# Distributing Time-synchronous Phasor Measurement Data Using the GridStat Communication Infrastructure

Ryan A. Johnston, Carl H. Hauser, K. Harald Gjermundrød, and David E. Bakken
*Washington State University, Pullman, Washington*
*ryan.a.johnston@gmail.com, {hauser,kgjermun,bakken}@eecs.wsu.edu*

## Abstract

*The emergence of phasor measurement units (PMUs) coupled with GPS time devices makes it feasible to directly compare timestamped measurements collected at different locations without needing to account for clock drift. GridStat is a flexible publish-subscribe status dissemination middleware framework with capabilities that include rate-filtered multicast for each subscription. In this paper we explore GridStat's ability to hide the complexities of distributed systems from application developers while efficiently providing time-synchronous groups of PMU data from physically disparate locations.*

## 1. Introduction

The growing availability of synchronous phasor measurement units (PMUs) and their deployment on power grids is making available a wealth of new real-time data about the state of the grid.

PMU data contains much more information than is typically measured in existing SCADA systems. PMU measurements are synchronized and timestamped using a global clock (GPS). A PMU's measurements, made 20 to 60 times per second, allow derivation of the bus current, voltage, phase and frequency where it is attached.

Existing mechanisms for using these data are mainly targeted at collecting them in archival central repositories for use in after-the-fact analysis of system response to normal and abnormal events. These collection mechanisms have limited ability to support use of the data for real-time applications.

Nevertheless, the ability to receive status data from a variety of remote locations opens new, largely unexplored opportunities for improving the power grid's control systems to provide better reliability and efficiency. Ongoing research, [18, 19], suggests that PMU data will be valuable in fast control applications, but widespread use of such mechanisms will require a more flexible communication infrastructure than exists today. Current fast controls are either local or require dedicated point-to-point communication links. Such a communication infrastructure makes it expensive to accommodate changing needs.

GridStat is a middleware framework for dissemination of operational power grid status data (both digital and analog). Middleware is a layer of software above the operating system which provides a common programming abstraction across a distributed computing system. Its high-level building blocks shield the programmer from many different kinds of heterogeneity that are inherent in a distributed system, including that of CPU, operating system, and programming language.

A fundamental design motivation of GridStat was to provide flexible, managed, low-latency sharing of status data among entities that participate in operation of an interconnected power grid, [7]. GridStat's management capabilities are intended to ease the deployment of applications by encapsulating solutions to problems such as quality of service (QoS), reliable delivery, and security in the communication infrastructure rather than leaving their solution to applications. GridStat was designed to provide high-bandwidth communication for power grid data, delivering the right data to the right computer at the right frequency with the ability to easily adapt to changing needs.

GridStat and PMUs thus appear to be highly complementary technologies. In this paper we explore the fit. After describing the PMU data stream (section 2) we show how GridStat's mechanisms (section 3) might be exploited in applications using PMU data (section 4). Of course the performance of the communication infrastructure is important if PMU data is to be used in fast controls. Section 5 presents performance results using the current GridStat prototype on emulated PMU data streams. We wrap up with a short discussion of related work (section 6).

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

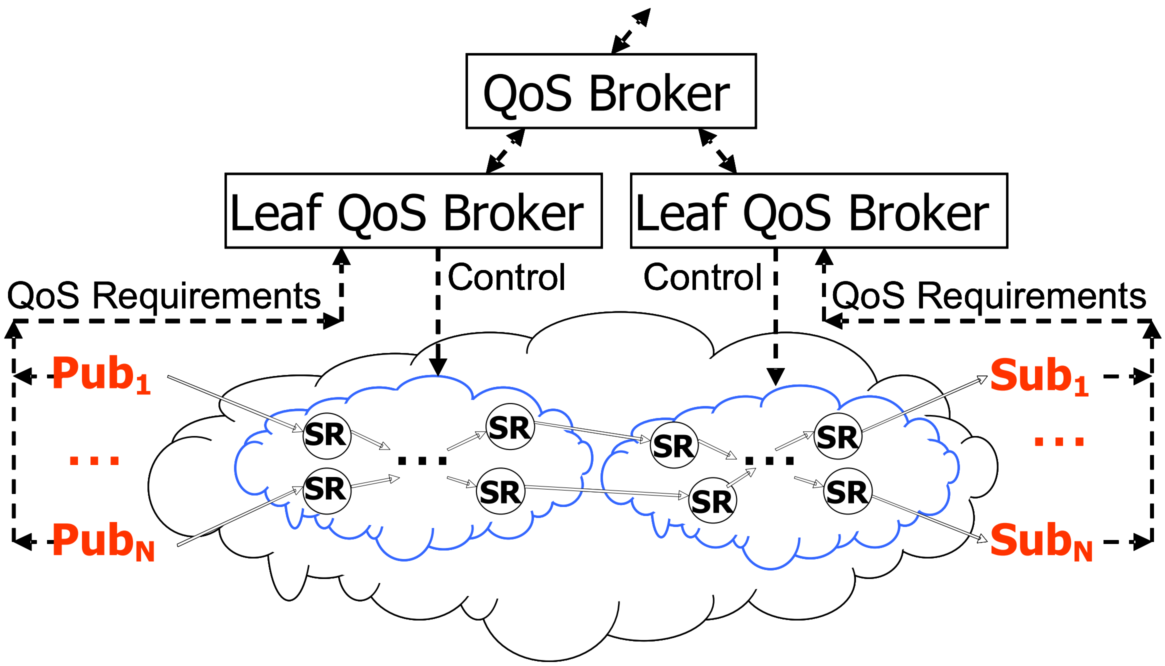


**Figure 1. GridStat architecture**

## 2. PMU data

PMUs are typically placed in a substation to monitor that substation's three phase voltages, as well as the current in lines, transformers, and loads that terminate there, [5]. These quantities are collected along with frequency, and each sample is timestamped with a time value that is accurate to within one microsecond of UTC. The transmission format of PMU data packets is defined in IEEE standard 1344, [8].

The length of an IEEE 1344 data packet depends on the number of phasor and digital quantities reported by a particular PMU. A basic PMU packet is 14 bytes long consisting of

- 4-byte second-of-century (SOC)
- 2-byte sample count (SMPCNT)
- 2-byte status (STAT)
- 2-byte frequency (FREQ)
- 2-byte rate-of-change-of-frequency (DFREQ)
- 2-byte CRC16

A separate configuration file specifies how many 4-byte phasors are inserted in each data frame between the STAT and FREQ words and how many 2-byte digital status words are inserted between the DFREQ and CRC16 words. For example, a PMU measuring 3-phase voltage phasors and no digital status would send 26-byte frames; one measuring 3-phase voltage, and current, and 32 bits of digital status would send 42-byte frames.

The configuration file also controls the frequency at which frames are sent. Twenty or thirty frames per second is common today. Researchers looking into use of PMU data for wide-area control suggest that moving to 60 frames per second would be useful in their application, [20].

Even at only 20 frames per second a PMU device produces far more data than does older-technology power system monitoring equipment designed for the 2- or 4-second SCADA polling cycle. At 60 frames per second a PMU measuring 3-phase voltage and current and 32 digital samples (42-byte frames) produces just over 20Kbits/second: huge by the standards of legacy SCADA but miniscule by the standards of even consumer-grade broadband networking service.

## 3. GridStat architecture and mechanisms

GridStat is a flexible communication framework for delivering periodic status updates between equipment operated by entities such as grid operators, energy suppliers, and marketers. Its architecture defines the relationships between GridStat entities. The architecture is designed to provide the flexible technical underpinnings needed to meet diverse status communication needs of the different kinds of business entities. At a more detailed level, GridStat's mechanisms encapsulate reusable communication paradigms such as multicast, filtering, QoS management, and redundant path routing. These are quite naturally part of the communication infrastructure and would be very difficult or impossible to provide without such built-in support.

## 3.1. Architecture

The GridStat architecture encompasses four primary types of active entities as shown in Figure 1. *Publishers* produce periodic streams of status values. *Status routers* accept status values from publishers and other status routers and forward them to other status routers and *subscribers*. Subscribers run applications that use (or store) the data. Publishers, status routers, and subscribers make up the GridStat *data plane*. The data plane's job is to move data packets quickly from publishers to subscribers. The GridStat architecture supports independent ownership and control of data plane components and anticipates that groups of components under common control will work together more closely than will components that are controlled by different business entities.

The fourth type of active GridStat entity is the *QoS broker*. QoS brokers are arranged in hierarchies that reflect geographic and business hierarchies. The QoS broker hierarchies constitute the management plane of GridStat. The lowest-level QoS brokers (leaf QoS brokers) each directly manage resources within a set of commonly-controlled data plane components. QoS brokers are responsible for establishing the route(s) through the status routers used by each subscription so as to ensure that each subscription's QoS requirements are met.

GridStat simultaneously provides a hierarchical management model and a flat delivery model. QoS brokers are not involved in packet-by-packet delivery: they merely direct the set-up of the status routers and monitor their performance, providing adaptation ability if status routers or links fail.

## 3.2. Status router mechanisms

Status routers include several mechanisms of use for PMU-based applications of which efficient multicast, synchronous filtering, packet combining and preconfigured modes are of the most interest for PMU applications. These mechanisms and their implementations are fully described in [4]. Here we provide only a short synopsis.

**3.2.1. Efficient multicast.** A major goal of GridStat is to allow sharing of status data. Multicast allows sharing without requiring multiple copies of each data item to traverse communication links as would be required if only unicast were provided. This makes better use of communication resources and reduces the latency seen by most subscribers.

**3.2.2. Rate filtering.** It is clear that not every subscriber needs every update produced by the publishers to which it subscribes. Rate filtering allows each subscriber to indicate, as part of its subscription, the rate at which it needs updates. The status routers implement rate filtering in conjunction with multicast and only forward packets that are desired by downstream subscriptions. Rate filtering is performed on the basis of timestamps contained in the packets—not the local clock of the status routers.

One of the biggest advantages PMU data has over conventional SCADA data is that it is synchronously timestamped with its acquisition time. By collecting measurements from various points taken *at the same time* an application can derive properties of the grid's state that are not available from data taken at different times. On the other hand, it is clear that some applications for PMU data will not require all of the samples taken during each second. Synchronous filtering allows an application to subscribe to every *n*th element of several PMU data streams and get samples that have "close" timestamps. (The question of whether or not the timestamps are the same depends on the PMU implementation.)

**3.2.3. Packet combining.** Status routers combine status packets into network layer 3 packets when it is possible to do so without incurring additional latency. This reduces layer 3 bandwidth overhead. More importantly it reduces the execution time overhead of both the sending and receiving status routers by reducing context switches.

**3.2.4. Operational modes.** Predefined modes allow the communication infrastructure to quickly adapt to changes in the operational state of the power grid. Each subscription operates in a fixed set of modes, specified by the subscriber. The QoS brokers can preload the status routers with multiple routing configurations for the different modes. This permits rapid switching between different bundles of subscriptions. These mode transitions are accomplished without disrupting subscriptions that exist in both the before-transition and after-transition modes. Doing resource allocation for preconfigured modes at the time subscriptions are created ensures that the new mode will become operational quickly and without risk that resource allocation will fail at perhaps a critical moment for the power grid.

## 3.2. Status packetization

The GridStat prototype uses a simple packetization scheme with each packet consisting of a basic 24-byte header (which includes 8 bytes of user data) and a variable number of 24-byte data extensions. A GridStat packet contains:

- 8-byte timestamp (milliseconds since Jan. 1, 1970)
- 4-byte publication ID
- 1-byte number of optional fields
- 3-bytes padding
- 8-bytes user data
- 0 to 4 24-byte optional user data fields

GridStat packets are encapsulated in Internet Protocol User Datagram Protocol (IP/UDP) packets in the prototype which uses Internet technology as its link layer.

# 4. Use of GridStat mechanisms for PMU data dissemination

While PMU data is valuable in many applications, the number of actual packets required per second varies depending on the application. GridStat provides the framework to allow a PMU to publish its data as fast as it can provide it, but only those packets that are required by subscribing applications are actually delivered. For instance, if a PMU is publishing at a rate of 60 times per second, and a subscribing application requests 1 packet every 2 seconds, only a single PMU data frame will traverse the GridStat network every other second. This middleware level packet filtering assures application developers that network traffic is only that which is needed to fulfill their particular application's requirements.

Packet filtering is useful for minimizing the use of network resources, but many applications require data from multiple PMUs to be effective. Such applications require that the packets delivered share a common timestamp, regardless of what path they take through the network. As mentioned in section 3.2.2, GridStat's synchronous filtering fulfills this requirement while preserving the network traffic reduction of packet filtering.

Two issues arise in encapsulating PMU frames in GridStat packets: timestamp mapping and size mapping. For timestamp-based filtering to be useful in PMU applications each packet timestamp must match the globally synchronized PMU timestamp of the encapsulated PMU frame. Packet timestamps refer directly to a global epoch. Timestamps in PMU frames (SOC || sample number within second), are referenced to a globally synchronized clock. However, because the number of samples per second is part of the PMU configuration, mapping a PMU timestamp to a GridStat timestamp requires knowledge of the PMU configuration—something that is *not* part of the PMU data packet. It is therefore necessary for the PMU publisher to interpret the PMU timestamp and construct the GridStat timestamp.

Because the GridStat packets expand by 24 bytes at a time the question of overhead due to internal fragmentation must be considered. For our running example of a PMU reporting 3-phase voltage and current along with 32 digital statuses (42-byte PMU frame) the corresponding GridStat packet is 72 bytes of which 14 are padding and 16 are packet header.

To support the collection of PMU data from multiple locations, GridStat provides an interface for specifying a set of publications for subscription. In this way, the entire set may be accessed from a subscribing application as a single unit rather than as a series of individual subscriptions.

As a middleware system, GridStat provides an abstraction layer that removes the complexities of distributed systems from the application developer. In addition, GridStat provides an application development framework to help streamline the process of creating applications using GridStat. This allows applications to be developed quickly and easily, even by individuals who may not have a background in distributed systems.

# 5. GridStat performance on PMU data

To demonstrate the suitability of GridStat to meet the data transmission requirements of PMUs, we developed two sample applications. A virtual PMU data publisher emulates a publisher attached to a real PMU producing data of the same size and at the same rate. A virtual phasor data concentrator (PDC) subscribes to all the emulated PMU data.

## 5.1. PDC requirements

We set out to show empirically that GridStat can fulfill the data communication needs of a PDC. We used the recommendations of the EIPP real time task team, [14], as a basis for our evaluation of GridStat. Those requirements specify that a PDC should be able to receive each PMU's data 30 times a second. Further, to be useful for real-time monitoring, data must be collected from at least twelve different sources. For real time adaptive control applications the communications infrastructure is expected to add approximately 10ms to the overall time delay of PMU data delivery. To meet short-term needs, therefore, GridStat should be able to provide a PDC data from at least 12 PMU publishers at a rate of 30 per second while contributing less than 10ms of delay. In the longer term there will be many more PMUs on the grid so tests were done with 100 PMU data streams.

## 5.2. Experimental setup

To illustrate that GridStat can meet the stated requirements we conducted four experiments. All of the experiments ran 100 separate publishers, publishing simulated PMU data frames either at 30 or 60 times per second. The PMU data frames were composed of six phasor fields and 2 digital channel fields which made the PMU frame size 42 bytes. With the additional GridStat header information and padding, the packet size was 72 bytes. In all four experiments, a single virtual PDC subscribed to the entire set of 100 published data items at a rate either of 30 or 60 times per second. All 100 publishers as well as the PDC were run on a single computer. This allowed use of the local clock to calculate end-to-end delay without concern for clock synchronization. All of the computers used in these experiments were current home-PC-class machines running Linux, and connected by a switched 100Mbps Ethernet.

**5.2.1. Single path delay jitter.** This pair of experiments used three status routers: two edge status routers connected by a single internal status router as shown in Figure 2. (The difference between edge and internal status routers is that edge status routers provide interfaces for publishers and subscribers.) Each of the routers ran on a dedicated computer. All 100 publishers were connected to one edge status router and the PDC was connected to the other.

**5.2.2. Redundant two-path delay jitter.** These two experiments were identical to the previous except for the addition of a second internal status router that independently linked the two edge status routers as shown in Figure 5.2.

## 5.3. Experimental results

Figure 4 and Table 1 present the results of the four experiments. In each experiment 500,000 PMU data frame transmissions were evaluated for end-to-end delay. In all of the experiments all of the frames that were sent were delivered.

In all four experiments the vast majority of PMU data frames were delivered in less than 2ms (bars 0 and 1 on the graphs), and over 99% were delivered in less than 10ms. End-to-end delays above 10ms, summarized in Table 1, indicate delay jitter attributable to use of a general purpose operating system, and Java with garbage collection. Increased delay and jitter when using redundant paths is attributable to the additional work performed at the edge routers.

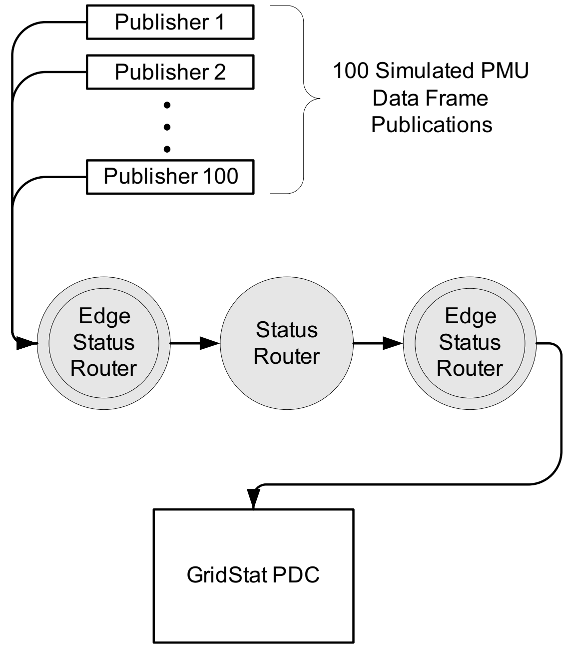


**Figure 2. Single path configuration**

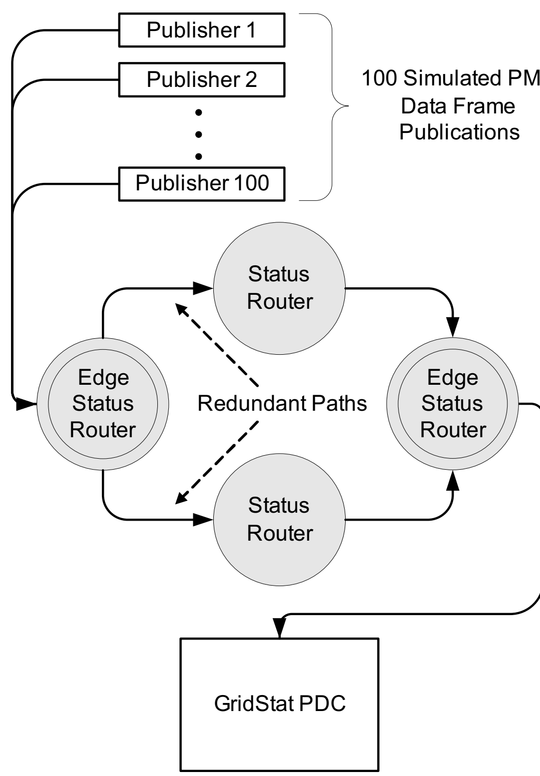


**Figure 3. Redundant path configuration**

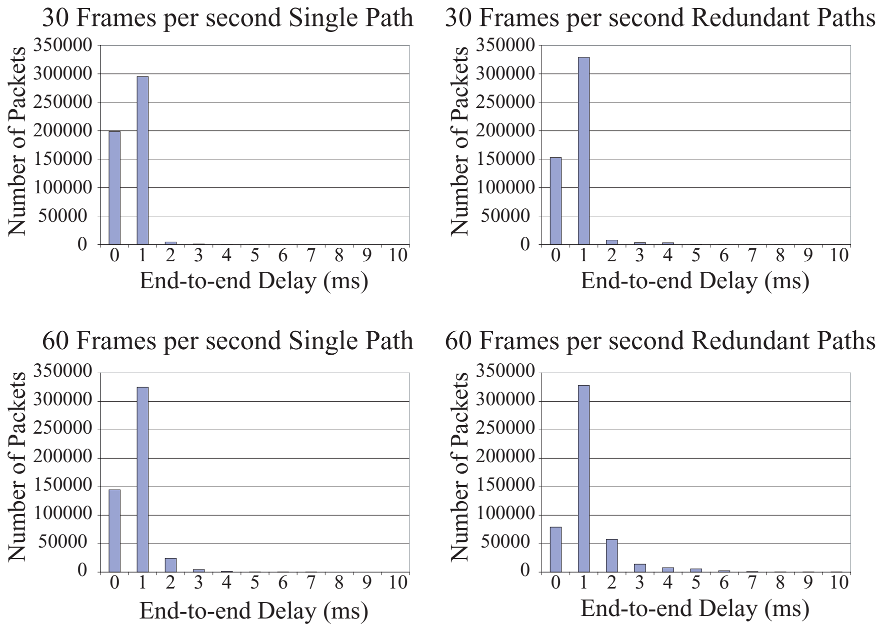


**Figure 4. End-to-end delay**

These experiments generally support the feasibility of using a multi-casting based infrastructure for PMU status dissemination. The phasor data requirements document from EIPP, [14], does not specifically address acceptable amounts and frequencies of jitter. We believe that both the jitter and latency observed in these experiments could be significantly reduced in an implementation that did not use Java. Of course this requires confirmation.

Further experiments are also needed to establish the number of routers that can be in a path while still meeting the delay and jitter requirements. Other GridStat experiments have shown a delay penalty of less than 1ms per additional router, [4]. Propagation time between widely-separated power grid entities must also be added.

**Table 1. Delays over 10ms**

| Experiment | End-to-end over 10ms | Percentage |
|---|---|---|
| 30/s Single Path | 173 | 0.034% |
| 30/s Redundant Paths | 2527 | 0.505% |
| 60/s Single Path | 108 | 0.022% |
| 60/s Redundant Paths | 3808 | 0.762% |

## 6. Related work

The Wide Area Measurement System (WAMS) on the U.S. Western Interconnection uses PMUs to gather data from several points in the interconnection to allow off-line analysis of the grid's performance, [15]. The data gathered has been instrumental in understanding outages on the system such as those occurring in July and August, 1996, [6]. WAMS has inspired a number of commercial offerings such as those of, for example, [11] and [12].

Currently, the Eastern Interconnection Phasor Project is deploying PMUs on the Eastern U.S. grid. EIPP is developing a communication infrastructure to collect and archive the measurements at the TVA, [1]. In addition to archiving, the TVA site makes a feed of all the collected data available to participating utilities within a few seconds after it is collected. GridStat, of course, would allow delivery with a few milliseconds of latency.

The FNET project is exploring uses for a large number of frequency monitors throughout the North American grid, [13]. Frequency monitors are similar to, but simpler than PMUs. FNET uses the Internet, without management, to collect data for analysis.

The Wide-Area stability and voltage Control System (WACS) project of the Bonneville Power Administration, Ciber, Inc. and Washington State University uses PMUs as sensors in a real-time system for stability and voltage control, [19]. Calculations from that project provide a time budget for creation, delivery and processing of PMU measurements in the Western Interconnection. The budget is based on the need to respond to a disturbance within 750-1500ms. Of this time budget approximately 67 ms is attributed to communication latency and jitter using the project's current point-to-point communication infrastructure. WACS uses a dedicated communication network to collect input from its sensors. Other projects using networked PMUs for control include [18] and [10].

To date none of the power systems projects using PMUs has adopted a middleware framework for its communication infrastructure. For example, the Data Distribution Service (DDS) for Real-time Systems Specification by the Object Management Group addresses communication of signals, streams and states, [16]. Existing implementations of the specification are targeted to the LAN environment and do not address QoS management issues associated with the wide area, [17, 21]. The PASS system was designed for monitoring the status of a communication network, [22]. It does address QoS in the wide area but does not provide independent QoS specification for each subscription.

## 7. Conclusions and future work

The prototype status router is implemented in Java. Care has been taken to avoid memory allocation (and hence garbage collection) when moving packets through the router. Beyond this, no unusual optimizations have been used. For production use a language closer to the hardware, such as C, or even implementation in specialized *network processors,* [2], should be considered. Nevertheless, the current Java implementation on current home-PC-class hardware is able to service large numbers of PMU data streams.

In the experiments reported here the packet combining feature described in section 3.2.3 played little role. A small amount of packet combining was observed during the experiments. It was not enough to be a major factor in the success of the experiments.

Although not discussed in detail here the GridStat PDC shows how support for periodic updates and redundant paths in the infrastructure leads to simple application code, [9]. There is no code in the publishers related to redundant path routing. The subscriber has code only to request redundant paths. Otherwise the application is unchanged from the single-path case. Exploration of the utility of operational modes for PMU applications also remains for future work.

The complexity of the communication relationships in the power grid's control system is an ongoing challenge for the industry. GridStat's goals are complementary to those of efforts such as IEC standards 61850 and 61970 that provide frameworks for defining object and service models for field equipment and information models for control centers. There is a great deal of research to be done to see how publish-subscribe communications meshes with these models.

Another challenging aspect of the modern power grid is the complexity of the relationships between the businesses that operate its components. Economic competitors must nevertheless cooperate to assure reliable grid operation. There is legitimate fear that information exchanged for the latter purpose could be used or manipulated to provide competitive advantage. PMU data, which reveals the power transfer, is therefore potentially sensitive. The infrastructure must deliver data *on time,* but also *with integrity* and *only to legitimate subscribers.* Research in trust management will inform the incorporation of these guarantees into the QoS brokers and status routers, [3].

## 8. Acknowledgements

This research was partially supported by NSF grant ITR CCR-0326006. We are grateful to Schweitzer Engineering Laboratories and to Digilent for their donations of hardware for this project. Discussions with Matt Donnelly at PNNL were instrumental in understanding the role that GridStat could play in disseminating PMU measurements.

## 9. References

[1] Carroll, J. "Eastern Interconnect Phasor Project: TVA's Super Phasor Data Concentrator Architecture" presentation at the Eastern Interconnect Phasor Project Meeting, April, 2005, Chattanooga, TN. Retrieved September 2005: http://phasors.pnl.gov/Meetings/2005%20April/presentations/TVAs%20Super%20Phasor%20Data_Carroll_EIPP.ppt

[2] Comer, D. *Network Systems Design Using Network Processors,* Prentice Hall, 2003.

[3] Drugan, O., Dionysiou, I., Bakken, D., Plagemann, T., Hauser, C., and Frincke, D. "On the Importance of Composability of Ad Hoc Mobile Middleware and Trust Management", *Proceedings of the 2nd International Service Availability Symposium,* Berlin, Germany, 2005. *Lecture Notes in Computer Science,* Springer, 2005, to appear.

[4] Gjermundrød, K. H., Bakken, D., Hauser, C. "GridStat Architecture and Mechanisms", in preparation.

[5] Hart, D., Uy, D., Phadke, A., Forsman, S., and Kunsman, S. "PMUs: Overview and Application to Electric Power Networks", *Proceedings of the 4th Multiconference on Systemics, Cybernetics, and Informatics (SCI 2000/ISAS 2000), volume IX*, Orlando, USA.

[6] Hauer, J. and Taylor, C. "Information, Reliability and Control in the New Power System", *American Control Conference,* Philadelphia, PA, June, 1998.

[7] Hauser, C., Bakken, D. and Bose, A. "A Failure to Communicate: Next-Generation Communication Requirements, Technologies, and Architecture for the Electric Power Grid", *IEEE Power and Energy*, 3(2), March/April, 2005, 47-55. Retrieved Sept. 2005: http://gridstat.net/GridStat-Power-Energy-March05.pdf.

[8] IEEE. "IEEE Standard for Synchrophasors for Power Systems", IEEE Std 1344-1995(2001), New York, 2001.

[9] Johnston, R. *Obtaining High Performance Phasor Measurements in a Geographically Distributed Status Dissemination Network.* MS Thesis, Washington State University, August, 2005. Retrieved Sept. 2005: http://www.gridstat.net/publications/GridStat-TR-EECS-GS-007.pdf

[10] Kamwa, I., Grondin, R., and Hebért, Y. "Wide-Area Measurement Based Stabilizing Control of Large Power Systems—A Decentralized/Hierarchical Approach", *IEEE Transactions on Power Systems,* 16(1), February, 2001.

[11] Larsson, M., Gardner, R., and Rehtanz, C. "Interactive Simulation and Visualization of Wide-area Monitoring and Control Applications", submitted to *Power Systems Computation Conference,* Liège, Belgium, 2005.

[12] Larsson, M., Rehtanz, C., and Bertsch, J. "Real-time Voltage Stability Assessment for Transmission Corridors", *Proceedings of IFAC Power Plants and Power Systems Control Conference*, Seoul, Korea, 2003.

[13] Liu, Y. "GPS/Internet Based Frequency Monitoring Network" presentation at the Eastern Interconnect Phasor Project Meeting, April, 2005, Chattanooga, TN. Retrieved Sept. 2005: http://phasors.pnl.gov/Meetings/2005 April/presentations/Post this oneEIPP_TVA_Liu_05.pdf

[14] Martinez, C., Parashar, M., Dyer, J. and Coroas, J. (eds.), *Phasor Data Requirements for Real Time Wide-Area Monitoring, Control and Protection Applications (Final Draft),* Consortium for Electric Reliability Technology Solutions (CERTS), January, 2005. Retrieved Sept. 2005: http://phasors.pnl.gov/resources_realtime/EIPP_RealTimeGroup _DataRequirements_Draft5 - Jan 26 05.pdf

[15] Mittelstadt, W., Krause, P., Overholt, P., Sobajic, D., Hauer, J., Wilson, R., and Rizy, D. "The DOE Wide Area Measurement System (WAMS) Project—Demonstration of Dynamic Information Technology for the Future Power System", *EPRI Conference on the Future of Power Delivery,* Washington, D.C., April 1996. Retrieved June 2005: http://www.osti.gov/bridge/purl.cover.jsp?purl=/254951-e5g22f/webviewable/

[16] Object Management Group. Data distribution service for real-time systems specification. Technical Report Version 1.0, Object Management Group, December 2004.

[17] Pardo-Castellote, G. "OMG Data-distribution Service: Architectural Overview". *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW03),* Providence, RI, May 2003.

[18] Quintero, J. and Venkatasubramanian, V. "A Real-Time Wide-Area Control Framework for Mitigating Small-Signal Instability in Large Electric Power Systems", *Proceedings of the 38th Annual Hawaii International Conference Software and Systems*, Hawaii, 2005.

[19] Taylor, C., Erickson, D., Martin, K., Wilson, R. and Venkatasubramanian, V. "WACS—Wide-Area Stability and Voltage Control System: R&D and Online Demonstration", *Proceedings of the IEEE* (Special Issue on Energy Infrastructure Systems), 93(5), May, 2005.

[20] Tomsovic, K., Bakken, D., Venkatasubramanian, V. and Bose, A. "Designing the Next Generation of Real-Time Control, Communication and Computations for Large Power Systems", *Proceedings of the IEEE* (Special Issue on Energy Infrastructure Systems), 93(5), May, 2005. Retrieved Sept. 2005: http://gridstat.net/Power-GridStat-ProceedingsIEEE.pdf

[21] van't Hag, J. "Data-Centric to the Max — The SPLICE Architecture Experience", *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW03)*, Providence, RI, May 2003.

[22] Zinky, J., O'Brien, L., Bakken, D., Krishnaswamy, V., and Ahamad, M.. "PASS: A service for efficient large scale dissemination of time varying data using CORBA", *International Conference on Distributed Computing Systems*, Austin, TX, June 1999.