the data attributes. A data set can contain both data attributes and data objects.

IEC 61850 uses several DER logical nodes and DER CDC (Common Data Classes) for specifying DER information model. Logical nodes DSCC (DER energy schedule control) which controls the use of DER energy and ancillary services schedules, and DSCH (DER Energy and ancillary services schedule) which defines a DER energy and/or ancillary services schedule, can be used for communicating the real-time market price information. DSCH contains the objects of CDC SCA and CDC SCR. SCA and SCR together provide a means of defining either an absolute or relative time array of setting values. Here, setting values can be the real-time price values. Both SCA and SCR have fields for specifying the currency of price and unit of electricity.

3.1.2 IEC 61850 Communication Pattern

In order to address the varied kinds of interactions inside and across the substations IEC 61850 defines five kinds of communication profiles, viz., ACSI (Abstract Communication Service Interface), GOOSE (Generic Object Oriented Substation Event), GSSE (Generic Substation Status Event), SMV (Sample Measured Value), time synchronization. ACSI is used by remote operators/monitors to query and control devices in a substation. All communication in ACSI is done in client-server mode where services are requested by client applications and responded by the servers. Before requesting a service from server, a client application must first establish a valid two-party application association (TPAA) with the server. GOOSE messages are used to transmit state of a device at a fixed intervals. These messages are sent via multicast/broadcast so that many recipients can be informed about sate of the station at the same time. It works on publish-subscribe mechanism on multicast or broadcast MAC addresses.

3.1.3 IEC 61850 Transport Protocol

ACSI uses TCP at transport layer, time synchronization messages are sent over UDP, GOOSE and SMV messages are broadcast over ethernet link layer, and GSSE uses GSSE T-Profile as described in [4]. GSSE T-profile uses a connectionless transport and network layer. The destination T-DATA address for a GSSE message contains a multicast MAC address. The source T-DATA address for a GSSE message contains a unicast MAC address.

3.1.4 Merits of IEC 61850

1. IEC 61850 is an established standard which is already being used by several DERs. Using the data model specified here would enable us to leverage the already existing communication infrastructure between DERs and aggregator for exchanging the real-time price information.
2. IEC 61850 provides data and communication security. IEC 62351 defines the security for IEC 61850. IEC 61850 references IEC 62351. IEC 62351 defines data and communication security for communication protocols using TCP/IP and MMS (Manufacturing Message Specification) based communications. Security objectives include authentication of data transfer through digital signatures, ensuring only authenticated access, prevention of eavesdropping, prevention of playback and spoofing, and intrusion detection [17].

3.1.5 Demerits of IEC 61850

IEC 61850 is largely based on client-server communication model. Although data models defined in 61850 can be mapped to different protocols, current mappings in the standard are to MMS (Manufacturing Message Specification), and GOOSE (Generic Object Oriented Substation Events). GOOSE follows a multicast/broadcast over MAC addresses for transmitting status messages over the Ethernet and cannot be used for communication over WAN. MMS, on the other hand, follows a point-to-point communication model, which is not suitable for large scale real-time price broadcasting.

3.2 MMS (Manufacturing Message Specification)

MMS is a standard messaging system for exchanging real-time data and control information between controllable manufacturing devices and/or computer systems [4] [8]. MMS enables the exchange of information in application and device independent manner. It is an international standard ISO 9506.

3.2.1 MMS Data Model

The key feature of MMS is the Virtual Manufacturing Device (VMD) model [13]. The VMD model specifies how MMS devices (servers) behave as viewed from an external MMS client application. In general the VMD model defines objects that are contained in the server (e.g., variables contained in the server), services that are available to a client for accessing and manipulating the server objects, and behavior of the server on receipt of service requests from the client.

MMS supports a wide range of variables, arrays and structures which can be used for representing and transferring the real-time price information. The current market price, currency of price, energy unit, start time and end time can be fields in a structure. Forecast for real-time prices can be included in the above structure as an array of another structure containing the desired fields. MMS allows for nesting of structures.

3.2.2 MMS Communication Pattern

MMS defines client/server relationship between networked applications and/or devices. A server is a device or application that contains a VMD (Virtual Manufacturing Device) and its objects (e.g., variables). A client is a networked application (or device) that asks for data or an action from the server. In other words, a client is a network entity that issues MMS service requests to a server. A server is a network entity that responds to the MMS requests of a client. MMS provides services such as Status, UnsolicitedStatus, and Identify for obtaining information and status about the VMD.

3.2.3 MMS Transport Protocol

MMS defines the use of connection oriented transport (ISO 8072/8073) or TCP.

3.2.4 Merits of MMS

Following are some of the benefits of using MMS for sending the real-time price information.

1. It provides interoperability by allowing two networked applications to exchange information without any change from the user of the applications.
2. MMS has a set of features which allow for real-time distribution of data across networks between peers.
3. MMS implementation supports a variety of communication links like Ethernet, Token Bus, RS-232C, OSI, TCP/IP, MiniMAP, FAIS and can connect to several types of networks. This allows for interoperability independent of the network connectivity.

3.2.5 Demerits of MMS

1. Most of the communication in MMS takes place in client/server mode. Here a client sends command/request to the server and server returns a response after processing the client request. This is a point-to-point mode of communication and would not scale well in a scenario where the price information needs to be broadcast to a large number of clients (potentially million) frequently (every five minutes).
2. Authors in [15] present an analysis of MMS and show than MMS protocol has insufficient security mechanisms, and the very few security measures that are available are not implemented in commercially available devices. The standard provides mechanisms for authentication and access control but no other security measures are included in the MMS standard.

3.3 CIM (Common Information Model)

CIM provides a common language and domain model which allows multiple energy management systems (EMS) to exchange management information [6]. In addition to representing the managed systems and management information CIM also provides means to control the managed systems. Hence, instead of having to connect all EMSs with each other in a mesh it is possible to connect the system with an integration platform with CIM as the common language.

3.3.1 CIM Data Model

The CIM represents all major objects used in an electric utility. CIM is documented as a set of class diagrams using Unified Modeling Language. The basic CIM model [1] consists mainly of eleven packages, viz., Core, Domain, Topology, Wires, Meas (measurement), Generation, Production, Generation Dynamics, LoadModel, Outage, Protection. Based on IEC 61970-301 further packages have been developed in later substandards. IEC 61970-302 focuses on data exchange between companies. It has three packages, i.e., EnergySceduling, Financial, Reservation. EnergyScheduling package supports exchange of electricity between energy companies. It provides capability to schedule and account for energy transactions between utility companies. Financial package covers billing aspects. Reservation package provides for energy transaction scheduling. IEC 61970-303 is mainly focused on SCADA (Supervisory Control and Data Acquisition).

3.3.2 CIM Communication Pattern

CIM specification does not describe specific CIM implementations or communication protocols, this is outside the scope of the specification.

3.3.3 CIM Transport Protocol

No transport protocol has been described in the CIM specification. A suitable transport protocol must be chosen based on the application.

3.3.4 Merits of CIM

1. CIM sends only instantiated class attributes in a message instance. Hence, the message payload is no larger than an XML formatted message.
2. CIM UML is extensible. Both standard and private extensions are possible.

3.3.5 Demerits of CIM

1. We need an additional translation/adapter layer to translate every message into CIM.
2. Mapping information to CIM is advantageous in a scenario where multiple different systems want to exchange data and communicate. In such a scenario interconnecting all the systems would require O(n^2) adaptors/translators. Using CIM as a common language can reduce the number of adaptors to O(n). This simplifies the interconnection and saves integration time. However, real-time price broadcast does not require a bidirectional exchange of information/data between devices on the grid. It will more closely resemble a producer consumer setup where there is one producer and many consumers. In such a scenario using CIM will cause additional translation overhead without much benefits.
3. Supporting CIM could be a considerable overhead for several end devices which are only receiving price information. Especially controllers with low memory, processor resources.
4. CIM XML files are transferred without any security signature to ensure the authenticity, integrity of the file contents. This means it would not be possible to verify that the contents of a CIM XML file have originated from the expected source and have not been tampered.

3.4 Olympic Peninsula

Olympic Peninsula (OP) project employed a two-way market in which both suppliers and consumers of electricity placed bids [9]. Consumers of the real-time market submitted price bids for the expected power to be used by them over next 5-minute interval. The bid price indicates the price at which they would be willing to curtail the stated power consumption. Most customers submitted two bids for each 5-minute interval, one for the controllable load and the other for uncontrolled load. The retail market was cleared every

5 minutes. The market's clearing price and cleared power was published back to all the bidders.

3.4.1 OP Data Model for Price Exchange

[9] does not give many details about the message format used for exchanging bids. However, it seems each cleared bid message to the bidders contains the real-time market price, currency of the market price, energy unit for the market price, start and end time for which the market price is valid, and total cleared power quantity.

3.4.2 OP Communication Pattern

The market features of real-time price contract operations were carried out centrally at a server. Participating customers (both consumers and producers) will send their price bids to the server every 5 minutes. The real-time market cleared every 5 minutes. The shadow market was set up to communicate the real-time aggregate local marginal price of electricity to each customer involved.

3.4.3 OP Transport Protocols

There is no mention of the transport layer used for communication. It seems to be independent of transport protocol and can work with TCP/IP, or UDP. However, UDP would need additional measures for ensuring reliability of communication.

3.4.4 Merits of OP Price Market

1. As both the suppliers of electricity and consumers of electricity place their bids on electricity, the actual price of available electricity can be determined accurately. There are no inaccuracies due to predictions/forecasts.
2. It can be applied within a building as was done in the OP project to create market competition between space conditioning zones. It also scales well regionally as the project did at multiple residential and commercial building locations on the Olympic Peninsula [9].

3.4.5 Demerits of OP Market

1. Price bidding from every end device every 5 minutes can potentially make the bidding server a performance bottleneck. Especially when millions of end devices would be required to place bids.
2. Most of the customers would not like to bear the extra overhead of deciding their bidding strategy very often.
3. In a real-time price broadcast scenario where the prices need to be published every 5-minutes there would be a real-time constraint on each of the end device to decide its bid and make sure it reaches the bidding server within 5-minutes. This may not be possible for devices with slow Internet connectivity or for devices facing network outage/congestion. In such scenarios the bidding server can get only partial information.
4. The bidding algorithm needs to be sophisticated enough to tackle the cases of partial and missing information.

5. Network security measures like encryption and message signature would be needed for each bidding message to safeguard the bidding process against attacks from malicious devices. These measures when applied to each message will only make scalability more challenging by increasing messaging overhead.
6. There was no consideration for providing security to the price messages. For example, no provision exists for ensuring authenticity, integrity, non-repudiation of the price messages.

3.5 FlexPrice

FlexPrice protocol is proposed for sending real-time price information to electricity producers and consumers to facilitate their electricity production/consumption planning [11]. FlexPrice assumes a client-server setting with a request response model of communication for information exchange.

3.5.1 FlexPrice Data Model

Two main types of messages are defined in FlexPrice protocol, i.e., request message and response message. A summary of the messages formats is presented here. Details of the message formats can be found at [11]. The request message can be as simple as just specifying the id of requesting client. A general request message would, however, contain an id, signalTypeId, startingTime, duration. Here 'signalTypeId' refers to the type of signal that is requested from the server.

A general FlexPrice response message contains an id which is either the operatorId or signalId, signalType, and entries/values of the signal. 'signalType' contains the following fields:-

- SignalTypeId: An id for identifying the signal type. Unique for every signal type in the project.
- Type: A project defined set of types for sending different kind of signals, like 'Spot price', 'Net price', etc.
- Uncertainty: If the signal is of uncertainty type then the kind of uncertainty can be specified here.
- Description: This is project dependent and is used to distinguish between multiple signals.
- Currency: This indicates the currency for the values in the entries according to ISO-4217
- perUnit: Describes the value used in the entries. For example, Wh, kWh, MWh.
- Entries: Actual signal entries.

3.5.2 FlexPrice Communication Pattern

The 'FlexPrice' does not define how data communication should be done or how the price signal should be exchanged between the sender and receiver. However, for common understanding a client-server mode of communication between the receiver and sender can be assumed. Here, the client sends a 'FlexPriceRequest' message to the server and server responds with a 'FlexPriceResponse' message.

3.5.3 FlexPrice Transport Protocol

There is no mention of any recommended transport layer protocol in [11]. The communication protocol is fairly independent of transport layer protocol. It can be implemented on top of TCP/IP or any other popular transport protocol of choice.

3.5.4 Merits of FlexPrice

1. FlexPrice can support geographically dependent services for customers, like Net pricing.
2. Clients in FlexPrice can choose to request only specific signals and not receive the entire message. This is particularly useful for controllers with limited memory resources and compute power.
3. In addition to the price forecasts FlexPrice also includes a field specifying the confidence of prediction values. This additional information can be useful to the customers for planning their electricity consumption.

3.5.5 Demerits of FlexPrice

1. No mention of support for security in the price messages. At the least we need to ensure authenticity, integrity, non-repudiation, and non-replayability of the messages.
2. FlexPrice is built around the client-server model of communication. Each client sends a request to the server and based on the specific request server responds. This type of model is not scalable and cannot support millions of customers.
3. Each time a client/customer needs new information, FlexPrice requires it to send a new request. This kind of messaging scheme can easily flood the network with request messages especially when the real-time prices needs to be exchanged every 5 minutes. There is no support for publish-subscribe messaging pattern.
4. In order to support simple requests from clients and geographically dependent services FlexPrice maintains a table of known clients and their settings. Table look-up for millions of entries can be slow. Moreover, when the prices change every five minutes, this could be a performance bottleneck.

3.6 GOOSE (Generic Object Oriented Substation Events)

A GOOSE message is used to exchange data between IEDs (Intelligent Electronic Devices) and is a part of standard IEC61850 [4] [5] [14]. It is a mechanism for fast transmission of substation events. The message can be broadcast to several IEDs simultaneously. GOOSE also provides for real-time data transfer between IEDs by directly embedding data into Ethernet data packets.

3.6.1 GOOSE Data Model

GOOSE supports the exchange of a wide range of possible common data organized by a DATA-SET. The DATA-SET may have several members (numbered from 1 up – the numbers are called MemberOffset). Each member has a member reference referencing the data attribute with a specific functional constraint. DATA-SET can potentially be used for exchanging real-time price information. GOOSE control block (GoCB) class definition has the following attributes.

- GoCBName (GOOSE control name): This attribute unambiguously identifies the control block.
- GoCBRef (GOOSE control reference): Unique path name of a GoCB.
- GoEna (GOOSE Enable): If set to TRUE, indicates that GoCB is enabled to send GOOSE messages.
- AppId (application Identification): A visible string that represents the logical device in which the GoCB is located.
- DatSet (data set reference): This attribute represents the reference of the DATA-SET whose values of members are transmitted.
- ConfRev (configuration revision): Represents a count of the number of times the configuration of the DATA-SET referenced by DatSet has been changed.
- NdsCom (needs commissioning): If TRUE, it means GoCB requires further configuration.

3.6.2 GOOSE Communication Pattern

GOOSE messages are sent on a regular basis to transmit state of a device at a fixed interval (e.g., every second). These messages are sent via multicast/broadcast so that many recipients can be informed about the state of the station at the same time. It works on publish-subscribe mechanism on multicast or broadcast MAC addresses. The publisher writes the values in a local buffer at the sending side; the receiver reads the values from a local buffer at the receiving side. The communication system is responsible to update the local buffers of the subscribers. A generic substation event control class in the publisher is used to control the procedure [14].

3.6.3 GOOSE Transport Protocol

GOOSE data is directly embedded into Ethernet data packets. GOOSE uses VLAN and priority tagging as per IEEE 802.1Q to have separate virtual network within the same physical network and sets appropriate message priority level.

3.6.4 Demerits of GOOSE

GOOSE is designed mainly for data transfer/broadcast over a LAN. Hence, it cannot be used for broadcasting price information over the Internet.

3.7 Transactive Control System

In transactive control, responsive demand assets bid into and become controlled by a single, shared, price-like value signal, which may be in turn, influenced by many local and regional operational objectives of the electric power grid [12]. This approach was first demonstrated for the control of a transmission constraint during the Olympic Peninsula GridWise project [9]. In [12] the authors generalize the transactive control approach by: First, formalizing a hierarchical node structure which defines the nodes and the functional signal pathways between these nodes. Second, fully generalizing the inputs, outputs, and functional responsibilities of each node. Third, including the predicted day-ahead future values in signals. This predictive feature allows the market-like bids and offers to be resolved iteratively over time.

3.7.1 Transactive Control Data Model

[12] does not discuss the details of the data model to be used for message exchange in the transactive control system. Each node receives a value signal input from the upstream node, this input is a single value time series. In principle the value time series should represent a predicted energy price over the next 24 hours, or so. The value signal is not necessarily the monetary energy price though. It is an objective value to communicate the value of the power grid's control benefits via a single value signal at each location. Similarly the demand time series represents a predicted demand for time intervals over next 24 hours, or so. Preferred unit for demand measurement is kW. The authors recommend that the time intervals used for the demand signal and price/value signal should be the same. Also, both demand and price/value should become more accurate and should have finer intervals near term.

3.7.2 Transactive Control Communication Pattern

Transactive control follows a hierarchical communication model. It requires the communication and information flow along the grid to be aligned well with the flow of electrical power. A node is defined as a physical point anywhere in the electric grid where demand may be aggregated and predicted. There is a contiguous hierarchy of nodes from generation point to the end users/devices. An example hierarchy of nodes in downstream would be generation power plant, distribution substation, distribution transformer, sites (home/building), site devices. In the upstream direction demand capacity is aggregated through the hierarchy from end users towards generation plant. A value signal, indicating the value of the power grid's control benefits, is propagated from the generation to the end users in downstream direction. The node will often relay the received input value time series to all the next downstream nodes without any modifications. However, the node may modify the value time series before relaying it downstream according to its own local operational objectives. Each node has only two necessary communication pathways. A value signal is communicated downstream through the node and the demand/capacity signal is communicated upstream.

3.7.3 Transactive Control Transport Protocol

The communication protocol is fairly independent of any transport layer protocol. It can be implemented on top of TCP/IP or UDP or any other popular transport layer protocol of choice. There is no mention of any recommended transport layer protocol in [12].

3.7.4 Transactive Control Merits

1. Transactive control defines very general set of inputs, outputs, and functional requirements of each node. This makes the approach available to a wide set of responsive assets and operational objectives.

2. The predictive feature of transactive signals allows the responsive resources to anticipate and therefore proactively participate in coming peak events. For example, by using electricity at a current lower price and hence possibly avoiding a future higher price.
3. The described approach does not depend on communication of device-specific information and decision making is highly distributed. Hence, the proposed approach might be less vulnerable to some cyber security threats.
4. Inputs and outputs of the generalized nodes are defined in a way that reduces the overall communication bandwidth and help in interoperability.

3.7.5 Transactive Control Demerits

1. In order to ensure safety of grid from security attacks it is necessary to ensure the authenticity, integrity, non-repudiation, non-replayability of the signals/messages being exchanged between the nodes. There is no mention of security in the proposed transactive control system. In the absence of these security measures an attacker can easily increase the load on grid by advertising unrealistically low electricity prices and hence incentivizing immediate electricity consumption. This could create an imbalance in the grid.
2. Transactive control requires demand prediction signals/values from responsive assets, for best operation. The methods for predicting demand of responsive assets are lacking and accurate demand prediction will be difficult using state of the art. Authors have acknowledged this fact in [12].
3. Requiring demand signals/predictions from the responsive assets would put additional overhead on the devices and nodes. In order to satisfy this requirement extra compute and communication functionality might be required on these nodes.
4. Transactive control enforces a hierarchical structure for information/signal communication. It requires the communication and control within the smart grid to be aligned well with the flow of electrical power. This is a strict requirement on the communication for correct operation of the system. It would necessitate setting up virtual (or physical) communication channels aligned with the electric power flow on the grid.
5. There is a significant deployment/upgrade overhead involved in practically deploying the transactive control system on the electrical grid. Potentially thousands of 'nodes' need to be modified according to transactive control requirements. Virtual (or physical) communication channels need to be setup between the nodes in order to enable the hierarchical communication. Further, accurate demand prediction capabilities need to be added to the responsive assets on the grid.
6. If we remove the hierarchical communication requirement and the demand signal flow upstream, the proposed transactive control system essentially reduces to real-time price distribution protocol proposed in [10] (discounting security). As opposed to transactive control system the proposed protocol in [10] has no special communication requirements, or demand prediction needs at the responsive assets, and it can easily be scaled to large number of users by using existing IP multicast technology (along with publish-subscribe).

4. Real-Time Price distribution protocol for Ecogrid EU

[10] describes the data, security, and scalability requirements to a Real-Time market price forecast application layer message format.

4.1 Data Model

Real-Time price message which is broadcast to all the users has the following attributes:-

- The Real-Time market price
- The currency of the Real-Time market price
- The energy unit for the Real-Time market price
- The start and end time for which the Real-Time market price is valid (UTC)
- Forecast for the Real-Time price consisting of any number of forecast values comprising the following
  - Real-Time price forecast
  - The currency for the Real-Time price forecast
  - The energy unit for the Real-Time price forecast
  - The start and end time for which the Real-Time market price forecast is valid

The application level protocol is independent of the transport and session layer protocols, and above message is encoded in a XML structure.

4.2 Communication Pattern

In order to distribute real-time price information to a number of receivers authors in [10] propose use of either publish-subscribe protocols or broadcast/multicast mechanism of information exchange or a combination of both. Publish-subscribe protocols are event based protocols, hence network load due to continuous polling is reduced. It also allows the publisher to know the number of subscribers. An alternate way of communicating the prices would be to periodically broadcast the real-time price information to all nodes on network.

4.3 Transport Protocol

MQ (Message Queuing or Message Oriented Middleware), XMPP PubSub, SIP Subscribe Notify are some of the protocols which can be used for publish-subscribe mode of communication between a real-time market operator and the customers. UDP (User Datagram Protocol) using IP Multicast addressing is a very scalable way to broadcast data in an IP network which can be an alternative for broadcasting real-time prices to all the participating customers periodically. [10] proposes to combine IP multicast and publish-subscribe technologies to scale real-time price broadcast to millions of users. Here, a real-time market operator jointly works with an ISP (Internet Service Provider). The real-time market operator provides the real-time price signal to the ISP using a publish-subscribe technology and the ISP pushes the real-time market price to its customers using IP Multicast within its network.

4.4 Merits

1. The proposed approach of real-time price distribution can be scaled to a large number of

customers.
2. It provides the desired level of security for the price messages.
3. The application layer protocol is independent of transport and session layers.

4.5 Demerits

1. The current XML message format includes information only for the consumers of electricity and no pricing information for selling electricity back to the grid is provided. Homes with photovoltaic (PV) deployments or wind turbines may want to sell surplus electricity back to the grid using Net metering. Existing XSD can be easily extended to include the electricity selling prices.
2. Currently there is no discrimination between the end devices receiving price forecasts. For e.g., a controller with limited memory and compute resources will also receive the same price message as any data center server.

5. Features Comparison

In this section we summarize and compare the important features of all the surveyed protocols. Table 1 highlights the security, scalability, feasibility of using the protocol data model for real-time price exchange, transport protocol, and communication pattern of all the standards and protocols covered in this report.

Table 1 – Summary of important features

	Communication Pattern	Transport Protocol	Real-Time Price Information exchange	Scalability to Million customers	Security
IEC 61850 [3]	Client-Server	TCP, OSI	Yes	No	Yes
MMS [8]	Client-Server	TCP, OSI	Yes	No	No
CIM [6]	NA	Any	Yes	Implementation dependent	No
FlexPrice [11]	Client-Server	Any	Yes	No	No
Olympic Peninsula [9]	Client-Server Bidding	Any	Yes	No	No
Hierarchical Transactive Control [12]	Hierarchical Communication	Any	Yes	No (Lot of message overheads)	No
GOOSE [5]	Publish-subscribe on multicast or broadcast MAC addresses	ISO/IEC 8802-3 (Ethernet)	Yes	No	NA
Proposed price distribution protocol [10]	IP Multicast + Publish-subscribe	Any	Yes	Yes	Yes

6. References

[1] IEC – International Electrotechnical Commission: IEC 61970-301: Energy management system application program interface (EMS-API) – Part 301: Common Information Model (CIM) Base. International Electrotechnical Commission, 2003.

[2] IEC – International Electrotechnical Commission: IEC 61850-7-420: Communication networks and systems for power utility automation – Part 7-420: Basic communication structure – Distributed energy resources logical nodes. International Electrotechnical Commission, 2009.

[3] IEC 61850 Communication Networks and Systems In Substations, Technical Committee 57, International Electrotechnical Commission.

[4] IEC – International Electrotechnical Commission: IEC 61850-8-1: 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. International Electrotechnical Commission, 2004.

[5] Generic Substation Events. http://en.wikipedia.org/wiki/GOOSE, 2012.

[6] Common Information Model. (Computing) http://en.wikipedia.org/wiki/Common_Information_Model_(computing), 2012.

[7] IEC 61850. http://en.wikipedia.org/wiki/IEC_61850, 2012.

[8] Manufacturing Message Specification. http://en.wikipedia.org/wiki/Manufacturing_Message_Specification, 2012.

[9] D.J. Hammerstrom et al., "Pacific Northwest GridWise Testbed Demonstration Projects: Part 1. Olympic Peninsula Project", Pacific Northwest National Laboratory, Richland, WA, Tech. Report PNNL-17167, October 2007.

[10] Input on the Real-Time Price distribution protocol for Ecogrid EU WP 3 Task 3.6. http://www.zurich.ibm.com/pdf/ecogrid/Input_on_a_price_distribution_protocol.pdf, 27 May 2012.

[11] "FlexPrice – The definition: Definition of FlexPrice, the container for Control-by-price signals". Internal draft from Eurisco, 2012.

[12] D. Hammerstrom, T. Oliver, R. Melton, and R. Ambrosio, "Standardization of a Hierarchical Transactive Control System", Grid Interop '09 Conference, 11/09.

[13] Overview and Introduction to the Manufacturing Message Specification (MMS). http://www.sisconet.com/downloads/mmsovrlg.pdf, 2012.

[14] IEC – International Electrotechnical Commission: IEC 61850-7-2: Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI). International

Electrotechnical Commission, 2010.

[15] J.T. Sorensen, M.G. Jaatun. An Analysis of the Manufacturing Messaging Specification Protocol. UIC '08 Proceedings of the 5th international conference on Ubiquitous Intelligence and Computing, 2008.

[16] Y. Liang, R.H. Campbell. Understanding and Simulating the IEC 61850 Standard. Technical Report UIUCDCS-R-2008-2967, University of Illinois at Urbana-Champaign, May 2008.

[17] IEC 62351. http://en.wikipedia.org/wiki/IEC_62351, July 2012.