To appear in *Proceedings of Power Systems and Communications Infrastructures for the Future*, International Institute for Critical Infrastructures, Beijing, September 2002.

Towards More Extensible and Resilient Real-Time Information Dissemination for the Electric Power Grid

David Bakken, Anjan Bose, Carl Hauser,
Ioanna Dionysiou, Harald Gjermundrød, Lin Xu, Sudipto Bhowmik
School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752 USA
{bakken,bose,hauser,idionysi,kgjermun,lxu2,sbhowmik}@eecs.wsu.edu

1 Introduction

The availability and integrity of the electric power grid heavily depends on maintaining a very close balance between its supply and demand. To achieve that balance, grid operators rely on real-time status information from grid entities such as generators and substations. Failing to match supply to demand accurately and in real-time can result in partial or complete breakdown of the grid [MR99]. Thus the information infrastructure is a critical component for reliable grid operations.

The power grid today has a very basic, hardwired data acquisition infrastructure suited for the monopoly in generating, transmitting and managing power that was present at design time (see Figure 1). Its status dissemination network was designed over 30 years ago and although the hardware has been upgraded over time, the overall design is still the same. This infrastructure cannot support status information dissemination to the many different participants, created as a result of the recent deregulation of the grid. Moreover, the present data rates are slow because of the antiquated design. A new information infrastructure is proposed to accommodate the new diversity and services in the power grid. Delivering real-time and historical data to the various interested parties must be done reliably and fast without jeopardizing the grid itself.

This paper presents **GridStat**, flexible and manageable status dissemination middleware for the electric power grid. To meet the new requirements, GridStat will use distributed systems technologies and protocols to securely deliver timely and accurate information to the participants who need it. Flexibility is required in order to dynamically add and remove grid participants and to deliver different sets of events to different participants in different circumstances. The status dissemination service's manageability features are required in order to ensure its reliable operation in the face of changes. To ensure security of grid operations (in the power sense) the status dissemination service must provide the computer and network security features such as message authentication, data integrity and confidentiality. The terms *secure* and *security* will refer to these concepts throughout this paper.

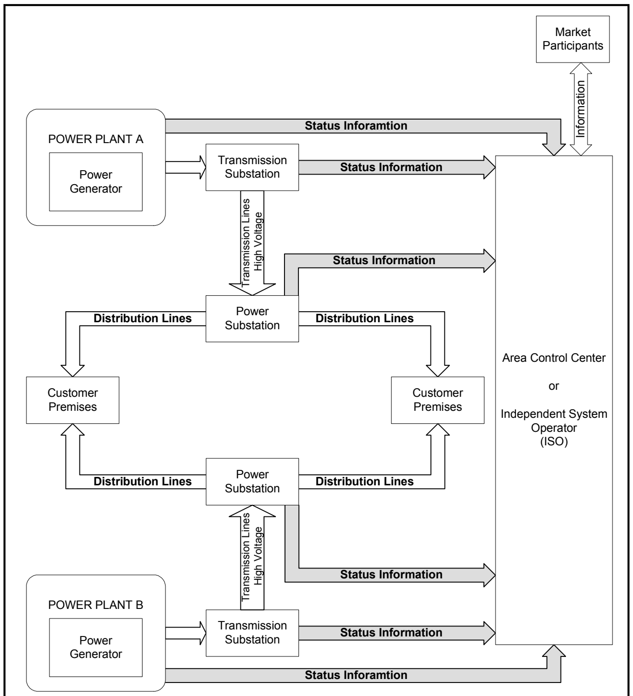

Figure 1: Power Grid Information Infrastructure Example

GridStat provides a publisher-subscriber model allowing producers of data (publishers) to post their information without the consumers (subscribers) being on-line during the posting operation. Consumers are able to subscribe to given types of events, and receive them whenever they are available. Intermediate entities temporarily store the data and distribute it to the interested subscribers. GridStat delivers Quality of Service (QoS) properties (such as redundancy and fault tolerance) as required by grid monitor and control applications. More details on the publisher-subscriber nature of GridStat and QoS properties are given later in the paper.

The rest of the paper is organized as follows: Section 2 explains the motivation behind status dissemination. Section 3 gives an overview of GridStat and its main components. Examples of GridStat and its support of QoS properties are presented in Section 4. Section 5 discusses the current status of GridStat and further intentions for future work. Related work is described in Section 6. Conclusions are outlined in Section 7.

2 Motivation for Status Dissemination Service

The current architecture of the grid was based on the existence of vertically integrated utilities, which controlled the entire power operation within a geographical area. The current grid operational system with its fixed, hardwired, sequential data acquisition infrastructure is inadequate for fulfilling today's challenges in the power industry. It therefore needs to be extended in order to provide the flexibility, manageability, and security required for efficient and practical communication of grid status and control information [NERC].

The present infrastructure does not provide any information flow other than to the central control centers or ISOs, which exclusively manage the majority of the huge amount of data received from substations and other grid entities. Additionally, technological advances in microprocessor-based secondary equipment have given rise to the use of intelligent electronic devices (IED) installed at substations, making even more status information available. The present design of collecting all data at a central computer is no longer suited for collecting this large amount of data at high rates.

As mentioned earlier, recent advances and regulatory reforms in the electric power industry promote competition by allowing independent power producers and transmission companies to enter this previously monopolistic industry. The new services that are emerging, such as establishment of bilateral contracts, require status information dissemination to a wider set of parties including power producers, individual consumers, and traders. Status information is currently unavailable to these parties due to lack of flexible communications. Yet information flow between entities within the grid is essential for stable, reliable operations.

The information system was also designed to handle a limited number, type and geographic proximity of failures. Because of this, failures at one of many points within the grid have the potential to severely degrade the communication necessary for its control. Further, electric power is currently at risk not only because of the possibility of unintentional failures, but also due to deliberate cyber attacks [OSF00].

Therefore, what is needed is a status service that will distribute information to legitimate parties in a timely, secure and accurate manner. Bandwidth reservation technology along with distributed real time computing can be used as the basic building tools for implementing a new grid communication infrastructure. The new infrastructure must be capable of delivering data to any registered entity according to each's QoS specifications.

3 GridStat: Status Dissemination Middleware

The heterogeneity of typical distributed computer systems is also evident in the evolving power grid. Different vendors of IEDs have incompatible communication protocols. Substations use applications implemented in a number of different programming languages and on different platforms. Future consumers of the data, other than the ISOs, will run their own applications on an environment provided by their own machines. In order to solve the problems caused by heterogeneity in platforms, programming languages and networks, middleware such as CORBA [VR01] can be used to mask the heterogeneity.

Middleware is a class of software technologies that provide common abstractions across a distributed system [Bak02]. GridStat's abstractions are publish-subscribe communications and middleware QoS management. Currently, there is no existing middleware with these attributes.

The publish-subscribe model is built on the principles of space, time and flow decoupling. The interacting entities, publishers and subscribers, do not need to know each other, or actively participate in the interaction at the same time. Moreover, neither entity is blocked while producing or consuming events. The idea behind this model is rather simple. Publishers post events and subscribers consume those events. Timely and reliable event delivery needs to be accomplished to all registered consumers, thus the importance of QoS management. During subscription, subscribers can specify the events that are of special interest to them using event properties and their values. For example, a substation can take the role of a publisher by publishing its current level of voltage every 5 seconds. Interested parties such as traders might want to know the substation voltage only when it reaches a threshold value. They therefore subscribe to the event posted by the substation with the additional constraint of voltage value. They will only receive those events that contain a voltage value that meets their specified requirements.

3.1 GridStat components

The main GridStat entities are categorized into active and passive entities. The active category includes publishers, subscribers and QoS managers. QoS policies and events are the passive entities.

The entire GridStat architecture heavily relies on message communication. Each message captures a single event, which is expressed in an event language. Any event generated by either publishers (original/simple event) or QoS managers (event constructed from existing events) has static properties such as name, owner, value and security information. In addition to these, an event has dynamic attributes that are time-related such as moving average, rate of change, moving average of the rate of change, and minimum and maximum values over a time interval. An event is also called status item in GridStat terminology. Sometimes subscribers read a large set of status items once to calculate a derived status item. This task is accomplished by user-defined condensation functions loaded in GridStat QoS managers.

One of the goals of GridStat architecture is to give tools to grid operation engineers to create common status items. For example, a failure detector or circuit breaker can have associated status items that are either true or false. The voltage of a substation is a periodic status item sent in a regular pattern (e.g. every 5 seconds). A potentially catastrophic situation might cause the immediate distribution of an alert status item. Grid engineers must be able to define and publish status items for these entities as the grid evolves.

QoS policies are the other passive entities. QoS policies are developed by QoS managers from requirements and specifications captured by the active entities. QoS properties include, among others, fault tolerance and timeliness. A description of properties supported by GridStat is given in the next section.

Publishers, subscribers and QoS managers are the active entities that initiate communication and status flow within GridStat. A publisher, the first end entity of the publish-subscribe model, publishes status items regarding transmitters, breakers, etc. and asserts QoS requirements for their delivery. For instance, a substation may publish all of its breakers' voltage every 1 minute. The other end entity, the subscriber, consumes the events generated by publishers and also asserts QoS requirements.

The last type of active entity, the QoS manager, is responsible for resolving conflicting QoS requirements and conflicts between QoS requirements and the network capabilities in order to develop QoS policies. A QoS manager tries to implement requests as set by subscribers, publishers or other higher-level QoS managers and deliver status items from the publishers to the subscribers according to the QoS policy. Control decisions are made at this node. The QoS manager can adjust the information flow according to time scales that correspond to slow, medium or fast responses. The QoS managers comprise a distributed system with the responsibility for adjusting QoS attributes of GridStat information flow in order to ensure stable grid operations. QoS managers are either edge managers or internal managers depending on their position. An edge manager is one that publisher and subscriber entities connect to. Internal managers accommodate the internal workings of GridStat without being visible by either publishers or subscribers.

One of the important characteristics of the GridStat model is that it gives the ability to its subscribers and publishers to remain anonymous throughout the entire status transmission process. The subscriber does not need to know the identity of the publisher in order to request the delivery of a status item. By the same token, the publisher is not concerned with which entities receive the status that it publishes. The only connection either entity has is with its local edge QoS manager. This node is responsible for locating the appropriate entities through possibly other internal QoS managers, and delivering QoS properties. In order to provide consistent naming, QoS managers contain a naming service, which will be responsible for assigning names to GridStat participants. This naming service will have the same hierarchical structure as the Domain Name Service (DNS), the naming authority of the Internet [RFC87].

3.2 QoS Specifications

One of the essential and novel features of the GridStat architecture is the ability to embed QoS properties within the middleware. In order to guarantee timely transmission and delivery of an event, the appropriate resources need to be reserved before publishing the event.

Some of the QoS properties that will be supported as part of the GridStat model are the ones below:

- Fault Tolerance: can be accomplished through redundancy in space and/or time depending on the priority of the event. Packet transmission will involve the use of redundant paths reserved prior to the transmission in order to provide fault tolerance. In addition to space redundancy, event retransmission can be employed for time redundancy.
- Delivery Timeliness: timeliness is meeting a requirement on the delay between transmitting data by the publisher and receiving it by the subscriber. The assumption made, at least for the power grid, is that each active entity will have an accurate time source available (for example a GPS receiver). Time consistency is maintained throughout GridStat. Timeliness should take into consideration, among other factors, the link delays and the processing required at each node.
- Publisher Rate: the publisher sets the rate at which an event will be published. If the rate is greater than the delivery rate, some packets might need to be dropped.
- Delivery Rate: the subscriber sets its own rate at which it requires delivery of packets. If the delivery is greater than the publisher rate, then it might be impossible to satisfy this requirement.
- Priority: each event should have a priority value assigned to it depending on its type. The QoS manager can choose the priorities for events before further distribution. Packets with higher priority are placed first in the event channel queue. Some implications might be starvation and lack of timeliness that arise from the fact that packets with lower priority may expire while waiting higher priority packets to be served.

The complete QoS specification is given in table 1.

Who sets what	**Publisher**	**Subscriber**	**QoS Manager**
Redundancy		√	√
Delivery Rate		√	√
Delivery Timeliness		√	√
Publisher Rate	√		
Priority	√		√
Confidentiality		√	√
Integrity		√	√
Access Control			√

Table 1: QoS Specifications

3.3 GridStat Perspectives

Information dissemination for the deregulated grid is solved by the GridStat architecture. The power grid with GridStat can be looked at from three different perspectives as outlined below:

- Physical Information and power flow: this is the actual flow of information based on the physical structure and geographical topologies.

- Hierarchy of QoS controls: depending on geographical and business criteria, a QoS hierarchy is constructed responsible for maintaining QoS properties and grid policies.
- Publish-subscribe logical view: this is a graph showing how the information is delivered from the publisher to the subscriber.

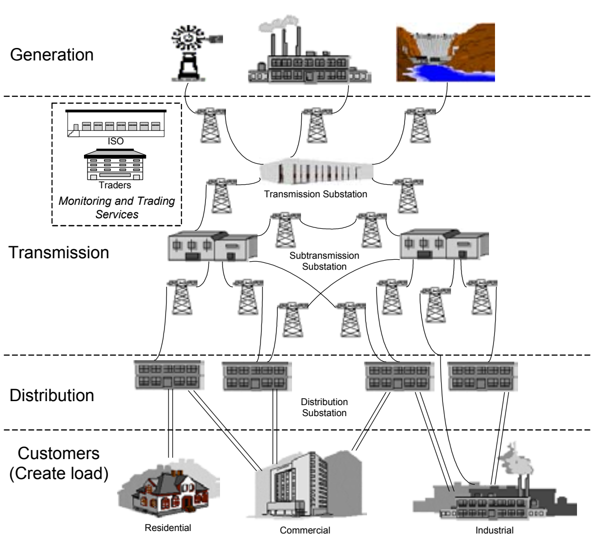

Figure 2: Power Grid Infrastructure Components

Consider Figure 2, which illustrates the major components of the power grid infrastructure as it is today. Currently all the status information collected by the substations is gathered at the ISO (see Figure 1), which performs off-line analysis to detect irregularities within data. The substation sends its current voltage level every say, 5 seconds, its daily activity once a day, its extra capacity every 5 minutes and alerts whenever lines are down. There is minimal communication between substations. Even if there is a direct data link, it doesn't get utilized efficiently: the majority of the communication goes through the ISO. Therefore, if the interconnecting line between two substations is experiencing any problems, the ISO will notify both substations rather than having the two involved entities directly notifying each other.

One aspect of deregulation is to be able to establish bilateral contracts between generators and customers (one-to-one contracts) without having any other intermediate entities getting involved (the ISO however oversees the feasibility of the transaction) [BG98, Gre96]. For example, the customer (a factory) may send to a trader its load needs every 4 seconds that will be matched by the contacted trader as part of the deal. The trader communicates the load needs to a generator. At the same time, the trader wants to get historical data regarding the power demands daily from a substation, so that it can adjust accordingly the prices it will negotiate with power generator representatives.

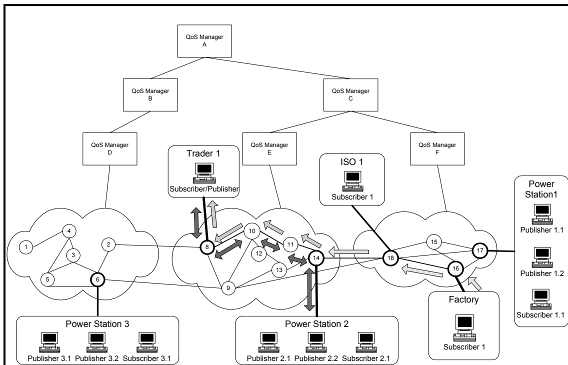

Figure 3: Publish-Subscribe with Hierarchical QoS Managers

The information flow is depicted in Figure 3 from the hierarchical viewpoint. Each entity in the grid is operated under some QoS manager, which allocates GridStat resources and manages the information network assigned to it with the sole goal of delivering the QoS required by the end entities (publishers, subscribers). The factory is a publisher. The power station and the trader take both roles. Note that both the trader and the power station 2 belong to the same QoS manager E and the factory subscriber resides under QoS manager F. Edge Managers are represented by bold circles, where the remaining nodes are internal QoS managers. Each cloud represents the network portion that is controlled by a QoS manager, for example QoS manager D controls the leftmost cloud. Communication between networks is achieved through connecting nodes.

The final perspective is the logical view of the entire publish-subscribe model, ignoring the physical and hierarchical structure. The aim is to deliver the status information from the publishers to the subscribers in a timely manner. Figure 4 illustrates the scenarios described earlier. The importance of this logical view lies in the fact that only the edge Managers are

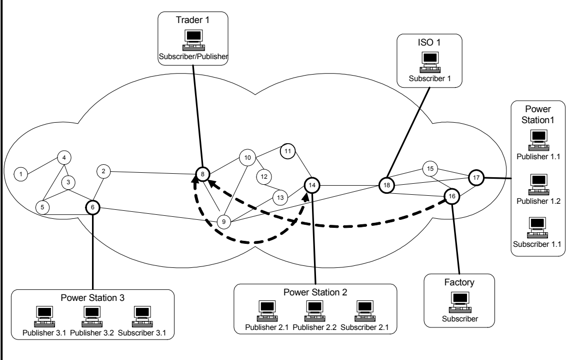

Figure 4: Logical Publish-Subscribe model

visible, leaving all the internal workings of the graph transparent to either the publisher or the subscriber.

The publisher is only aware of its QoS manager and it puts trust on the entire grid structure for prompt delivery of its posted status information. The subscriber also relies on its own edge QoS manager for receiving the status information that it registered for.

4 GridStat: Examples

This section illustrates some of the internal workings of GridStat for a simple realistic scenario. Suppose that the entities trader 1 and trader 2 want to receive a status item from the Power Station 2 that supplies both traders with power. They want to receive this status message containing the actual power supplied every 2 seconds with a desired timeliness of 1 second but can tolerate a 1.5 second delay in the worst case. A sample graph is shown in Figure 5. All the links have two numbers associated with them; the first one indicates the link delay for transmitting the packet and the second one shows the delay when encryption for confidentiality and authentication is employed. The link-to-link encryption/decryption adds to the total delay of the link.

Let's suppose that the substation can publish the status every 1 second. The question to be answered is whether GridStat can deliver the status information with the desired properties.

Case 1: Timeliness property

Suppose that the only property than needs to be satisfied is the timeliness property. Trader 1 will register with its local QoS manager (node 8). QoS manager 8 will try to locate a path to the entity named Power Station 2 with the desired properties. Routing algorithms will be employed for discovering the path (or paths) and computing the delays, which will determine whether the request is feasible, or not. In this case, the path <8, 10, 11, 14> is the optimal path with path delay of 0.4 seconds. In case that the link between node 8 and node 10 is down, the path <8, 9,10,11, 14> can be taken with path delay of 0.5 seconds. Similarly, a path from Trader 2 to Power Station 2 will be determined (<11,14>).

Case 2: Timeliness and redundancy

In order to assure delivery, both subscribers require that the publisher send the packets with redundancy of 2. The QoS manager connected to the substation (node 14) decides that the packet should traverse <8, 10, 11, 14> and <8, 9, 13,14 > as the two optimum paths to Trader 1. If node 9 becomes unavailable, the packet can still be sent through the first route but a replacement is needed for the second route. The only one left that meets the desired timeliness is <8, 10, 12, 13, 14>. As far as Trader 2 is concerned, the packet will traverse <11,14> and <11,10,12,13,14>.

Case 3: Timeliness, redundancy and security

This is a demanding case where 3 QoS properties need to be fulfilled. There is only one path (for either Trader) that satisfies the timeliness requirement with respect to the link costs when security operations are required. But no other redundant path can be found providing secure and timely delivery of the status to the subscribers. This is the case that shows that there are trade offs among the various QoS dimensions. Maybe sometimes the application can sacrifice security in order to gain redundancy. Managing conflicting QoS requirements so as to deliver the most important is the task of the distributed QoS managers.

5 Status and Future Work

We are currently designing and developing the middleware framework, GridStat, to deliver status information for the power grid in a fashion conducive to its survivability. The low-level middleware mechanisms are being implemented. Furthermore, we are conducting simulation studies of the distributed information structure of the power system. The goal of the study is to discover how the real-time requirements of the grid can be met by distributed system architectures. Both network delays and processing delays at nodes are taken into consideration. We have also started collaborating with CERT Coordination Center at CMU to integrate Easel simulation language at the middleware layer, which will be used to simulate GridStat with large numbers of participants.

We are also creating a prototype system to integrate power grid data published using the Extensible Markup Language (XML) [HRP+01]. XML is quite capable of describing irregular data so in order to provide better service for consumers, we are developing a query language called ApproXPath [Xu02] that handles irregular data by supporting inexact (as well as exact) searches.

One of the important QoS properties is security and access control. Each application might want to deploy mechanisms providing different security services. Therefore, the application designers must be able to decide which particular security services (authorization, integrity, authentication, availability, confidentiality) are needed for their particular case. GridStat, as we are building it, contains a security services framework. The population of this framework with security policies that match the needs of grid operations (who can subscribe to what, where can information be routed, what confidentiality is required for particular alerts) remains for the future. Likewise GridStat's QoS framework allows grid operations to express timeliness constraints and know when they can be met. But what those constraints should remain for future work.

6 Related Work

Several research efforts are exploring ideas similar to GridStat. PASS (Piece-wise Asynchronous Sample Service) [ZOB+99] is a service that provides information about the status of communication network resources, indicating their availability or not. The system is neither manageable nor applicable to the power grid in any way. It does not support status patterns either. PASS has been fielded in military exercises.

Sienna is an Internet-scale event notification middleware framework [CRW00]. Sienna is implemented as a distributed network of servers that provide clients with access points offering a publish-subscribe interface. It is a best-effort service and therefore does not support any QoS properties. But the main difference between Sienna and GridStat is that the former is entirely based on events and doesn't try to take advantage of more specialized semantics of status items like the latter.

The Spinglass project [BRV01] at Cornell University is a gossip-based probabilistic atomic broadcast protocol. One of the optimization tools in the Spinglass project called gravitational gossip can be used as a mechanism for publish-subscribe middleware. It provides security, reliability, and a high probability of message delivery of a specified amount of messages. Gravitational gossip pushes all the complexity to the multicast layer, which reduces flexibility of its applicable areas. It also lacks the ability to specify and reserve QoS properties such as timelines and delivery rate.

Xie et al. have recently proposed a new information architecture for power systems that provides direct communication between grid entities and addresses the need for redundancy in these facilities [XMV+02]. GridStat middleware provides the tools needed to build robust, flexible, manageable and secure distributed status dissemination software over the low-level communication infrastructure such as proposed there.

7 Conclusions

Recent deregulation is causing major changes in the current power grid infrastructure. One of the changes is accommodating the status information needs of the increasing number of participants in the grid operation. New services and optimizations require more quantity, timeliness and diversity of the information flow than is currently being provided. GridStat status dissemination middleware will provide solutions in a flexible, manageable and secure manner.

The simple interface of GridStat is advantageous because grid operation engineers can utilize GridStat in power system applications without any further assistance. All internal workings of GridStat are transparent to the applications and application programmers. GridStat active entities can communicate status information with one another while relying on GridStat to manage the Quality of Service.

Finally, we anticipate that GridStat will be used in other critical infrastructures and applications besides the electrical power grid. The stock market, military and e-commerce are some areas where GridStat can be applied.

8 Acknowledgments

This work was funded in part by the US Dept. of Commerce, National Institute of Standards and Technology Grant #60NANB1D0016 (Critical

Infrastructure Protection program), in a subcontract to Schweitzer Engineering Labs Inc.

9 References

[Bak02] Bakken, David, "Middleware", Chapter in *Encyclopedia of Distributed Computing,* J. Urban and P. Dasgupta, eds., Kluwer Academic Publishers, 2002, to appear. See

http://www.eecs.wsu.edu/~bakken/middleware.pdf.

[BG98] Bakken, B.H. and Grande, O.S. "Automatic Generation Control in a Deregulated Power System", *IEEE Trans. on Power Systems*, vol.13, no.4, November 1998, pp. 1401-1406.

[BRV01] Birman, Kenneth P., Renesse, Robbert, and Vogels, Werner. "Spinglass: Secure and Scalable Communications Tools for Mission-Critical Computing", *International Survivability Conference and Exposition. DARPA DISCEX-2001*, Anaheim, California, June 2001.

[Gre96] Green, R.K. "Transformed Automatic Generation Control", *IEEE Trans. on Power Systems*, vol.11, n.4, November 1996, pp.1799-1804.

[CRW00] Carzaniga, Antonio, Rosenblum, David, and Wolf, Alexander. "Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service", *In Proceedings of the Nineteenth ACM Symposium on Principles of Distributed Computing (PODC2000)*, Portland, OR, July 2000.

[HRP+01] Hunter, D., Rafter, J., Pinnock, J., Dix, C., Cagle, K., Kovack, R., *Beginning XML*. Wrox Press Inc, Chicago, 2001.

[MR99] Mountford, John and Ricardo Austria. "Keeping the lights on", *IEEE Spectrum*, June 1999, p. 34-39.

[NERC] North American Electric Reliability Council, NERC Operating Manual, Policy 10. See http://www.nerc.com.

[RFC87] Mockapetris, P. " Domain Names – Implementation and Specification", RFC-1035, ISI, November 1987.

[OSF00] Oman, Paul, Schweitzer III, Edmund O. and Frinke, Deborah. "Concerns about Intrusions into Remotely Accessible Substation Controllers and SCADA Systems", Schweitzer Engineering Laboratories, WA, USA, 2000, http://www.selinc.com/techpprs/6111.pdf.

[VR01] Verissimo, Paulo and Rodrigues, Luis. *Distributed Systems For System Architects*. Kluwer Academic Publishers, 2001.

[Xu02] Xu, Lin. "ApproXPath: An Approximate Tree Matcher for XML", Masters' of Science Thesis, Washington State University, May, 2002.

[XMV+02] Xie, Z., Manimaran, G., Vittal, V., Phadke, A.G., Centeno, V. "An Information Architecture for Future Power Systems and Its Reliability Analysis", *IEEE Trans. On Power Systems*, to appear, 2002.

[ZOB+99] Zinky, John, O'Brien, Linsey, Bakken, David, and Krishnaswamy, Vijay, Ahamad, Mustaque. "PASS: A Service for Efficient Large Scale Dissemination of Time Varying Data Using CORBA", *in Proceedings of the Nineteenth International Conference on Distributed Computing Systems (ICDCS '99*), IEEE, Austin, Texas, May 31-June 4, 1999.