LANSIM and Its Applications to Distributed EMS

Shau-Ming Lun* **Tetiana Lo*** **Felix Wu*** **Liam Murphy*** **Amitava Sen+**

* Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, California

+ ABB Systems Control
Santa Clara, California

Abstract - Distributed energy management systems (EMS) open up a host of alternative design options. Simulation plays an important role in evaluating performance and in comparing alternative designs. Currently all the proposed distributed EMSs are local area network (LAN)-based. A LAN simulator, LANSIM, has been developed. To illustrate the application of LANSIM, comparisons are made with different distributed EMS configurations, different LAN technologies (Ethernet and FDDI), and different Ethernet implementations.

Keywords : Energy management systems, control centers, distributed systems, open system architecture, simulation, local area network.

1. Introduction

Recent advances in computer and communications technologies have increased the options available to energy management system designers. These advances include increasingly powerful workstations and communication links operating at higher and higher data transmission rates. This has led to the development of communication network protocols supporting high-speed data transmission between different types of workstations. The net result of these advances is that it is now technologically feasible to distribute the processing load of an EMS among several processors, leading to the concept of a distributed EMS. The trend in system architecture is towards an open system[1], one that can be replaced, in part or in its entirety, without relying on one vendor for the complete implementation. A distributed EMS with modular components and standardized interfaces is easy to implement as an open system.

Several distributed EMS configurations have been reported recently. In [2] a LAN-based EMS was simulated to evaluate the performance of the distributed system. A functionally distributed system was presented in [3], in which different EMS functions were implemented in separate functional units connected to each other and the input/output unit by a dual LAN. Currently all the proposed distributed EMS designs are LAN-based (e.g. [4], [5]).

This paper was presented at the 1993 IEEE Power Industry Computer Application Conference held in Scottsdale, Arizona, May 4–7, 1993.

In the design of a distributed EMS, there is a need to know where the relevant technologies are heading, and what is expected to be available in the near future. Among the trends, for example, are improvements in distributed operating systems, distributed databases, and new fiber-based communication networks and protocols. To be able to take advantage of these improvements, a flexible EMS design is required, so that the EMS can evolve as they become available.

The rate at which these advances are occurring is perhaps as important as the advances themselves. Performance in the microprocessor, communications and computer areas has improved dramatically in the last 5--10 years, so that the combined effect of these innovations is revolutionary. The static design process used until now, involving an expected 10 or 15-year EMS lifetime and a single vendor, relies on a stable technology base and well-defined performance requirements. However, the state-of-the art is changing rapidly, and the range of knowledge necessary to consider all options in EMS design is now much wider than even a few years ago. This poses a major problem for EMS designers, who face a much more unpredictable technology future and a much more complex set of design issues than before. A new, dynamic approach to EMS design is needed[6].

A flexible approach to EMS design should be capable of evaluating the impact of design component alternatives on overall system performance. It should also allow the modeling of future configurations to which the system could evolve, to ensure that the structure can support necessary or desirable changes. If the EMS already exists, it should be possible to determine the system's evolution path, by evaluating the effects of proposed modifications on system structure.

Since prototyping is not possible for considering a great many options in advance, and no analytic model exists for such a complex system as a distributed EMS, simulation provides the only alternative for evaluating performance and comparing designs. The widespread availability of powerful workstations means that simulations run faster, and larger-sized problems can be tackled.

An EMS simulator should be designed to be (i) modular, (ii) flexible, and (iii) extendable. It should be easy to modify or replace a module. By having a library of alternatives for each module, the simulator can explore a wide range of designs with minimal effort. Modules should be easily added or replaced, as new technology for EMSs becomes available.

An object-oriented design (OOD) approach [7] provides these properties when used as the underlying design philosophy for an EMS simulator. In an OOD the various modules are defined as classes, and specific design choices are then instances of their class. The use of inheritance minimizes the work required to extend functionality of the modules. Information hiding, the concept of abstracting internal details of other objects, provides the necessary modularity and flexibility.

At University of California, Berkeley (UCB), an integrated planning and analysis environment for communication networks, called NetPlan, has been developed[8]. NetPlan is based on object-oriented technology; tools and network elements are defined as classes of object. Users can easily create new tools and add new equipment models through inheritance. Tools have been developed to simulate wide-area and local-area networks. LANSIM, a local area network performance simulation tool in NetPlan, can be used to evaluate the real-time performance of a LAN-based distributed EMS. To illustrate the use of LANSIM in this performance evaluation, we study an example of the type of distributed EMS currently under discussion in the literature. This consists of a small number of processors connected to each other and to the dispatcher consoles by a LAN. We examine two present-day alternatives for the LAN : Ethernet and FDDI. Our study configurations will be discussed in more detail in Section 4.

With these configurations and LAN technologies, we investigate LAN performance under heavily-loaded conditions, to observe if the EMS is still capable of meeting its real-time requirements. We also investigate the effect on performance of distributing the processing further by running state estimation in a separate dedicated processor.

In addition to the results we present for this study, we show by example the ease of examining various cases using LANSIM , and how a new technology or configuration can be incorporated without difficulty into the simulation model. This flexibility demonstrates the power of our simulation approach, not just in this specific case, but in general.

2. Local Area Network

Two LAN protocols are examined in this paper: Ethernet and FDDI.

Ethernet is a multi-access broadcast communication system operating at 10 Mbps. Packet transmissions are coordinated through a statistical arbitration procedure, and there are no preallocations of time slots or frequency bands. Ethernet uses the CSMA-CD protocol[9], whereby a node must wait for the channel to become idle before transmitting and must stop transmitting once a collision is detected. Any node that detects a collision invokes a collision consensus enforcement mechanism that briefly jams the channel. Nodes involved in the collision reschedule transmission to take place after a random delay period computed using the binary exponential backoff algorithm (BEBA).

FDDI is a 100 Mbps network having an optical fiber dual ring topology and utilizing a timed token rotation protocol[10]. A node gains access to the ring to begin transmission by capturing the token, a unique frame which is circulated sequentially. The token is forwarded immediately after completing transmission. Each node repeats the frames it receives to its downstream neighbor, and when the frames return to the originating node, it is stripped by not being retransmitted. FDDI supports two modes of transmission: synchronous and asynchronous. Synchronous traffic has a preallocated maximum bandwidth and a guaranteed minimum response time. This traffic can be transmitted by a node whenever it receives a token. FDDI provides eight asynchronous priority levels. Asynchronous traffic bandwidth is allocated dynamically from the remaining bandwidth. Asynchronous transmission is not allowed if the time since receiving the last token exceeds the operating target token rotation time T_Opr, a negotiated value representing the system's shortest time interval between successive receptions of the token by a node.

3. The LANSIM simulator

3.1. Ptolemy

LANSIM is based on the Ptolemy software environment, an object-oriented system developed at UCB[11]. Programmed in C++ and running on a UNIX workstation, Ptolemy is a very flexible framework, supporting heterogeneous system specification, simulation, and design.Each model of computation is called a *domain* and consists of an extensible library of functional blocks. The basic unit of computation in Ptolemy is the block, represented graphically by an icon with terminals corresponding to its input/output *portholes*. A block may be atomic(*star*) or composite(*galaxy*). Applications are constructed graphically by connecting blocks. At runtime, a scheduler invokes the blocks; data is exchanged in the form of *particles*, discrete units which may be of several types: integer, real, complex, or a general structure such as a data packet. Blocks may have *states*, user-settable data structures which may be monitored from one execution to another. Ptolemy's strength and uniqueness lie in its capability of supporting the many aspects of communication system modeling and simulation and the ease with which new application-specific design environments may be built.

3.2. DE Domain

Individual nodes of the Ethernet and FDDI networks are modeled as stars in the discrete-event(DE) domain of Ptolemy. DE stars function as event-processing units which receive and process particles from the outside and generate output events after a user-given latency. Serial and ring interconnections of stars constitute the respective local area networks. One type of particle is defined as a data packet with an associated time stamp generated by the block producing the particle and represents an event corresponding to a change of system state. The DE

scheduler processes events in chronological order until the global time reaches a user-specified "stop time." A star is executed whenever a new event appears at the input portholes. After execution, output events may be generated at its output portholes i.e. packets to be transmitted to another station.

3.3. Ethernet Model

Each node is modeled as possessing two input and two output portholes for receiving frames from and broadcasting in both directions. Initially, the nodes are assumed idle, and the Ether quiet. A node wishing to transmit a packet passes a start-of-transmission (SOT) event onto the channel. Generation of the SOT event, announcing the beginning of transmission and that the Ether is busy, takes place when either a node receives a data packet from the upper layer or when a retransmission due to a collision is scheduled to occur. The node is now in the transmission state. While in this state, if the node receives another SOT event, a collision has occurred; the node moves to Jam mode, sending jam packets and rescheduling transmission to take place after a random delay computed using the BEBA. Otherwise, if no collision occurs, an end-of-transmission(EOT) event is sent, signaling the successful transmission of a packet, followed by the data packet obtained from the queue. The node then moves to the wait state, waiting the minimum packet spacing time before transmitting a SOT event again. The protocol model is represented by the flow diagram below:

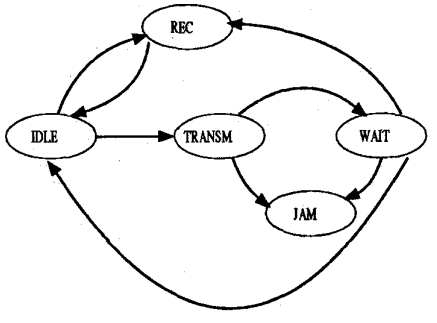

Figure 1 State Diagram of Ethernet Model

3.4. FDDI Model

A single FDDI ring is modeled based on the ANSI Standard for FDDI[12]. Initially, it is assumed that an arbitrarily chosen node possesses the token. The user assigns the synchronous bandwidth and eight asynchronous priority thresholds for each node. During the first token rotation, all node late count registers(LCTs) are set to 1; only synchronous data may be transmitted. Thereafter, normal ring operation ensues. The star may be thought of as possessing four states of operation. In the idle state, the LCT value is 0; the node is awaiting the token and possibly processing packets that appear from the ring. When the token arrives, the current value of the token rotation timer (TRT) is placed in the token holding timer(THT); TRT is reset by T_Opr, and synchronous traffic may be transmitted. After the synchronous bandwidth timer or TRT has expired, the node may transmit asynchronous packets for the duration of THT, forwarding the token afterwards and returning to the idle state. While in the idle state, if the TRT expires, LCT is incremented, and the node enters another state. Synchronous data is transmitted upon receiving the token; the token released, and the node returns to the idle state. Expiration of the end-of-transmission timer signals the completed transmission of a packet. The node checks the status of the timers and queue for the availability of packets for continued transmission.

It should be pointed out that in our models the transmission media and nodes are assumed error-free; packets are transmitted and arrive at destinations without error. Furthermore, it is assumed that no node failures, Ether or ring malfunctions occur, and the networks operate at fixed transmission rates: 10 Mbps for Ethernet and 100 Mbps for FDDI.

3.5. Station Model

A workstation model in LANSIM is composed of five types of blocks: LAN protocol, application, traffic generator, receiver, and report generator, which are interconnected as shown in Figure 2. For each type of block, a class library containing a wide variety of commonly used blocks is supported by LANSIM.

A LAN protocol block models a LAN technology, such as Ethernet or FDDI. Application blocks contain the traffic models of EMS functions, such as Economic Dispatch (ED) or State Estimation (SE). The traffic model specifies the probability distributions of the packet generation rate and packet length. The traffic generator in each station generates packets according to the traffic models in the adjoining application blocks. In order to investigate the effects of packet length on system performance, buffering and packet fragmentation are also modeled in the traffic generator. The receiver accumulates and processes packets produced by all traffic generators and passes them to the report generator blocks, which perform statistical analysis on the traffic passing through the station. Examples of functions provided by the report generator block include calculating average traffic delays and displaying packet length distributions.

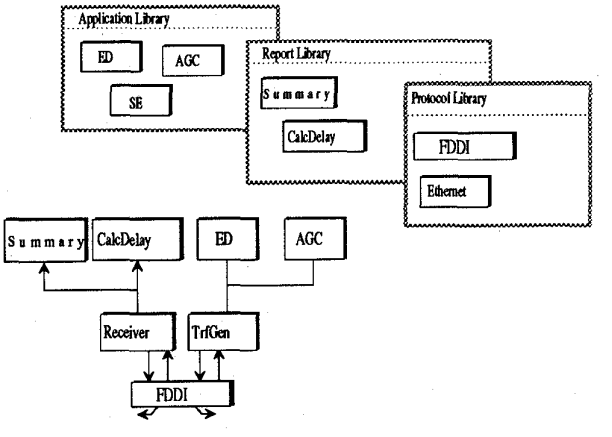

Figure 2 Station Model

3.6. Simulation

LANSIM is capable of simulating the dynamic behavior of various LAN-based EMS architectures. A simulation case may be constructed based on the particular configuration of the LAN-based EMS we wish to simulate. This includes specifying the number of workstations, functions distributed in each station, and the communication protocol operating in the system. Individual workstation models may be built by selecting blocks from the class libraries. Each station model is defined as a composite block and represented as an icon in the LANSIM user interface. These icons are connected graphically according to the system topology. The simulation case is then executed. Parameters such as execution frequency of a function and propagation delays in the communication protocol may be easily entered or changed by using pop-up menus at runtime.

Alternative EMS configurations may be implemented by creating and connecting additional station models. For example, the communication protocol can be changed by replacing the original protocol block with an alternative block from the protocol library in each station model. The distribution of the EMS functions can also be varied: to study the effects of executing SE in a separate processor, an additional station is created, containing the the newly-defined SE traffic model. New EMS functions and communication technologies may be incorporated by developing models for them in the LANSIM class libraries. A distributed EMS based on both wide area and local area networks can be simulated by combining LANSIM with wide-area network simulators in NetPlan. The NetPlan environment with its block-oriented graphical construction provides extraordinary modularity, flexibility, and extendibility.

4. Distributed EMS Simulation

4.1. System Configuration

We study the LAN performance of two distributed EMS configurations. The objectives of this study are twofold. First, we compare Ethernet and FDDI LAN technologies in terms of their abilities to satisfy network performance requirements. Second, we examine the effects on network performance due to executing state estimation in a separate dedicated processor. Our base configuration, shown in Figure 3, is basically the same as that used in [2] except 5 consoles are assumed here. The second configuration, where the state estimator is executed in a separate machine, is shown in Figure 4.

The configurations studied are generic models for a distributed EMS, and are not intended to directly represent any specific supplier's configuration. Other configurations, such as those using additional LANs and alternate high-speed interconnects are also possible, but have not been included to keep the analysis generic.

An EMS has several stringent real-time performance requirements. For example, it is generally accepted that the delay of the data indicating a change in status from the front-end processor to the console screen should be less than one second. Since there are other processes such as engineering unit conversion, point processing and alarm detection that are included in this delay requirement, it is typical to budget a maximum of 10 ms delay for digital status data. These real-time requirements must be met even under the worst-case loading conditions, since the EMS still must be capable of performing its basic function, controlling the power system. The approach we take in this study is to examine network performance during peak loading conditions. Our simulations focus on the real-time behavior of the system, rather than the average performance[2]. While a network may have acceptable average performance, it may fail to meet the stringent real-time requirements.

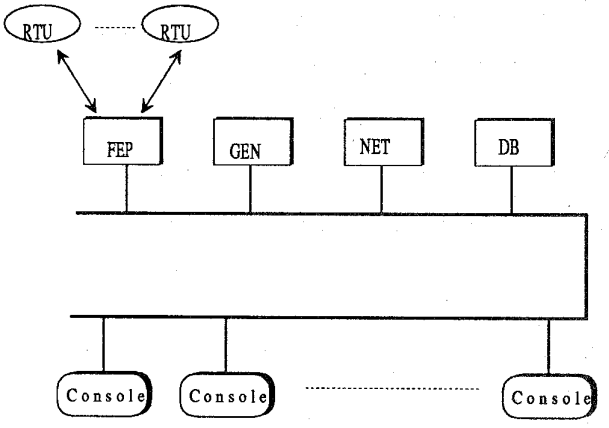

Figure 3 Base Configuration

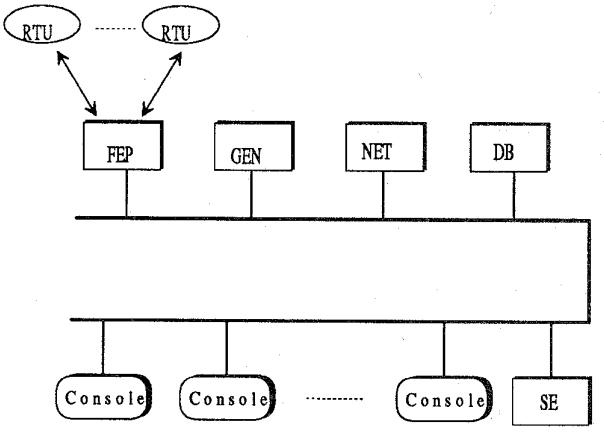

Figure 4 Configuration with a Dedicated SE

4.2. Traffic

Our study is based on EMS requirements typical of a medium-sized utility, with 50 dispatchable generator units, 500 buses in the internal network model, and 500 buses in the external network model.

We assume that during peak conditions, study functions are suspended. The front-end processor(FEP) polls the RTUs every 2 seconds and acquires analog and digital status data which is sent to the SCADA database processor(DB). The state estimation results from the network processor(NET) are sent to DB;this process is assumed to occur every minute in the base case. The output from AGC and ED in the generation processor(GEN) is also sent to DB. The alarm data is sent from DB to consoles; the occurrence is assumed random and with a uniform distribution. The consoles send alarm acknowledgments back to DB. There is also data from DB to consoles for updating the local database. We assume here that the picture data is stored in the local databases in the consoles, consistent with the client-server architecture of present-day full-graphics MMI. Supervisory control requires data to be sent from consoles to FEP; this traffic is assumed random with a Poisson distribution. Other background traffic includes DB to GEN and DB to NET. The packet rate and packet length of each type of traffic are given in Table 1. For traffic that is modeled as random, the packet rate is to be interpreted as its expected value.

We also model buffering and packetizing in our traffic model. Buffering is commonly used for analog data traffic from FEP to DB, for example. One buffering scheme allows data to accumulate in the buffer until either 1 kbyte of data is obtained or the time since the first packet was received exceeds 100 ms. The data is then sent out as a single packet. Packetizing is used to segment large messages into small packets; for example, the data from NET to DB is typically divided up into 1 kbyte packets apiece or smaller before being sent. For the configuration where a dedicated workstation is used for state estimation(SE), the volume of traffic is assumed to be 150 kbytes from SE to NET after each execution of SE.

	DB	Console	NET	GEN	FEP
FEP (digital)	4000/sec (8)				
FEP (analog)	500/sec (12)				
NET	1/min (150k)				
GEN(AGC)	1/2 sec (1k)				
GEN (ED)	1/5 sec (800)				
DB		10~15/sec (100)	1/8sec (50k)	5/sec (100)	
Console	5/sec (100)		1/50sec (200)		1/25sec (100)

Table 1 Traffic in Distributed EMS Study
{ Packet rate in packets per timer interval
(packets length in bytes) }

4.3. Results

Simulation results for the Ethernet base case are listed in Table 2. The actual network traffic is rather bursty, and unless the simulation period is long enough, the average delay depends on the duration of the simulation. A 30-second period is used in order to ensure only one peak traffic occurs when SE results are sent from NET to DB. The actual average value should be even smaller. Table 2 indicates that the average delays are well within real-time requirements (10 ms for digital data and 100 ms for analog data). However, unacceptable delays are observed during peak traffic due to the continuous transmission of long packets by one or more stations, resulting in increased channel latency for the other stations. The digital delay distribution over a 2-second period is shown in Figure 5.

	SCADA data (FEP to DB)		Alarm (DB to Console)
	Digital	Analog	
Average Delay	0.736 ms	1.246 ms	1.093ms
Maximum Delay	107.4 ms	80.366 ms	164.7ms

Table 2 Delay for the Base Case, Ethernet

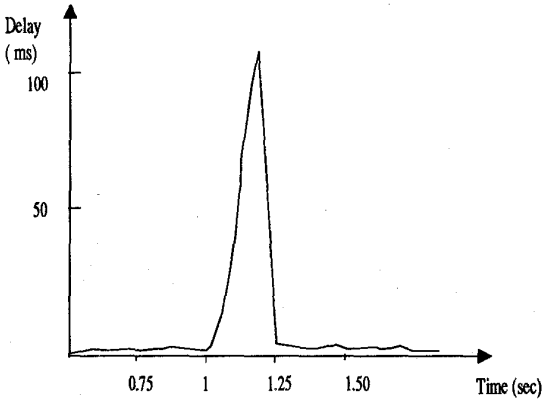

Figure 5 Packet Delay of Digital Data, Ethernet

A FDDI network is used to support the same traffic. In this case, the SCADA and alarm data is sent via synchronous transmission with TRT set at 8 ms. The results are shown in Table 3. A comparison of maximum delay values shows that FDDI improves the performance by more than an order of magnitude. While FDDI is 10 times faster than Ethernet, its ability to meet timing requirements is attributed to its bandwidth allocation scheme and guaranteed response time.

	SCADA data (FEP to DB)		Alarm (DB to Console)
	Digital	Analog	
Average Delay	0.017ms	0.065ms	0.024 ms
Maximum Delay	0.213ms	0.210ms	0.088 ms

Table 3 Delay of FDDI

The packet lengths resulting from the fragmentation of large-size data are significant factors affecting Ethernet performance. Larger fragmentation lengths will generally increase the delay of other traffic for a short period of time. The effect of packet size is shown in Figure 6.

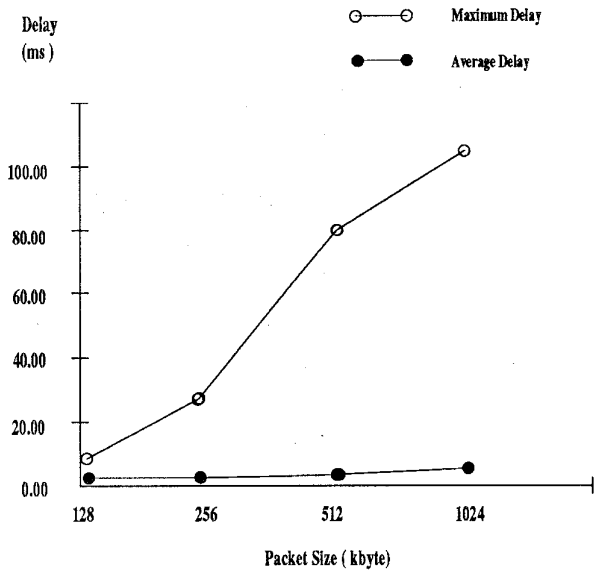

Figure 6 Delay vs. Packet Size, Ethernet

It should be noted that using smaller packetizing lengths will result in an increase in the number of packets, and consequently, a substantial overhead in packet processing time. Currently, the operating system is not modeled in our simulation; thus, the effects of this overhead do not appear in the results.

The effect of buffering is shown in Table 4. Buffering reduces the maximum delay somewhat (around 10% in this case) as well as the packet processing time which is again not simulated here.

	SCADA data (FEP to DB)	
	Digital	Analog
With Buffer	108.66 ms	90.526 ms
Without Buffer	109.4 ms	104.9 ms

Table 4 Maximum Delay with Buffering, Ethernet

Pulling out the State Estimator causes an increase in overall delay on Ethernet as shown in Table 5. This condition may worsen when additional applications such as OPF are also run distributedly. The performance degradation is again due to collisions and retransmission attempts in Ethernet; FDDI, on the other hand, does not have such problems.

	SCADA data (FEP to DB)		Alarm (DB to Console)
	Digital	Analog	
Average Delay	1.652 ms	1.987 ms	1.59 ms
Maximum Delay	108.50 ms	99.745 ms	181.43 ms

Table 5 Delay with State Estimation, Ethernet

5. Conclusions

LANSIM, a LAN simulator based on an object-oriented design environment, is used to evaluate the network performance of a LAN-based distributed EMS. The real-time behavior of two LAN technologies, Ethernet and FDDI, is examined. Real-time response requirements dictate that real-time, rather than average, behavior should be examined. To study the performance of various configurations of a distributed EMS, a flexible and extendible tool such as LANSIM is required. The state estimator is pulled out to study the impact of running network applications distributedly. A conclusion of this paper is that the move towards a more distributed EMS imposes significantly greater throughput requirements on LANs and a more thorough analysis of system performance under more realistic conditions should be conducted.

Acknowledgment

We would like to thank Bob Burn of ABB Systems Control for extremely helpful discussions.

References

[1] J. Britton, 'Utility Control Centers Open to Change', IEEE Computer Applications in Power, Oct. 1991, pp. 35-39.

[2] K. Kato and H. R. Fudeh, 'Performance Simulation of Distributed Energy Management Systems', IEEE Trans. on Power Systems, Vol. 7, No. 2, pp 820-827, May 1992.

[3] M. Kunugi et al, 'Performance Validation of a

Functionally Distributed Energy Management Architecture', IEEE Trans. on Power Systems, Vol. 7, No. 2, pp 828-834, May 1992.

[4] G. Ockwell and R. Kreger, 'The Impact of Hardware on Open Architecture Design', paper 92WM159-4, presented at the IEEE/PES Winter Meeting, New York, Jan. 26-30, 1992.

[5] R. Podmore, 'Criteria for Evaluating Open Energy Management Systems', paper 92WM157-8, presented at IEEE/PES Winter Meeting, New York, Jan.26-30, 1992.

[6] L. Murphy and F.F. Wu, 'An Open Design Approach for Distributed Energy Management Systems', paper 92SM447-3 presented at IEEE/PES Summer Meeting, Seattle, WA, July 12-16, 1992.

[7] P. Coad and E. Yourdon, *Object-Oriented Analysis*, 2nd. Ed., Yourdon Press, 1991.

[8] S. M. Lun et al, 'NetPlan : An Integrated Network Planning Environment', Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecomm. Systems, Jan. 1993.

[9] R. E. Shotwell, ed., *The Ethernet Sourcebook*, 3rd. Ed., North-Holland, 1985.

[10] F. E. Ross, 'An Overview of FDDI: The Fiber Distributed Data Interface', IEEE Journ. on Selected Areas in Comm., Vol. 7, No. 7, pp. 1043-1051, Sept. 1989.

[11] J. Buck , S. Ha, E. A. Lee, and D.G. Messerschmitt, 'Ptolemy: A Platform for Heterogeneous Simulation and Prototyping,' Proc. European Simulation Conf., Copenhagen, June 1991.

[12] American National Standard Institute, *FDDI Media Access Control (MAC-M)*, June 1989. Working Draft Proposed American National Standard.

Shau-Ming Lun received the Ph.D. from UC Berkeley and is currently a postdoctoral researcher in Electrical Engineering & Computer Sciences at the same institution.

Tetiana Lo received the B.S. and M.S. degrees from UC Berkeley and is currently a Ph.D. candidate in EECS at the same institution.

Felix Wu received his Ph.D. from UC Berkeley, where he is currently a professor of Electrical Engineering and Computer Sciences. Dr. Wu is a Fellow of IEEE.

Liam Murphy received the Ph.D. from UC Berkeley and is currently a Lecturer in EECS at the same institution. Dr. Murphy is a Member of the IEEE.

Amitava Sen received his Ph.D. from the Carnegie Mellon University and is currently a Product Manager with ABB Systems Control Inc., Santa Clara, CA.

Discussion

A. Mayer Sasson (Consolidated Edison Co., 128 West End Avenue, New York, NY 10023):

It is well known that the configuration of EMS systems in the nineties has changed radically from the previous generation of isolated primary backup platforms. Indeed EMS systems consist of many computer components all networked together. While previous EMS systems were relatively static, today's workstation based systems have components added to them quite frequently. The concept of an universal graphical user interface has replaced that of slave terminals for EMS operators. This implies that access must be provided from any operator workstation to all systems on the network. Recent times have also witnessed the advent of wide area networks connected to the EMS for access to other utility systems such as substations, generating plant, AM/FM facilities, distribution control centers, corporate systems, etc.

The above implies that the network imbedded in EMS systems is a dynamic one. There is a need to have tools available on-line to perform studies to determine the effect of these changes on the performance of the system. Studying the impact on the traffic in certain portions of the network by a proposed new addition is of great value. There are always alternative designs possible and these need to be evaluated ahead of time. It would appear that simulation systems such as the one reported in this paper would be a valuable tool in the hands of EMS designers.

Closure was not provided by the Author.