Smart Generation and Transmission with Coherent, Real-Time Data[1]

David E. Bakken[2], Anjan Bose, Carl H. Hauser, Edmond O. Schweitzer III[3], David E. Whitehead[3], and Gregary C. Zweigle[3].

Technical Report TR-GS-015
School of Electrical Engineering and Computer Science
Washington State University; Pullman, Washington, USA
August, 2010

Abstract

In recent years much of the discussion involving “smart grids” has implicitly involved only the distribution side, notably advanced metering. However, today's electric grids have many challenges involving the rest of the grid—the bulk power system—that can be mitigated by making it more intelligent. An enabling technology for helping the bulk power system that has emerged in recent years is coherent, real-time data such as synchrophasors. In this paper we describe major challenges facing electrical generation and transmission today, including distributed generation (both micrograds and renewables), that availability of these measurements can help address. We overview applications utilizing coherent, real-time measurements that are in use today, or proposed by researchers. We then describe, normalize, and then quantitatively compare key factors for these power applications that influence how the delivery system should be planned, implemented, and managed. These include whether a person or computer is in the loop; and for both inputs and outputs: low latency, rate, criticality, quantity, and geographic scope. This represents a significant expansion of the NASPInet Service Class concept. From this, we derive the baseline communications requirements of a data delivery system supporting these applications and suggest implementation guidelines to achieve them. Next, we overview the state of the art in the supporting computer science areas of networking and distributed computing (including middleware), and analyze gaps in available network protocols, commercial middleware products, and utility standards in this area. We finally overview the emerging NASPInet as well as WSU’s GridStat project, which since 1999 has been defining, designing, developing, and deploying middleware to meet these emerging needs of the bulk power system.

[1] This paper is a significantly expanded version of an invited paper for *Proceedings of the IEEE*. It will likely appear there in 2011. This footnote in this TR will be updated to reflect this; please cite the ProcIEEE paper if the results you wish to cite from this Technical Report are also in that version (when it comes out). URL: http://gridstat.net/publications/TR-GS-015.pdf
[2] Contact author; bakken@wsu.edu, +1-509-335-2399
[3] Affiliated with Schweitzer Engineering Labs, Inc. These authors have provided much material for Section 3, but WSU solely is responsible for the content of this Technical Report.

Contents

List of Figures

List of Tables

1. Introduction and Background

Large power grids around the world were built mostly or completely from the ground up. During the middle of the twentieth century utilities integrated into larger power grids in order to improve reliability. In such a structure, an entire grid such as the USA Eastern Grid or Western Europe's ENTSO-E (formerly UTCE) *ipso facto* operates at the same frequency, so supply and demand must be balanced in real time across each grid.

Unfortunately, limitations of combining utilities became apparent in part due to a large blackout in the northeastern USA and southeastern Canada in 1965. As a result of this, it was realized that utilities need to have better visibility into their operations beyond what can be sensed in a control center. From this, SCADA was born. Regrettably, communications technologies deployed in the grid are largely unchanged from the 1960s, although they have been augmented in a piecemeal basis by newer networking technologies. Four decades later electric grids still have inadequate communications; we are literally "flying blind" [Eco04].

This lack of adequate situational awareness by power grid operators has been a leading contributor to power disturbances cascading into large blackouts (for example, the ones in the USA/Canada and Italy/Switzerland in 2003). Today, large grids generally have many sub-parts that do not coordinate their actions and have only very basic (and slow) communication between these sub-parts. This lack of situation awareness means that, as has happened in a number of blackouts in the last decade, one or two events such as a transmission line can happen an hour or two before the blackout occurs, but nobody has anything close enough to the entire "big picture" of the grid and thus the significance of these events that ultimately start the blackout is not understood by any person or computer. Such informational disconnects are inevitable, given that for examples the grids in North America have 3500 participants that can affect their stability, including the small company First Energy that was at the core of the 2003 blackout in North America [Blackout2003].

The rudimentary communications also means that opportunities for better protection, control, and efficiency are either impossible or far too expensive (typically with "one-off", hardcoded communications for each application family or even individual application). For more background, see [HBB05,BHG+07].

There are many chicken-egg problems involved with modernizing the power grid, including its data delivery infrastructure. In our opinion, the best way to solve many of these is to holistically and simultaneously consider power grids' dynamics and its data delivery infrastructure, both their steady states and those perturbed by a power contingency or a failure or cyber-attack involving the data delivery infrastructure. One key recent technology is those involving sensor data given microsecond-accurate timestamps then delivered in real-time to give a coherent picture of a grid for operators, and soon for closed-loop control and broader protection.

In this paper, we offer such a holistic view of the power grid and the use of such sensors, involving applied electric power and computer science researchers who have been looking at this problem in this manner for more than a decade. We first overview fundamental problems in the power grid that can be mitigated by such coherent, real-time data[4]: reliability, efficiency, and integration of renewable resources such as wind and solar. We then describe a wide range of applications utilizing such coherent, real-time data in order to mitigate these fundamental problems. We then normalize and summarize the communications requirements, including not just traditional quality of service (QoS) metrics such as latency and rate but also broader behavioral and configuration metrics, which we call "*QoS+*", including geographic scope, criticality, and amount of data. Next, we describe how these QoS+ requirements must necessarily impact the data delivery system of power grids, including absolute requirements and highly-recommended implementation guidelines. As part of this, we also compare how existing

[4] By *real-time data*, we mean data that is delivered with predictable latency and reliability, presumably low and high, respectively.

network-level technologies, middleware, and power protocols map onto those requirements and guidelines. Finally, we overview the decade-long research into the GridStat data delivery system and of the emerging NASPInet concept that GridStat has influenced.

Such a holistic treatment of power and IT dynamics is timely given the recent interest in the "smart grid". Indeed, a recent, future-looking article by highly regarded authors (including the father of synchrophasors) [HPR10] notes that five key technology areas (KTA) have been identified, and (**emphasis** is ours):

> **Foremost among these KTAs will be integrated communications**. The communications requirements for transmission enhancement are clear. Broadband, secure, low-latency channels connecting transmission stations to each other and to control centers will enable advances in each of the other KTAs.
>
> - Sensing and measurements will include phasor measurement data streaming over **high-speed channels**.
> - Advanced components, such as all forms of flexible ac transmission system (FACTS) devices, HVDC, and new storage technologies will **respond to control signals sent to address perturbations occurring in milliseconds**.
> - Advanced control (and protection) methods will include differential line relaying, adaptive settings, and various system integrity protection schemes that **rely on low-latency communications**.
> - Improved interfaces and decision support will utilize instantaneous measurements from phasor measurement units (PMUs) and other sources to drive fast simulations and advanced visualization tools that can help system operators assess dynamic challenges to system stability.
>
> Each of these elements will be applied to the modernization of the grid, at both the distribution level and the transmission level.

Finally, we note that this paper is intended for researchers and practitioners in both electric power and computer science. It is written such that power researchers and practitioners should gain some understanding of the issues that have to be addressed on the computer science side, and vice versa. As such, it cannot possibly get into great details of electric power research issues for a given problem, or on the computer science side discuss system and fault models and other systems issues in huge depth. Those topics are part of ongoing papers which are meant for one "side" or the other, but not "both" as this paper is.

2. Power Grid Problems and Goals

We now overview fundamental problems in today's power grids that can be mitigated by coherent real-time data.

2.1 Reliability

Evolving to a non-carbon based electrical infrastructure will require the handling of high penetrations of non-traditional generation sources which behave differently from the vast base of existing generators. The smart grid will have to be able to utilize these intermittent sources of generation without compromising reliability and efficiency. The ability to control and operate the grid with more precision will make it possible to do so.

The reliability of the power grid is its ability to deliver electric power from generation to load without disruption. Thus the grid must be able to withstand minor and major disturbances without losing any customers. Interruption of electricity supply is not only inconvenient to the user but it affects the overall economy (productivity) of the region.

Of course, reliability is usually enhanced by adding redundancy into the grid and providing enough margin for the loading of the grid. On the other hand, operating the grid at much lower levels than the limits, introduces inefficiency as the transmission system is not fully utilized. Thus there is always some compromise between reliability and efficiency, both of which have to be optimized.

The addition of intelligent analysis and control helps reliability in two ways. As the power system gets loaded closer to its limits, the monitoring tools can alarm the operator to limit violations and the analysis tools can alert the operator when the system is vulnerable to contingencies. The importance of these real-time functions was demonstrated during the 2003 USA-Canada blackout in which the system was getting more vulnerable over several hours but neither the alarming system nor the state estimator was working properly to alert the operator to deteriorating operating conditions.

The other way reliability can be helped is with control and protection schemes that can prevent the system from instability or collapse. Again, during the 2003 blackout in North America, once the last contingency occurred the cascading over several US states and one Canadian province happened too quickly for operator intervention. Under such circumstances, the only way to avoid such cascading would be to utilize fast control or protective schemes to isolate impacted areas and/or adjust some controllable values.

2.2 Efficiency

The efficiency of a power grid is its ability to minimize the cost of generation which is facilitated by the transfer of large amounts of power while incurring the least losses in the transmission system. Because the transmission lines have limits, the maximizing of efficiency requires a constrained and non-linear optimization problem which is done in the day-ahead hourly energy market as well as in real time.

The availability of fast controls in the grid also enhances the efficiency because these controls can prevent instability of the system, thus allowing higher rates of power transfer over the transmission system which raises the efficiency.

2.3 Renewable Integration

The smart grid has to accommodate renewable resources and in fact, the increased level of sensing, measurement and control can monitor the interruptible wind and solar resources and can quickly bring up backup generation to counteract the loss of wind or solar. At higher penetrations of wind and solar, there are other problems such as the fact that solar photovoltaics do not have any inertia, making the system more unstable.

The sensing and control in the smart grid can help this situation. Indeed, some renewable sources have very different power characteristics (e.g., wind does not provide reactive power, the inadequacy of which can lead to a blackout), and even the most experienced system operators at utilities have little to no experience with or understanding of how the power dynamics of renewable energy sources will impact the dynamics of their particular system, especially during rare (but potentially blackout-inducing) contingencies. Coherent, real-time data can thus help mitigate this lack of understanding, which is being exacerbated by the large number of experienced system operators retiring in recent years (and projected to retire in the coming years).

3. Techniques for Enhancing Generation and Transmission Based on Coherent Real-Time Data

Time-synchronized measurement devices are becoming a standard part of the power system and provide microsecond time accuracy using GPS-based clocks. Measurements called *synchrophasors* are increasingly being made in the power grid [PT08,MC08,TAB+08, CKT+09]. Synchrophasors are measurements that represent both the magnitude and phase angle of a 50 or 60Hz voltage or current waveform at a particular time synchronized to a system-wide reference such as a Global Positioning System (GPS) clock. A few years ago, synchrophasor technology was found only in stand-alone instruments called phasor measurement units (PMUs). Today, synchrophasor technology is also found in meters, protective relays, and fault recorders, which dramatically lowers the cost of implementing synchrophasor-based control and protection strategies. Station phasor data concentrators (PDCs), which gather synchrophasors from several sources within a substation, and distributed synchrophasor control devices are important new system components, providing distributed aggregation and control functions. Furthermore, new communication architectures, which include in-network data concentration, real-time distribution, and fast fault recovery provide an infrastructure with the necessary high reliability.

This section discusses applications that can bring increased reliability, stability, and security to the entire power grid, *if* communications are adequate. For each application family, we summarize its communications aspects, which are expanded upon in the rest of this paper.

3.1 State Estimation and Measurement

Knowing the system state is an important first step for reliable control of the power system [AE04]. In the electric grid, the state of the system is the voltage and angle of every bus in the system. Schwepp introduced the first state estimation system [SW70]. In traditional state estimators, the state is estimated from voltage (no angles) and power flow measurements using interative, nonlinear algorithms that do not always converge.

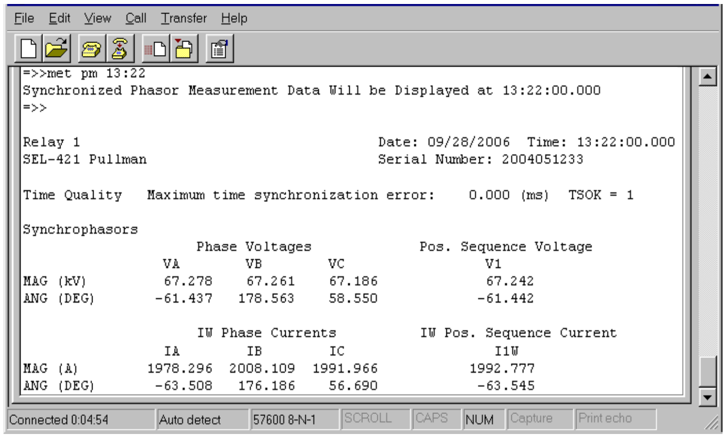
Fast state calculation is increasingly important for the quick response time requirements of control loops that are coupled to new dynamic renewable energy sources. Direct synchro-phasor measurement of the state at some buses allows use of faster state estimation techniques to be used to estimate the state at other buses. Synchro-phasor-based state estimation offers another advantage as well. State estimators must keep track of dynamic power system topology in order to correctly estimate the system state. Using traditional methods, all measurements are not taken at the same instant in time. Although voltage magnitudes change slowly with time, there is the possibility that the measurements taken could be from two different system configurations. For example, consider a case with a breaker opening. If some measurements are from before the breaker change and some are from after the breaker change, then effectively there are two completely different systems. A conventional state estimator might include both sets of measurements when attempting to solve the state and fail to converge. With time-synchronized measurements, the precise timestamps enable aligning all measurements, including contacts, disconnect switches, and tap changer values, so accurate system states can be calculated.

As will be discussed later, with new distributed synchrophasor control devices, once the state is calculated locally in a substation, it is easy to share across a wide-area network using the time-stamps resulting from the synchrophasor-based state calculation [SW07]. This improves power system reliability and meets the increasing reliability expectations from electric power customers.

We now describe a series of state estimation and measurement algorithms, building up from little to much required communications, that improve on the current practice.

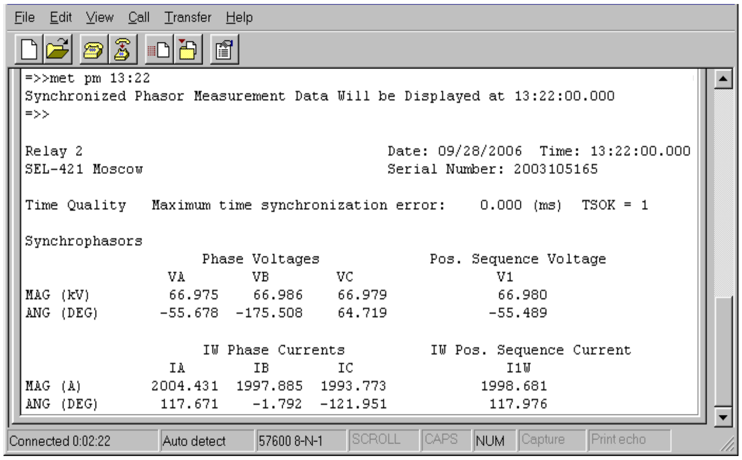
3.1.1 Baseline Configuration Testing Case

As a baseline case, it is interesting to note that a very simple state measurement application is immediately available with synchrophasor measurements and does not require any communication infrastructure. One common wiring problem that is difficult to detect during commissioning is rolling power system phases. For example, consider the case where the VA source is wired to the VB terminals, VB to VC, and VC to VA. Using a local PMU and simple terminal connection, a technician can immediately check for this condition because signal phases are referenced to a common time standard [SW08]. Figure 1 shows a display of two relays with PMU capabilities. Notice the time of initiation, 13:22, is identical for each. The phase angle difference between VA on Relay 1 and VA on Relay 2 shows that the phases are correctly connected with the phase shift due to the impedance of the line. Thus, synchrophasors can instantly provide these basic power system improvements to the power engineer, even without additional software applications, EMS systems, or communications infrastructure. The only tool required is a computer with a serial ASCII connection to a PMU. Figure 1 depicts this solution. (a) shows synchrophasor snapshot for Relay 1 using the Meter PM Command at 13:22; (b) shows synchrophasor snapshot of Relay 2 using the Meter PM command at 13:22.



```
=>>met pm 13:22
Synchronized Phasor Measurement Data Will be Displayed at 13:22:00.000
=>>

Relay 1                                   Date: 09/28/2006  Time: 13:22:00.000
SEL-421 Pullman                           Serial Number: 2004051233

Time Quality   Maximum time synchronization error:    0.000 (ms)  TSOK = 1

Synchrophasors
                     Phase Voltages             Pos. Sequence Voltage
                 VA        VB        VC                 V1
MAG (kV)         67.278    67.261    67.186             67.242
ANG (DEG)       -61.437   178.563    58.550            -61.442

                     IW Phase Currents          IW Pos. Sequence Current
                 IA        IB        IC                 I1W
MAG (A)        1978.296  2008.109  1991.966           1992.777
ANG (DEG)       -63.508   176.186    56.690            -63.545
```

(a) Relay 1



```
=>>met pm 13:22
Synchronized Phasor Measurement Data Will be Displayed at 13:22:00.000
=>>

Relay 2                                   Date: 09/28/2006  Time: 13:22:00.000
SEL-421 Moscow                            Serial Number: 2003105165

Time Quality   Maximum time synchronization error:    0.000 (ms)  TSOK = 1

Synchrophasors
                     Phase Voltages             Pos. Sequence Voltage
                 VA        VB        VC                 V1
MAG (kV)         66.975    66.986    66.979             66.980
ANG (DEG)       -55.678  -175.508    64.719            -55.489

                     IW Phase Currents          IW Pos. Sequence Current
                 IA        IB        IC                 I1W
MAG (A)        2004.431  1997.885  1993.773           1998.681
ANG (DEG)       117.671    -1.792  -121.951            117.976
```

(b) Relay 2

Figure 1: Result of ASCII command for two relays connected across a transmission line

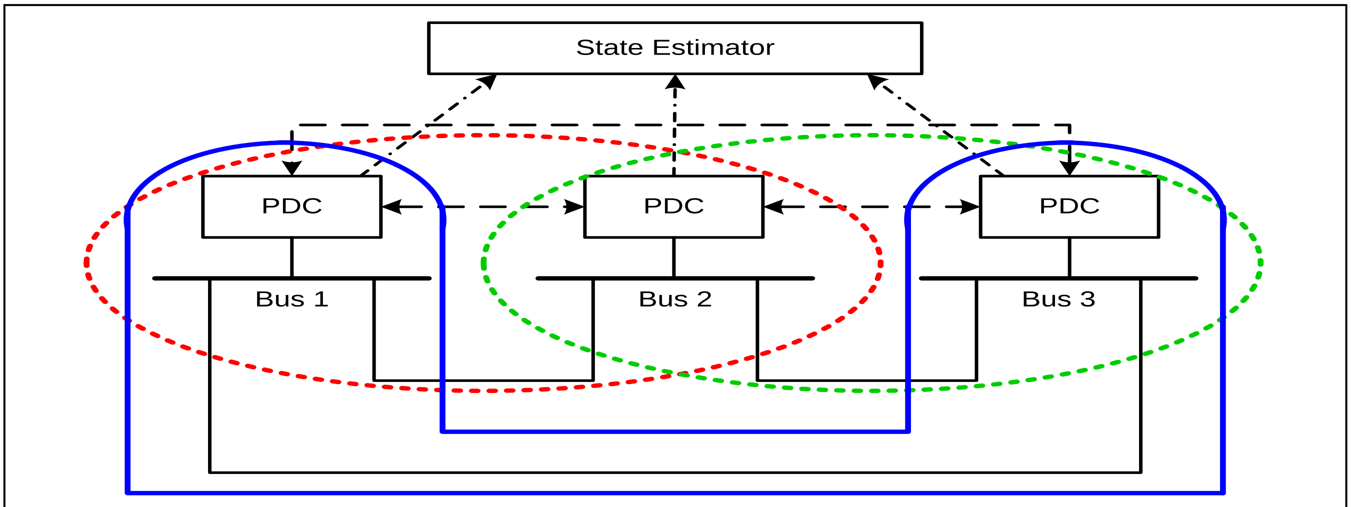


Figure 2: Distributed (Hardcoded) Peer-to-Peer Communications

3.1.2 State Measurement within or Near a Substation

Time-synchronized measurements also create the capability to calculate the state within substations and localized regions. Overall system state estimation is then a matter of aggregating and reconciling the local estimates. Figure 2 shows an example of how local coherent measurements can improve system reliability. (For simplicity, the PDCs are shown connected directly to the power system buses. In most systems the PDC connects through a PMU to the bus.) Each PDC, located at the substation level, collects the voltages, currents, associated phase angles, and electrical topologies of the system as required by the state calculation engine in the PDC. The data are also exchanged between the PDCs so that the state is refined based on measurements from adjacent substations. Although the communications here is relatively hardcoded and limited, even this has benefits. The data exchange provides redundant communication paths to a state estimator in the event that the primary communications channel is temporarily lost. Should direct communications be interrupted, an adjacent PDC forwards the data.

3.1.3 State Measurement across Substations

Figure 3 shows a two-level state estimator [YSB09] that more systematically exploits synchrophasor capabilities across a wide area. This estimator simplifies the total state estimation process by detecting and correcting topology and data errors early in the estimation process. It also can greatly lower the quantity of data that needs to be sent to a control center, with respect to sending all of the PMU data. This particular scheme uses only two levels, but there is no inherent reason that similar techniques could not be done for more levels of a hierarchy; e.g., substation, utility sub-region, utility, ISO/RTO, NERC.

3.1.4 Summary of QoS+ Requirements

The communications quality of service (QoS) aspects of these state estimation and measurement schemes can be summarized as follows. The inputs, consisting of power flows, voltage and current magnitudes and phase angles, are sent periodically. The required latency for these inputs is fairly forgiving because there is only a person in the loop; tenths of seconds or even seconds is adequate, and a rate of a few Hz or less suffices. The inputs are at the substation and the criticality of the inputs depends on the application using the estimated state. The output goes to the applications that require the estimated state and has much the same communication QoS requirements as the inputs do.

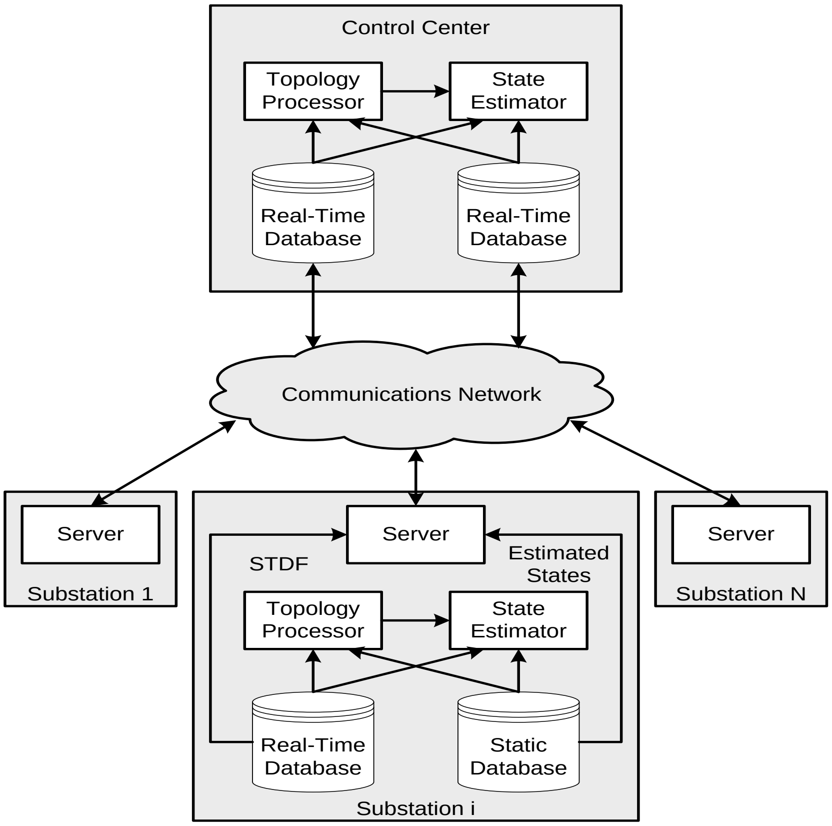


Figure 3: Distributed State Estimation (Two-Level)

3.2 Operator Displays

Operator displays are the primary window by which people monitor the operational state of the grid. Most existing operator displays update slowly based on data collected from a SCADA (Supervisory Control and Data Access) system every few seconds. These data are insufficient to reveal some crucial dynamic phenomena, such as oscillations, that indicate undesirable operating conditions. For example, with so much new renewable generation being connected to the power system, it is difficult to analyze the power system in sufficient detail to predict some of these oscillations, so detecting them when they occur is crucial. Unfortunately, oscillations might not be detectable from the slowly updating SCADA data. Presenting operators with results of analysis based on synchrophasor measurements made at much higher rates offers a remedy for this. Many systems have been described in the literature [MPA+04,TPZ+08].

The communication latency constraints for wide-area visualization are not strict: updates every few seconds are sufficient and displays can lag by a few seconds, given that there is a person in the loop. However, the quantity of data gathered with synchrophasor measurements is large because of the high sampling rate and measurements must traverse an entire utility or ISO (wide-area). Further, it is not critical that every measurement arrive, however, if there is a gap in communications, the operators may need up to the last several seconds retransmitted so they do not miss critical information during a fault or other problem. This kind of data transfer is different from some uses of sensor updates, where each update may be critical to deliver. We call this kind of transfer a bulk data transfer, and this application is labeled “Catch up for Operator Displays” Section 4.2.

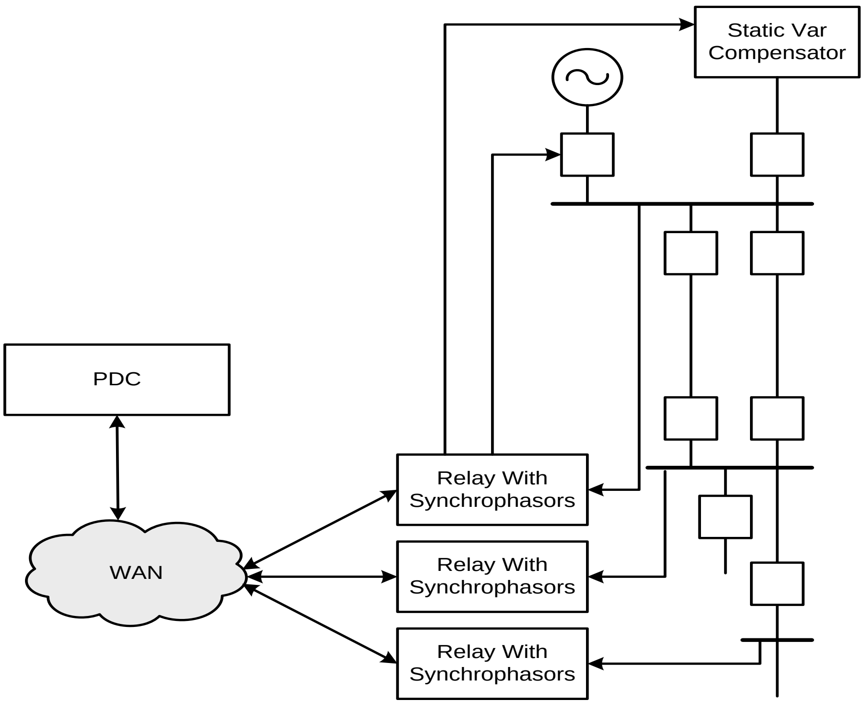


Figure 4: Distributed Wide Area Control Block Diagram

3.3 Distributed Wide-area Control

3.3.1 General Overview

Control of the power grid needs to be improved due to two major factors. First, the grid is inherently getting more stressed each year as increased demand and supply outstrips the addition of new long-distance transmission capabilities; there are more "miles times megawatts" being travelled each year. Second, renewable energy sources are much more variable, and their effect on the grid's stability much less known, than sources such as hydroelectric, coal, and nuclear with which operators and planners have greater experience.

This can be mitigated by moving from slower operator control towards more use of closed-loop feedback control. A generic architecture for such a control regime is depicted in Figure 4.

As an example, Southern California Edison has applied synchrophasors for wide-area dynamic voltage control by installing a PMU at the Big Creek Generation Station [JTT+07]. The PMU sends voltage measurements to the central control office where they are integrated into a static var compensator (SVC) controller. The SVC is located some distance away from the generation station. The SVC controller maintains local voltage at the SVC within proper operating range while simultaneously avoiding overvoltage at the generation station. The benefit of synchrophasors and PMUs for this system is not so much their time synchronization as it is their high speed and uniform sampling rate. The total measurement and communication latency requirement for this system was one second. This requirement was difficult to achieve with existing SCADA systems but relatively easy to achieve with time-synchronized phasors. Similar FACTS-based applications for HVDC systems and TCSC controllers are also proposed in [PT08].

Another control application is based on measuring power system modes—low frequency electro-mechanical oscillations at characteristic frequencies. System disturbances, such as generation shedding or line tripping, can excite a mode. These oscillations become more pronounced when wind generation is added to the power system

[KRK91]. When oscillations are well damped, the system returns to a stable state after the disturbance; however, lightly damped or negatively damped oscillations cause instability. Clearly the power system is never operated in an unstable mode and is designed with large stability margins. The system topology, however, can change in unexpected ways during a disturbance, which can lead to an unstable system. Because of the power system size and the complexity of new renewable generation, it is difficult to predict all possible topologies, parameters, and associated modes. This limitation can be circumvented, however: the uniform sampling rate of synchrophasor measuring devices unlocks the ability to directly calculate the frequency, magnitude, and damping factor of each power system mode.

3.3.2 Summary of QoS+ Requirements

For this family of applications, the input data are voltage and current. The allowable latencies for inputs vary from roughly 250 msec to a few seconds. The required rate for inputs varies, too: for voltage control it can be as slow as 1 Hz, while oscillation control may required 60 Hz. The input data delivery is critical, though missing data for up to a second does not cause problems given that this application family is handling fairly slow-moving phenomena. If input data are missing, there is no need to retransmit. The geography of inputs can vary widely, depending on the control scheme.

The output is a control signal to a reactive power controller or to a power system stabilizer, with latency requirements similar to those for the inputs. The rate of outputs can be slower than the inputs, because a control signal may only be needed every few seconds. However, a lower output rate makes the control loop slower so increasing it offers benefits in some configurations. The quantity of the output signals is small, though obviously goes up with increased output rates. The geographic scope of the outputs is similar to the inputs. Other possible control actions (outputs from the control application) include re-insertion of a transmission line, shedding generation, shedding load, or adjusting compensation devices such as shunt capacitors.

3.4 Distributed System Integrity Protection Schemes

3.4.1 General Overview

Distributed system integrity protection schemes are another class of wide-area control whose implementation is facilitated by ability to communicate synchrophasor data across the grid. A system integrity protection scheme (SIPS), also known as a remedial action scheme (RAS), provides the next level of protection after relays that respond to local power system emergencies [HNM+08]. One class of SIPS is contingency-based, where the scheme responds after a predefined event occurs, such as a breaker opening. Another SIPS design methodology is based on analog quantities such as frequency or voltage. With this design, the scheme responds if the frequency or voltage exceeds or drops below a threshold. For example, if an under-frequency condition develops, the system may shed load if the generation is unable to supply the required power.

3.4.2 Case Study: Using Synchrophasors to Respond to Angular Instability

A specific example of a synchrophasor-based RAS built to respond to angular instability is the Comisión Federal de Electridad (CFE; *México*) automatic generation shedding scheme as shown in Figure 5 [MJG+06]. In this scheme, synchrophasor measurements are taken at the Chicoasen and Angostura buses. The remote location of the Angostura generation station presents unique challenges for reliable system operation. The total hydroelectric generation capacity is 900 MW. The local load is less than 100 MW. The remaining power is transmitted over two long transmission lines to the national system and to which it also connects through a 115 kV network.

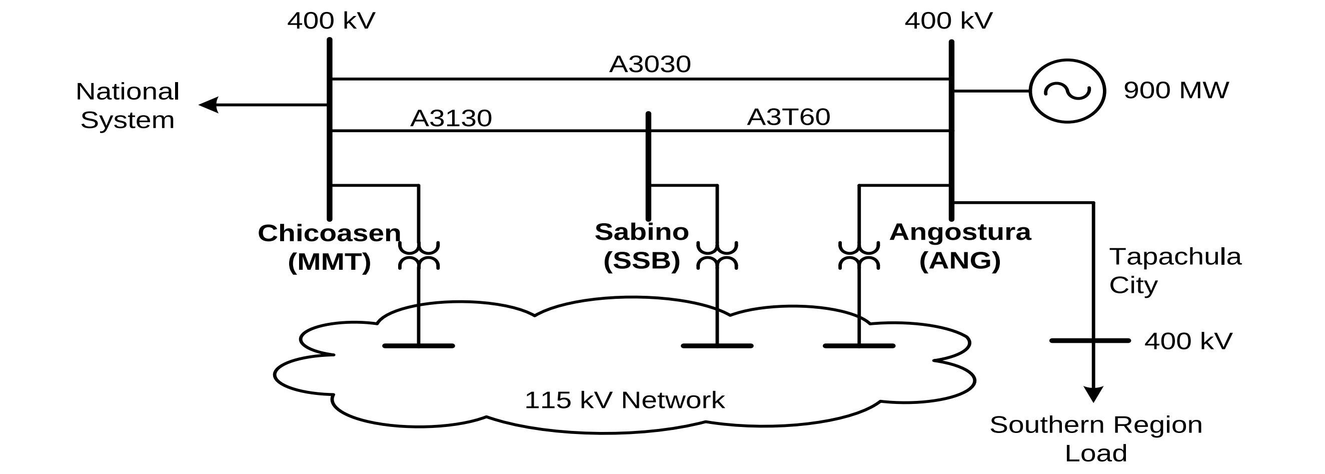


Figure 5: Block Diagram of the CFE Cystem.

If two of the lines between Angostura, Sabino, and Chicoasen are lost, the Angostura generators may experience angular instability. This is shown in where the angle separation accelerates. To prevent the 115 kV network from overloading, generation is shed.

A traditional RAS requires a breaker status signal at every bus. This system was simplified using a relay with time-synchronized phasors at each end of the line to measure the angle and compare the difference against a threshold.

System modeling using a Real Time Digital Simulator (RTDS) shows that a double-line outage results in an initial angle difference of 14 degrees; enough to cause system instability (see Figure 6). A single-line fault results in a difference of less than 7 degrees and does not cause stability problems. As a result of these studies, a threshold difference of 10 degrees was selected. Relays with PMU capabilities were placed at Angostura, Sabino, and Chicoasen. Each relay measures the local voltage and line currents and sends remote synchrophasor data to the other relays. The relays time-align and then compare the local and remote phase angles. If any of the relays detect that the load exceeds the maximum threshold of 10 degrees it will trip to prevent system instability. This scheme requires a data exchange of 20 messages per second, which is easily met by the 19,200 baud fiber-optic serial connection between the relays.

Figure 7 shows the result of the synchrophasor RAS responding to a double-line loss and tripping the generation after 100 ms from the point where the angle jumps from approximately 4 degrees to approximately 14 degrees. The system remains stable. Note the much smaller range of angles in this figure compared to that in Figure 6.

3.4.3 Summary of QoS+ Requirements

Inputs to a SIPS include voltage and current, breaker status, and power. Its communications aspects can be extremely challenging. The input rate is the highest of all of the applications considered, and the latency must be very low. The criticality of its inputs (and outputs) is extremely high. For example, a SIPS might be installed in order to transfer more energy over a line than it can handle under all contingencies. Therefore, if a contingency happens it will have to respond by curtailing generation or load: if the SIPS fails, the contingency can cascade into a blackout [WEB10]. The quantity and geographic scope for SIPS varies widely and is similar to that of distributed control.

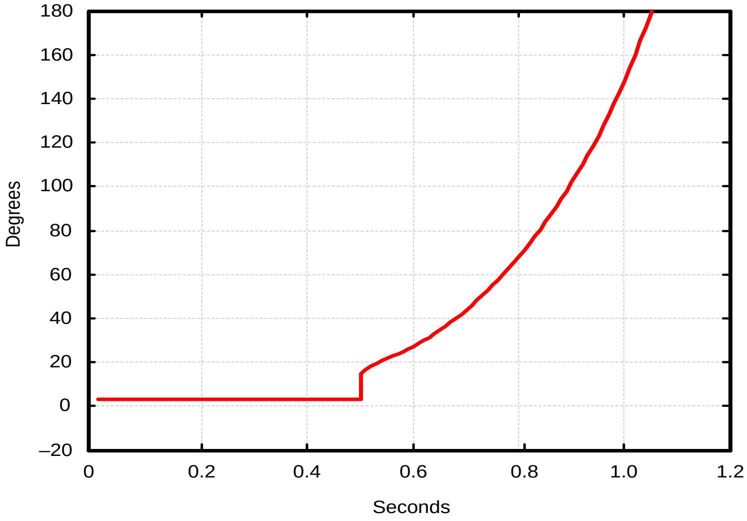


Figure 6: Angular Difference for a Double Contingency Condition (without an SPS)

Outputs from a SIPS are a condition-based control signal to initiate any of a number of actions to compensate for the contingency; e.g. tripping a breaker, generator or load. The outputs should be delivered with very low latencies. The criticality of the control actions is high though the quantity is low. The SIPS output control signals often have to be delivered less distance than the inputs because the SIPS logic tends to be located closer to the grid element that it is controlling

3.5 Synchronous Distributed Control

3.5.1 Baseline Scenario

Renewable generation is forcing more variability into the power system and one fundamental problem here is that when operators make system changes involving a number of compensatory control actions they do so by making one change at a time. This causes the grid to have unnecessary transient disturbances.

Synchronous distributed control [SWZ+09] can reduce variability and keep the system stable. This technology uses the distributed time signal that is already available to relays in the power system and ties that time to specific operator commands. The commands are issued to the relays in advance of the anticipated operating time and

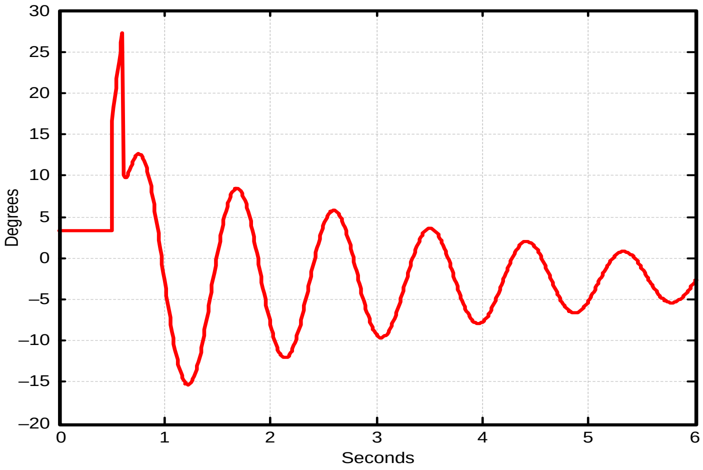


Figure 7: Angular Difference for a Double Contingency at 500ms and Tripping of the SPS 100ms later.

validated for accuracy to ensure they have not been maliciously compromised. Then, at the preselected time, they execute in precise coordination.

Consider a traditional scenario for taking a line out of service. In Figure 8, Lines 1 and 2 are part of the transmission network. Line 3 connects the transmission and distribution networks. Bus B4 is a distribution bus. The transformer between buses B3 and B4 is a mechanical on-load tap change transformer. Consider what happens when Line 1 is taken out of service.

First an operator sends a command to open breakers CB1 and CB2, decreasing the voltage at Bus B2 due to the increased impedance from the generator through the remaining Line 2. As a result of this voltage decline, the transformer between B3 and B4 taps up to restore the distribution voltage to its target levels. If the transmission voltage at Bus B2 decays to a value below the desired minimum, the operators may insert the parallel capacitor into the system. This raises the transmission bus voltage but then requires the transformer to tap back down in order to avoid exceeding the distribution bus target voltage levels. Figure 9 illustrates the system response to these changes.

These sequential operations result in unnecessary stress on the power system, which could potentially contribute to a more broadly cascading event if it happened at an inopportune time. Synchronous distributed control instead works as follows: the operator selects an appropriate *set* of commands to accomplish *all* of the desired changes; the commands are sent to a coordinator (e.g. a PDC) at each involved substation; the PDCs send appropriate subsets of the command list to Intelligent Electronic Devices (IEDs) to confirm that they are in states appropriate for carrying out the commands; after receiving confirmation from each IED that the sequences of commands are ready to run, the PDC indicates to the operator that the system is ready for initiation; the operator validates that all components are ready, no cyber security alarms have been received, and the change is still desired; the operator then arms the system and sends the start time to the PDC; the PDC and IEDs execute their commands at the preselected time.

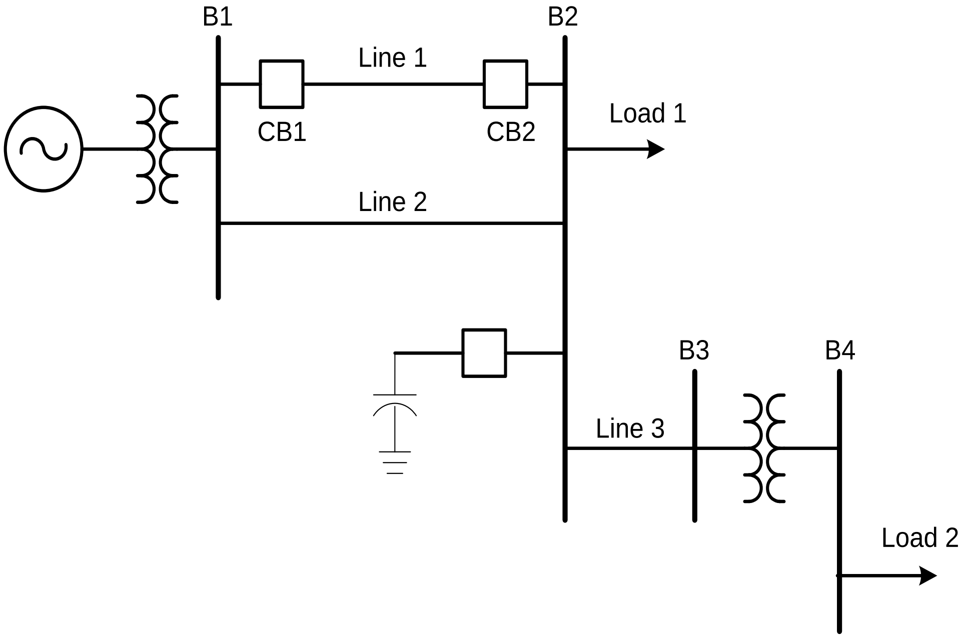


Figure 8: Power System Model to Analyze Synchronous Distributed Control in Real Time

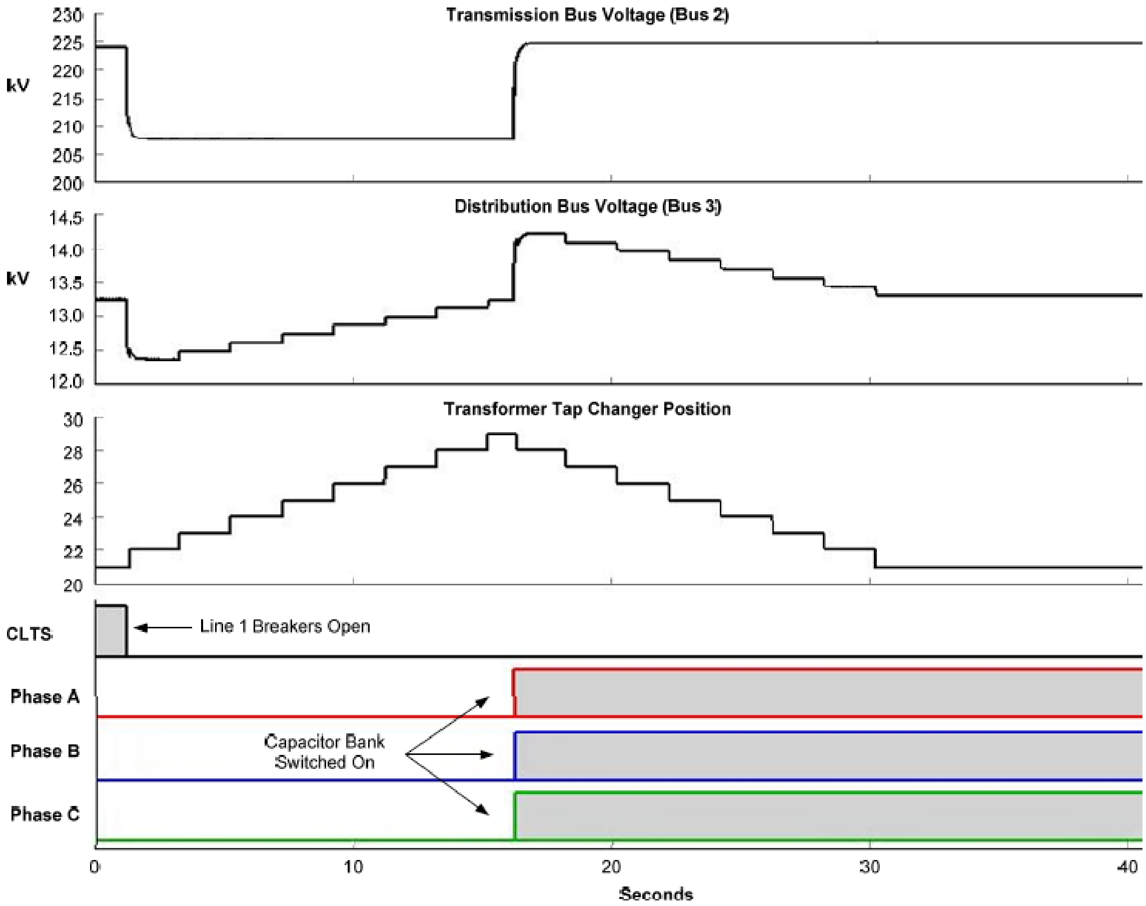


Figure 9: Effects of Uncoordinated Changes in Renewable Generation Production

Following are the concrete steps outlined above that are involved with taking a line out of service, and illustrates the distributed synchronous control technology using PDCs:

1. The operator selects an appropriate set of commands and sends them to the substation PDCs.
2. The PDCs send appropriate subsets of the command list to each IED in the system.
3. Each IED returns its status to ensure that it is ready and that the equipment does not have active diagnostic failures.
4. After receiving confirmation from each IED that the sequences of commands are ready to run, the PDC indicates to the operator that the system is ready for initiation.
5. The operator validates that all components are ready, no cyber security alarms have been received, and the power system is in its appropriate state.
6. The operator then arms the system and sends the start time to the PDC.
7. The PDC and IEDs execute the sequence of cmds at the preselected time & precisely the same instant.

This precision is possible only by using time-synchronized technology and results in minimal system impact as shown in Figure 10. The reduction in transients improves reliability and leaves additional margin for the uncontrollable dynamics of renewable sources.

The synchronized control system also improves overall safety. Local processing at the substations, for example in a PDC, will question a set of control commands that calls for vital actions, e.g., circuit-breaker tripping or reactive power insertion. Setting the control commands for a future time allows an interval when these commands can be

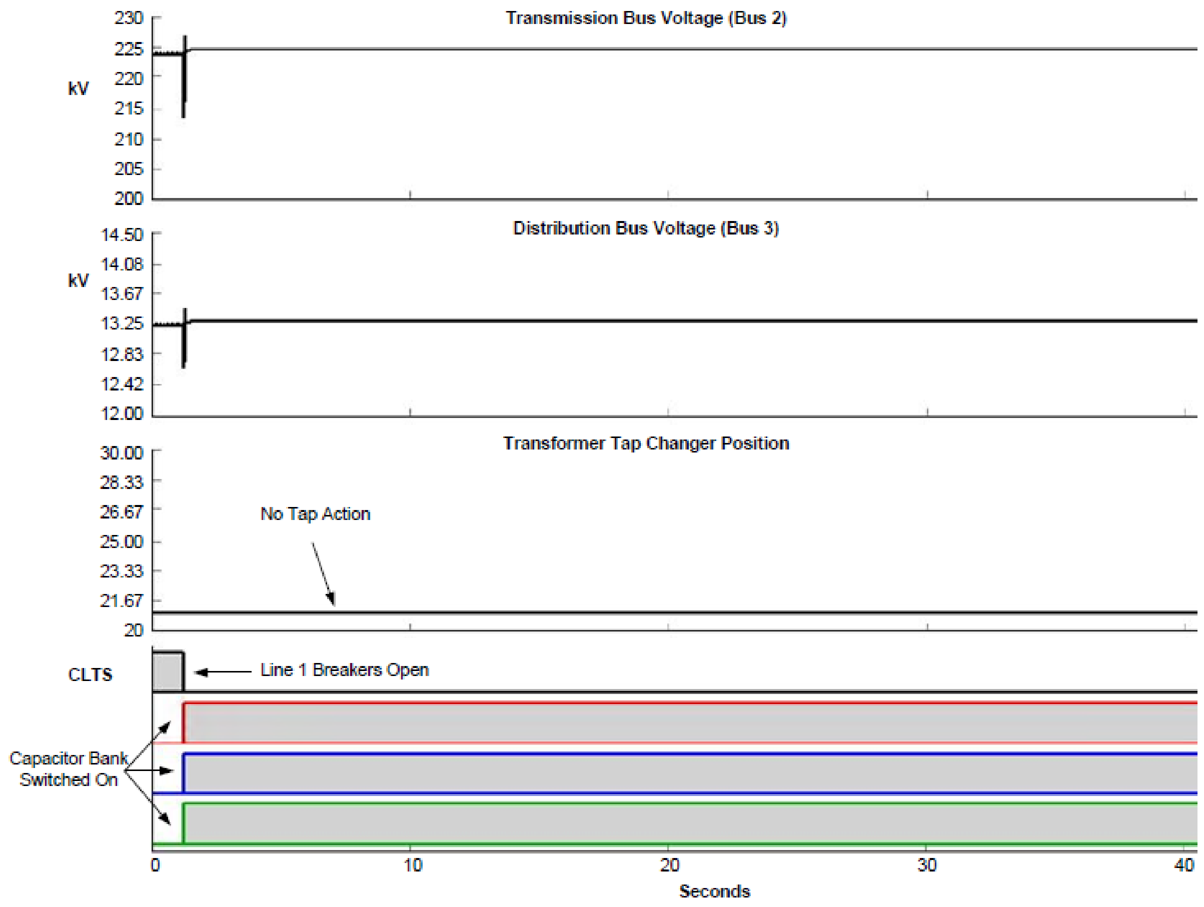


Figure 10: Effects of Coordinated Changes on Renewable Generation Production

re-analyzed for proper function. A distributed synchrophasor control device requests control validation from the system operations center or source of the synchronized commands. A local logic engine uses contingency analysis to validate the requested operation, such as confirming that opening a circuit breaker will not result in unacceptable voltage drop or even a voltage collapse. The system includes an alarm to alert the operator when a new series of controls is initiated. Only after validating the commands will the operator arm the system to execute at the desired time. .

3.5.2 Summary of QoS+ Requirements

The communications and related requirements for this application family are quite modest. See Section 4.2 for more details.

3.5.3 Enhancement Possibilities using Advanced GridStat Mechanisms

We note that such synchronized distributed control can be enhanced by using GridStat’s remote procedure call (RPC) mechanism [VBG+10], described further in Section8.5.2. This RPC mechanism has two grid-specific enhancements compared to traditional remote procedure call mechanisms. First, when a message containing a command to change an actuator (or other) setting arrives at the device, a check can be done based on live sensor variables, and if the check fails the call will be aborted before it is sent to the actual device. This can enhance the safety of field personnel by ensuring that a line that should not be energized (i.e., the operators and EMS software thinks it is not in service) is not energized and hence dangerous to field personnel. Further, it would be straightforward to enhance this RPC with a two-phase (or even three-phase) commit familiar from the database transaction world [BHG87], in order to provide higher consistency and broader coverage of failures and errors.

The second mechanism gives more assurance to the client application that initiated that initiated the actuator call actually succeeded. Here, when the reply returns to the client indicating the new status of the actuator, a check on live sensor feed data can be performed (after a programmable delay). This allows the power engineer to program a simple predicate that checks that the physical actuator (or other device) actually did change. Such a predicate would check a live sensor variable that should have changed if the physical actuator actually did what it was commanded to do. Should this predicate evaluate to false, the program or operators can be alerted in order to investigate the status of the actuator. Should this predicate evaluate to true, this gives the application and operators greater confidence that the desired change occurred, because a fundamental limitation of general distributed computing systems is that reply messages can be dropped (though hopefully only rarely in a WAMS-DD!) and thus the client application cannot be sure if the request call reached the device ("server" in client-server parlance). This mechanism, a form of a *physical feedback loop*, is also described further in Section 8.5.2

3.6 Renewable Generation Islanding Control

3.6.1 Overview of Control Regime

Yet another use of synchrophasor technology coupled with communications arises in *controlling renewable generation* itself. Two important cases are generation frequency control when distributed generation is islanded and generator isolation when islanded. Figure 11 depicts a generation frequency control application in use by Abbott Pharmaceuticals that uses synchrophasor technology to control an islanded system [SWZ+09]. When the plant and utility systems are connected, the grid controls system frequency and the governor controls generator power output. During an islanded state, the governor is switched to isochronous mode for frequency regulation.

When measurements at the two relays are not time-aligned, angle measurements are not available, so frequency data alone must be used to determine grid connection. The most difficult detection condition is when the island frequency and system frequency are nearly the same. Angle data helps disambiguate the situation in this case; furthermore, angle data is essential for safe reconnection after islanding occurs.

A similar detection challenge also faces a solar photovoltaic system (PV). Presently the IEEE 1547 Standard, "Interconnecting Distributed Resources With Electric Power Systems" specifies that the source must disconnect from a locally islanded system within two seconds. Such a requirement is important for safety reasons, quality of

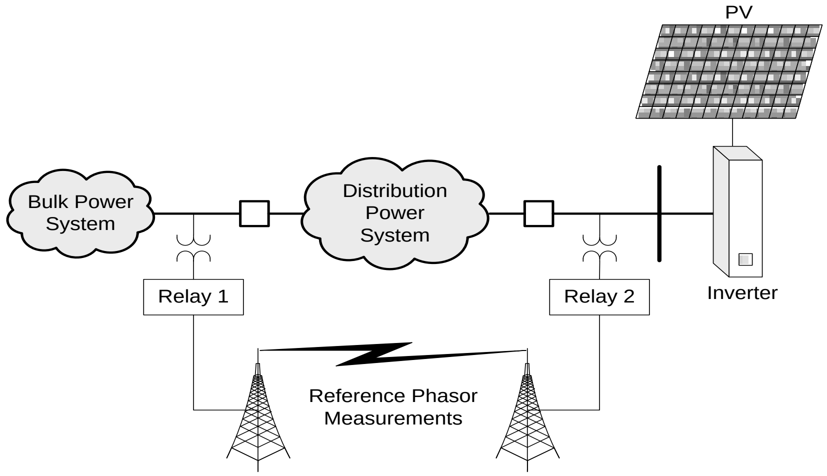


Figure 11: Anti-Islanding Scheme Using Relays, Synchrophasors, and an Inverter

power, and out-of-phase reclosing avoidance. However, as the density of PV power increases, forced islanding reduces power system reliability. In the future, it will be important to keep PV online during power system events because the large quantity of generated power will help keep the system stable. For this reason, it is important to find better control methods that scale with the growing generation of PV power.

IEEE 1547 also requires disconnecting for sagging voltage under high demand. With a small amount of generation, this requirement is reasonable, but disconnecting a high-density solar generation source will cause the low-voltage condition to accelerate. Synchrophasors enable a wide-area view of the system and therefore enable solutions that can keep distributed generation, such as PV, online during transient conditions. Further, if the output control signals can be delivered with very low latency then reclosers can be made faster which can greatly reduce power quality problems in a disturbance.

3.6.2 Derivation and Performance of the Islanding Scheme

The traditional approach to islanding control uses either local voltage and frequency information or breaker status to determine if the frequency or voltage magnitude is outside thresholds set by planning and engineering These methods have difficulty detecting an island and responding quickly if the power mismatch between the islanded source and the local load is small. Also, this approach can use many communications channels, causing overall poor reliability [MSM+09]. Using synchrophasor technology, islanding control for a PV system is set up as shown in Figure 12 [SWZ+09]. The relays include PMU capabilities and are connected by a wireless link. The solar PV is connected through a breaker to the distribution power system and then to the bulk power system. Both relays acquire voltage phasor measurements locally. Relay 1 then sends the synchrophasor values $\angle V_k^{(1)}$ to Relay 2 at a rate of 60 messages per second. Relay 2 receives the remote synchrophasor values and calculates the angle differences between the remote and local $\angle V_k^{(2)}$ values.

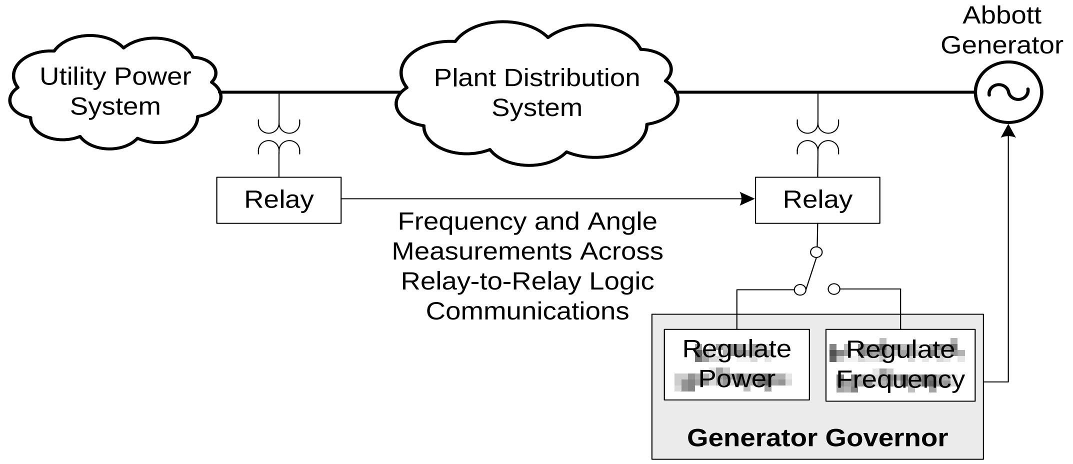


Figure 12: Frequency and Angle Measurements across Relay-to-Relay Communications Link

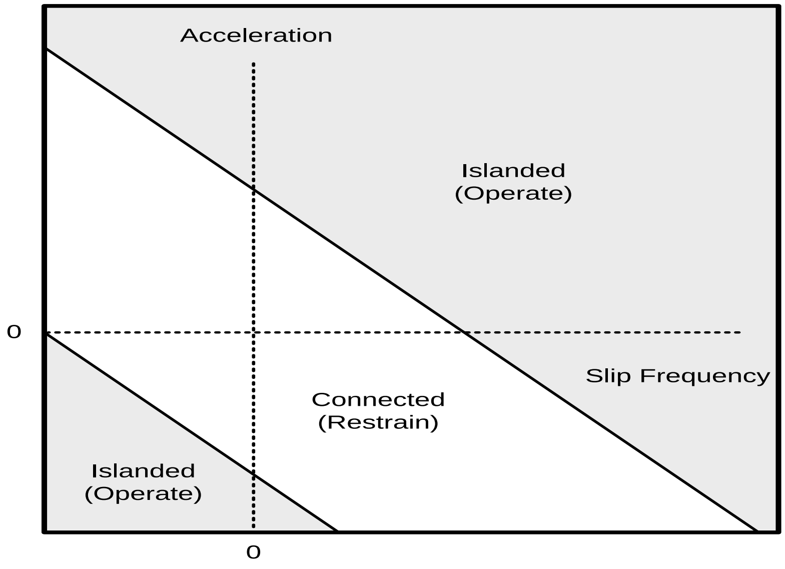


Figure 13: Islanding and Connectedness Based on Acceleration and Slip Frequency

The angle difference between the relays is defined as δ_k in (2). The change of δ_k with respect to time defines a relative slip frequency, S_k (3), where MRATE is the synchrophasor message rate. The change of slip frequency with respect to time, measures the acceleration between the two terminals. This value is defined as A_k in (4).

$$\delta_k = \angle V_k^{(1)} - \angle V_k^{(2)} \qquad (2)$$

$$S_k = \left(\delta_k - \delta_{k-1}\right)M \quad R \qquad (3)$$

$$A_k = \left(S_k - S_{k-1}\right)M \quad R \qquad (4)$$

Combining slip (S_k) and acceleration (A_k) results in the island detection phase diagram shown in Figure 13.In steady state, the slip and acceleration are at the origin. When an island condition occurs, slip and acceleration are possible, and either can push the phase into the operate region of the phase diagram. Thus, the system is connected when the slip (Equation 3) and Acceleration (Equation 4) are within the Connected (Restraint) region of the diagram.

3.6.3 Summary of QoS+ Requirements

Most of the communications aspects for this islanding scheme, including inputs and outputs, are quite similar to those of SIPS, described above. The inputs can be considered to be very critical, but we note that often there will be local anti-islanding schemes deployed in case the communications fail. These local schemes are not as good because of their uncoordinated operation can cause power quality problems. Another strategy would be to automatically trip (disconnect) the distributed generation source if communications fail. This is reasonable if the amount of distributed generation (including renewable energy sources) is small.

3.7 Overview of Transient Stability and Ancillary Services

Wide area distributed contols also have roles to play in improving *transient stability* and providing *ancillary services. Transient stability* is a problem with many power systems in which the transfer limit on some transmission corridors are limited by the fact that short-circuits make the system unstable. Controls of various kinds – shedding load and/or generation, changing SVC or HVDC settings, etc. – are used to mitigate these instabilities, thus allowing higher limits on the transmission corridor. The main difficulty is that the instability occurs quite fast – within a second – thus requiring any control action to take place within 50 to 100 msec to maintain stability. Such fast controls are not possible without having a high bandwidth, low latency communication system.

Ancillary services is a catch-all term for services other than energy that are needed for the operation of the grid. The number and type of ancillary services tend to differ from region to region depending on how the local organizations decide to define these services for setting up markets, contracts and billing. In general, there are always some ancillary services for the provision of capacity reserves, for balancing generation with load, and for voltage control. The *load balancing service* requires a closed loop control that increases or decreases generator outputs to follow the changes in load. This closed loop control uses measurements of frequency, tie-line flows and generator levels as inputs every 2 to 4 seconds and sends output signals to the generators. *Voltage control* is often local but some regions are using area-wide voltage coordination which requires communications similar to that of load balancing. The communications requirements for these services are modest and can be handled by present-day bandwidths connecting the control center. They establish the low end of the range of requirements for real-time control. Other roughly similar applications that have wide geographic scope include Dynamic Line Rating and Central Excitation Control [PT08].

3.8 Automated Contingency Drill-Down

Present best practices are for an operator to have a list of checks and actions prepared when a contingency is approached or reached. However, a data delivery system with the properties described later in this paper can greatly enhance the way that operators deal with an emerging or existing contingency.

In such a contingency response, operators will automatically be provided with additional data relevant for emerging contingency. This may include seeking data at a higher rate or new data from different locations than those used in normal operation. Additionally, new operator display windows can be automatically created to display this new data. This enables the operators to rapidly “drill down” into the emerging contingency, and hopefully prevent it from cascading.

Today, power engineers conduct contingency analysis studies to create the aforementioned operator checklists. However, the same analyses could easily be used to describe which new data should be subscribed to and how it should be depicted in different display windows. The GridStat system, described later, provides *mode change mechanisms* to provide exactly the agile data response outlined above [AGB+09]. These mechanisms exploit the fact that the data needed for an emerging contingency are known far ahead of time (during the study), and rarely would random (or even unplanned) variables be requested in the fact of an emerging contingency. This enables the modes to be switched (for example, from “steady state” to “approaching Contingency 31”) in a second or less in wide-area configurations.

Such mode change mechanisms are useful not only in responding to anomalies in power dynamics, but also can similarly be used to reply to either accidental IT failures or malicious cyber-attacks. In both cases, given systematic monitoring in the data delivery system, the effects of those failures/attacks can potentially be avoided. Such a strategy could, for example, throttle down the data delivery of noncritical applications (or ones not germane to the emerging power or IT contingency) to free resources to enable the adding of additional delivery paths for critical data.

The data delivery requirements for automated contingency drill-down applications are mostly similar to those of operator displays (though we do note that such drill-down techniques would be useful in conjunction with SIPS).

3.9 NASPInet Non-Real-Time Delivery Applications

The control, protection, and visualization strategies presented earlier in this section all had real-time requirements on the delivery of a single message. For completeness, we conclude this section by noting two different classes of traffic that are planned for the future NASPInet [Nor08], because of their implications for data delivery mechanisms as explained later in this paper (the complete list of NASPInet classes is summarized in Section 7.3). Both of these classes involve bulk transfer of data, compared to the rate-based sensor inputs or the rate-based or condition-based outputs described previously. They are similar in nature to the “Catch up” data for operator visualization described in Section 3.2. In short, they have no sub-second data delivery requirements for a given senor update, but it is important that the data transfers in this class happen in a reasonably predictable amount of time.

An *event* is a serious disturbance in the power grid, usually leading to a blackout at some scale. Utilities are required to log key sensor data in a database, typically called a historian, so regulatory authorities, such as NERC in North America, can ascertain the root cause of the problem. *Post-event data transfer*, then, involves transferring key related database entries for an event. The data messages of the transfer need not have any kind of latency guarantees, because the post-event analysis will be conducted offline. However, it is important to be able to transfer a reasonable amount of event data within a few hours or at most a few days. Of course, if the size of the required dataset is too large, it may not be possible to do this without interfering with important real-time data. But some communications resources must be reserved for this: arguably one of the most important post-event applications is *model validation*, which clearly must be supported.

Another class of bulk transfer is for *research purposes*. Power researchers need access to access actual data in order to validate proposed new control and protection strategies. NASPInet will also be used for this background traffic. However, research traffic is handled on a best-effort basis when there is spare capacity (as will usually be the case if there are no power or IT disturbances). Unlike post-event data, there is no attempt to provide even soft guarantees on a completion time. However, when there are presently no power contingencies or IT anomalies, we believe that, in practice, much useful research traffic will be supported.

4. Power Application Requirements Mapped to Data Delivery Service Requirements: Where Power and IT Dynamics Meet

The power applications described in the previous section have a wide range of data delivery requirements in many dimensions. In this section, we summarize those requirements to show the breadth of the requirement space and introduce the idea of a data delivery system for wide area measurement (henceforth WAMS-DD). The information in this section is key information that engineers who plan, build, and manage the data delivery infrastructure need to know in order to provide the required guarantees. Collecting and exploiting this information is not common practice in today’s power grids; in our experience, there tends to be an assumption that the communications will be “good enough.”

Table 1: Normalized Values of QoS+ Parameters

Difficulty (5:hardest)	Latency (msec)	Rate (Hz)	Criticality	Quantity	Geography	Deadline (Bulk traf.)
5	5–20	240–720+	Ultra	Very High	Across a grid or multiple ISOs	<5 sec.
4	20–50	120–240	Highly	High	Within an ISO/ RTO	1 min.
3	50–100	30–120	Medium	Medium	Between a few utilities	1 hr.
2	100–1000	1–30	Low	Low	Within a single utility	1 day
1	>1000	<1	Very Low	Very Low (serial)	Within a substation	>1 day

4.1 Normalizing WAMS-DD Parameters

We now present these requirements in a qualitative form, normalized to indicate the level of difficulty, where 5 means most difficult and 1 means least challenging to provide. This normalization enables intuitive comparison of different properties that have very different ranges, to get a sense of the wide ranges of difficulty or easiness involved for different power applications. It is important to note that a given application will not have all of its values in the same row: some requirements for a given application may be quite stringent (e.g., ultra-low latency) while others may be more forgiving (e.g., low volume of traffic).

Table 1 provides representative values of these data delivery requirements:

Latency: what latency is required for the delivery?

Rate: at what rate does/should the input be delivered, both now and in the future?

Criticality: how critical is this input [Ele04]? I.e, what is the severity of the consequences if data are not delivered for a short period of time?

Quantity: how much data needs to be delivered?

Geography: how far does the data have to travel?

Deadline: for bulk data transfer (defined shortly), when does the transfer have to be completed?

As noted earlier, some of these parameters are called *quality of service (QoS)* by networking researchers. We denote this entire collection, then, as *QoS+* to indicate that it includes other information needed in communications system design. QoS+ also refers to cyber-security issues, though these are beyond the scope of this paper.

Table 2: Diversity of Data Delivery of Selected Power Application Families

a		Traditional State Estimation	Direct State Measurement	Operator Displays	Catch Up for Operator Displays	Distributed Wide-Area Control	Distribute SIPS	Synchronous Distributed Control	Renewable Gen. Islanding Control	Transient Stability	Ancillary Services	Automated Contingency Drill-Down	Post-Event (Analysis	Research
Paper Section		**3.1**	**3.1**	**3.2**	**3.2**	**3.3**	**3.4**	**3.5**	**3.6**	**3.7**	**3.7**	**3.8**	**3.9**	**3.9**
Loop Entity		**P**	**P**	**P**	**P**	**C**	**C**	**C**	**C**	**C**	**C**	**P**	**P**	**P**
In[i] **(most difficult input)**	**Kind**	**SS**	**SS**	**SS**	**Co**	**SS**	**SS**	**SS**	**SS**	**SS**	**SS**	**SS**	**Co**	**Co**
	Lat.	**1–2**	**1–2**	**1**	**1**	**2–4**	**4–5**	**2–4**	**2-3**	**5**	**1**	**1**	**1**	**1**
	Rate	**1–2**	**1–2**	**2–3**	**—**	**2–3**	**5**	**1-2**	**2–3**	**—**	**—**	**2–3**	**—**	**—**
	Crit	**1-5**	**1-5**	**1-5**	**1–2**	**5**	**5**	**5**	**4-5**	**5**	**1–3**	**5**	**1-5**	**1-5**
	Quan	**3–5**	**1–2**	**3–5**	**1–2**	**3–5**	**2–4**	**1-3**	**1-3**	**1–2**	**1–5**	**3–5**	**5**	**1-5**
	Geog	**5**	**1-5**	**5**	**5**	**1–5**	**1–5**	**1–5**	**2-3**	**4-5**	**3–5**	**3–4**	**3–5**	**3–5**
	Dline	**—**	**—**	**—**	**5**	**—**	**—**	**—**	**—**	**—**	**—**	**—**	**2–3**	**1**
Out[j] **(most difficult output)**	**Kind**	**SS**	**SS**	**SS**	**Bu**	**Co**	**Co**	**Co**	**Co**	**Co**	**SS**	**SS**	**Bu**	**Bu**
	Lat.	**1–2**	**1–2**	**1**	**—**	**3-5**	**5**	**3-5**	**3–5**	**5**	**1–2**	**1**	**—**	**—**
	Rate	**1–2**	**1–2**	**1**	**—**	**—**	**—**	**—**	**—**	**—**	**1–2**	**2–3+**	**—**	**—**
	Crit	**3**	**3**	**3**	**1–2**	**5**	**5**	**5**	**5**	**5**	**1–3**	**5**	**1–2?**	**1**
	Quan	**3–5**	**1–2**	**1**	**2–4**	**1-2**	**1**	**1**	**1**	**1**	**1**	**3–5**	**5**	**5**
	Geog	**1–2+**	**1–3+**	**1**	**1–2+**	**1–5**	**1–5**	**1–5**	**2-3**	**3–5**	**2**	**3–4**	**5**	**5**
	Dline	**—**	**—**	**—**	**5**	**—**	**—**	**—**	**—**	**—**	**—**	**—**	**2–3**	**1**
NASPInet Class		**—**	**B**	**D**	**—**	**B**	**A**	**A**	**A**	**A**	**A**	**D**	**C**	**E**

4.2 Comparing Normalized WAMS-DD Parameters for Selected Power Applications

We now use the classes identified in Section 4.1to summarize the QoS+ requirements for the power applications described in Section3. This is depicted in Table 2.

The columns of this table are the different applications. The rows are the QoS+ attributes of the application's data delivery requirements along with three other kinds information about the application:

Loop Entity: Where does the app's output go: a person (**P**); or a computer (**C**)?

Inputs and Outputs: for the inputs and the outputs is the data delivery:

SS: streaming sensor updates;

Co: condition-based i.e., aperiodic events triggered by some condition; or

Bu: bulk data transfer?

Note that the inputs and outputs for a given application can be different; for example, it can take in **SS** updates but only emit an output when those inputs show a certain condition (**Co**). Also note that **Co** and **Bu** inputs and outputs do not have a delivery rate and that a **Bu** input or output does not have a required latency (which in this table represents a per-message guarantee), but it has a (soft) deadline (which no other kind of data has).

NASPInet Class: what service class is this kind of traffic (see Section 7.3).

It is crucial to observe in Table 2 that the requirements on WAMS-DD of even this small set of applications have great diversity. This means that the data delivery requirements for the power applications of an entire grid are very broad, and many different kinds of traffic have to be managed in order for each application to receive its required delivery guarantees. That is, this is exactly the opposite of “one size fits all” regarding data delivery!

Further, we note that the dynamics of the power grid can be affected by the dynamics of the data delivery. This is something which has rarely been studied (some examples we know of are [BTB04,NKM+07]), but needs to be developed as a best practice in the future. Otherwise, instabilities in the WAMS-DD may destabilize the power grid. Finally, we note that ***electric power researchers and practitioners are not used to reasoning about their applications in these terms***. However, it is crucial that they begin to**:** ***this is exactly the information needed by the designers, implementers, and deployers of a WAMS-DD*** in order to ensure a WAMS-DD meets its requirements yet is not massively over-provisioned (and thus potentially unaffordable or at the very least, extremely wasteful).

We now examine what these data delivery requirements are in greater detail, along with issues involved with implementing them.

5. Coherent Real-Time Data Delivery Enabling These Applications

5.1 System Model

Figure 14 depicts the architecture of WAMS-DD. Application programs or firmware that emit a stream of updates are called *publishers*, which are denoted Pub_1 through Pub_N in the diagram; Pub_1 for example outputs updates to variables X and Y. Applications which receive these updates are called *subscribers*, denoted Sub_1 through Sub_N. In the diagram, Sub_1 subscribes to Y from Pub_1 and W from Pub_N.

In the usual case in publish-subscribe (pub-sub) systems, neither publisher nor subscriber need to know about each other: they are decoupled such that they only know about the variable they publish or subscribe to, and how to contact the delivery system. In cases where the subscriber requires confirmation that the update came from its legitimate publisher—which may be common with WAMS-DD—data integrity techniques from the computer security field can be utilized by the data delivery system.

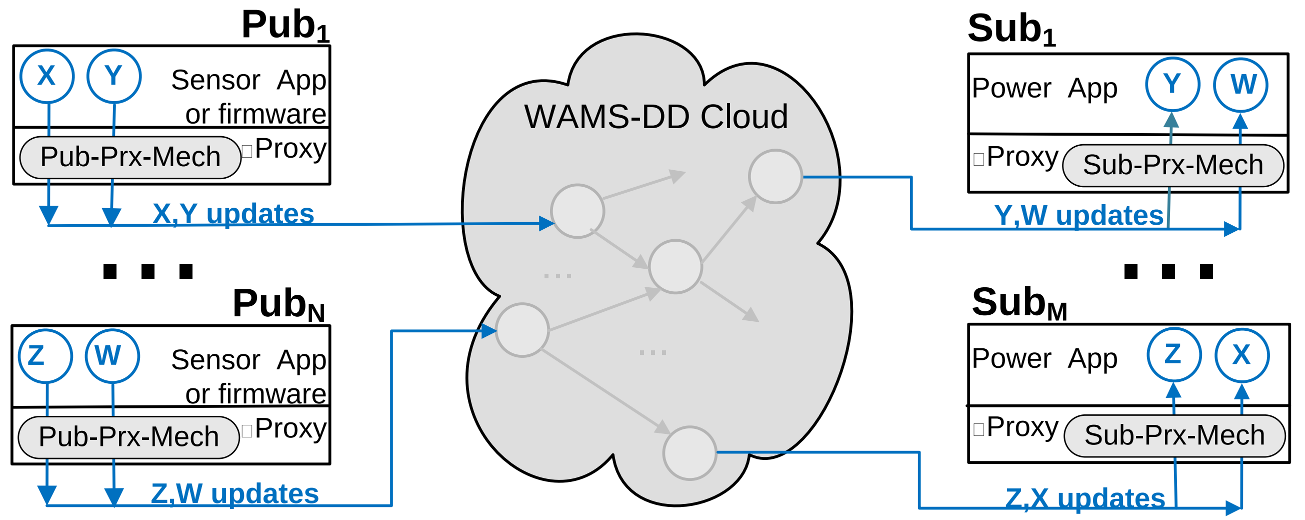


Figure 14: Architecture and System Model of WAMS-DD

Creating a publish-subscribe delivery path requires two steps. Publishers register their variables with the delivery system (only once per variable, not once per subscriber), and subscribers request a subscription to a given variable. For both publishers and subscribers, the delivery system returns a handle to a piece of code called a proxy, which is generated (at compile time) by the data delivery middleware. This proxy contains logic provided by the data delivery service, which, besides doing the usual middleware proxy activities such as packaging of the parameters into a message, is also a place where data delivery mechanisms may reside. These proxies can be used in different ways, as shown in Section 5.4.2.1.We denote a publisher-side proxy as *Pub-Prx-Mech* and the subscriber-side proxy as *Sub-Prx-Mech*.

After the variable is registered and subscribed to, updates to variables flow from publishers to subscribers, as shown in blue in Figure 14. To do this, they traverse what we call the WAMS-DD Cloud. This is opaque because, as shown later in this section, it can be implemented in different ways resulting in different tradeoffs. For the purposes of our system model, the WAMS-DD Cloud consists of a graph where the edges are network links and the nodes contain forwarding mechanisms can forward a message on its way towards a subscriber.

Updates from a publisher of a sensor variable thus traverse one or more paths to be delivered to a given subscriber. Along a given path an update may be delayed (so its required delivery latency cannot be met) or dropped due to failures in a network link or forwarding node, or to a cyber-attack. However, the probabilities of an update not meeting its delivery requirements can be held extremely low by carefully controlling the WAMS-DD, and by allocating multiple paths for important updates. That is, a WAMS-DD can be constructed so that the ontime delivery probability is very high, so long as its ***design constraints*** are met. Informally, these include forwarding capacity per node, maximum link traffic, number and kind of benign failures and cyber-attacks, etc.

We now overview the delivery requirements in Section 5.2, and then in Section 5.3 describe implementation guidelines that can be used to meet these delivery requirements with extremely high probabilities. In practice, we believe this can be done better than the typical practice of using dual isolated networks for critical protection applications, while at the same time supporting many more application families with thousands of update flows. However, such delivery technologies clearly need to be proven in the field before any migration to them can begin to be contemplated. Further, the techniques described in Section 5.3 are significantly better than renting an MPLS circuit from a telecommunication provider, because, as shown in Section 5.4.1, MPLS falls quite short on even the minimal delivery requirements for WAMS-DD.

5.2 Delivery Requirements for WAMS-DD

We now overview *delivery requirements* (DR) that WAMS-DD must meet [BHG+07,Nor08], not including the details of cyber-security related ones.

Requirement 1. *Hard, end-to-end (E2E) guarantees* must be provided over an entire grid. If the guarantees are soft or non-existent, then it is foolish to build protection and control applications that depend on the data delivery. The guarantees must thus be deterministic: met unless the system's design criteria have been violated (e.g., traffic amount, number of failures, and severity of cyber-attack).

Requirement 2. WAMS-DD *must* have a *long-lifetime* and thus be designed with future-proofing in mind. This is crucial in order to have its costs amortized over many projects, utilities, grids, etc. The goal of NASPInet, for example, is to last at least 30 years.

Requirement 3. *Multicast* (one-to-many) is the normal mode of communications, not point-to-point. Increasingly, a given sensor value is needed by multiple power applications.

Requirement 4. End-to-end guarantees must be provided for a *wide range* of QoS+. Data delivery for the grid is not "one size fits all" [Ele04], as shown in Section 3 and Table 2. For example, to provide very low latencies, very high rates, and very high criticality/availability to all applications would be prohibitively expensive. Fortunately, many applications do not require these stringent guarantees, but their less stringent requirements must nevertheless be met.

Examples of the wide ranges that must be provided follow (summarized partly from Table 1 and Table 2):

A. **Latency & Rate**: ten milliseconds or less up to seconds (or hours or days for bulk transfer traffic); .001 Hz to 720 Hz or more

B. **Criticality/Availability**: IntelliGrid [Ele04] recommends five levels of availability of data, from Ultra to Medium.

C. **Cyber-security**: Support a range of tradeoffs of encryption strength compared to delay induced and resources consumed.

Requirement 5. Some merging and future SIPS, transient stability, and control applications require *ultra-low latencies*, delivered (one-way) on the order of a half or full power cycle (8-16 msec in the US) over hundreds of miles and possibly across most of a grid [HNM+08,BBH+10]. Thus, any forwarding protocols should not add more than a millisecond or two of latency (through all forwarding hops) on top of the speed of light in the underlying communication medium, which is roughly 100 miles/msec.

These latencies must be provided in a way that:

A. is ***predictable***, and ***guaranteed for each update message***, not a (much weaker) aggregate guarantee over longer periods of time, applications, and locations such as provided by multiprotocol label switching (MPLS) technology [RFC-3031]. Each sensor update needs to arrive within its required guaranteed deadline. As we will see below, virtually all technologies widely deployed in today's best effort internet do not provide such per-packet guarantees. (ATM is a notable exception, but it does not provide multicast, and it is not a realistic end-to-end solution for the entire grid for other reasons.)

B. ***tolerates*** (non-malicious) ***failures*** in the WAMS-DD infrastructure. No system can tolerate unlimited kinds and numbers of failures. However, much like the power grid must continue in the face of one or more knowable contingencies, the IT infrastructure on which it increasingly depends must still provide these hard, end-to-end guarantees in the face of failures (up to design limits).

C. ***tolerates*** (malicious) ***cyber-attacks***. Power grids are known to be subjects of extensive study and probing by multiple organizations that have significant information warfare capabilities, including nation states, terrorist organizations, and organized crime. WAMS-DD must adapt and continue to deliver data despite cyber-attacks of a designed severity (a bar which should be designed to be increasable over the life of the system). Note that a bug in hardware or software that generations spurious traffic can have an effect similar to that of a cyber-attack.

Requirement 6. Extremely ***high throughput*** is required. Today's synchrophasor applications are generally limited to 30 or 60 Hz in the USA, largely because the communications systems they use are not designed to support higher rates. To not provide much higher sustainable throughput would greatly limit the number of new applications that can help the grid's stability. Indeed, not just synchrophasors but digital fault recorders (DFR) and IEDs in substations provide a wealth of data which is not tapped today. It is quite conceivable, and arguably likely, that "If you build it, they will come" and there will be many thousands of synchrophasors, DFRs, and other sources of sensor updates across a grid. Indeed, DFRs output (today only to disk) at 720 Hz and typically sample at 8 Khz, but their output is presently not used remotely due to communications limitations. If key DFR data could be delivered from a set of DFRs across a grid at 720 Hz many new opportunities open up for transient protection without using expensive dedicated networks, or for "drilling down" into the root causes of an ongoing power contingency using additional contingency-specific data, as described in Section 3.8.

These six requirements must all be met if power engineers are to justify depending on data delivery for their applications. It is crucial that they be able to do so, however, given that the it is almost universally accepted that the grid is inherently getting less stable each year —for example, due to renewable energy sources (which have very different power characteristics) and load outstripping transmission construction—and there are emerging applications which can help mitigate this. And, as quoted in Section 1, [HPR10] notes (**emphasis** is ours),

> Advanced control (and protection) methods will include differential line relaying, adaptive settings, and various system integrity protection schemes that **rely** on low-latency communications.

Providing the above capabilities in WAMS-DD can enable a much wider array of power applications to be deployed, with less time and cost, than if the electricity sector continues with "business as usual" on the data delivery front, or applies technology that will not meet these requirements.

We are not aware of any commercial or military market for a wide area data delivery infrastructure that either has such stringent requirements or exploits aspects of an WAMS-DD including ability to enforce complete perimeter control, ability to know the vast majority of the traffic ahead of time, much fewer kinds of traffic, and other factors incorporated into the implementation guidelines described next in this paper. The reason is quite simple: there is no other market for such stringent requirements, and so, predictably, vendors have not over-designed their products to meet these difficult requirements. In our opinion, however, these requirements are quite achievable, based on state-of-the-art in distributed real-time embedded (DRE) computing—see for example [KKK+01,SS02,CGC+04]—as long as a careful end-to-end analysis is done [SRC84], and the core data delivery mechanisms are not saddled with unnecessary features. Part of this, for example, is not using TCP/IP or web services; while they may have some "reliability", they provide very unpredictable latency [BCH+05,HRW+09]. However, much broader reliability had been explored in the fault-tolerant distributed computing community, from where appropriate lessons, both good and bad, should be heeded [CKV01,DSU03].

5.3 Implementation Guidelines for WAMS-DD

The requirements outlined in the previous subsection were kept to a bare minimum. In order to achieve them, however, we believe it will be necessary to utilize a number of *implementation guidelines* (IG), many which are quite different from what is provided in today's best-effort Internet and what has been the conventional wisdom in networking and distributed computing research. In this section we enumerate and explain these IGs

Some of the IGs below (e.g., **IG4** and **IG5**) are actually deemed requirements for NASPInet [Nor08], but we describe them here as IGs because it is possible to build WAMS-DD without them (though we believe that would be inadvisable for an inter-utility backbone such as the proposed NASPInet). These IGs are drawn from a number or sources, including our knowledge of what the state of the art in distributed computing has demonstrated is feasible, best practices in other industries, and decades of experience gained in DARPA wide-area application and middleware projects of ours and others [STB86,ZBS97,LBS+98,PLS+00,KKK+01,SS02,SLA+02CGC+04].

We note that these guidelines refer to best practices of how to ***build*** a WAMS-DD. Other guidelines (beyond the scope of this paper) will apply on how to *use* one and will need to be developed as best practices. For example, in our experience, many power engineers assume that with synchrophasors, they should always have the phasor data concentrators (PDCs) inside their utility. However, this is a very bad idea for updates that need to be delivered with ultra-low latencies (**DR5**). A PDC aggregates many PMU signals, does error correction and angle computation, then outputs the collection of this information for a given PMU time slot (this is called *time alignment*). Such a PDC may have dozens of PMU signals coming into it, so doing time alignment means that the output has to wait until the slowest PMU update arrives. In this case, the updated sensor values will have suffered significant delays even before they leave the utility to be transported by a wide-area WAMS-DD such as NASPInet. Thus, for those updates that require ultra-low delivery latency, any PDC or other time alignment should be placed as close to the subscribers as possible (ideally in their local area network), even at the cost of a small amount of either wasted bandwidth and duplication of PDCs. Similarly, data that is required with extremely low latency should not have a database in its path: it can be entered into a database after it is sent out, but the database must not slow down the fast delivery path.

We also note that the scope of these IGs involves only the data delivery system for WAMS-DD. It does not include the supporting services that will be required for configuration, security, path allocation, resource management, etc. It will be important for the WAMS-DD that the use of these tools avoids hard-coding these choices, but rather allows them to be specified in a high-level policy language (or at least a database) [SLA+02,PLS+00,Bak09]. For an example of a hierarchical version of such services (a "management plane"), see [GDH+03,BHK+10].

Table 3 provides an overview of the IGs and the DRs which require the given IG. We now explain each of the IGs in turn.

Guideline 1. *Avoid post-error recovery mechanisms*. Traditional protocols for the internet in general and reliable multicast protocols from the fault-tolerant computing research community use post-error recovery [DSU03]. In these protocols the receiver either sends a positive acknowledgement (ACK) when it receives a message, or it sends a negative acknowledgment (NACK) when it concludes that the message will not arrive. However, this can add considerable latency when a message[5] gets dropped: three one-way latencies are required plus a relatively large timeout. This violates **DR1**, **DR5A**, and **DR5B**.

The better alternative is to send sensor updates (messages) proactively over multiple disjoint paths, each of which meets the latency and rate requirements [TBV+05,GBH+09]. Indeed, if multiple independent messages, each going over a QoS-managed path, cannot meet the delivery deadline, then sending ACKs or NACKs is very unlikely to help, and indeed will only make things worse.

Note that the guideline to avoid post-error correction is only for data that has guarantees on a per-message basis. Bulk data transfer is similar to a remote file transfer and will almost certainly employ post-error correction. However, those mechanisms must be different from the ones that have to provide per-message guarantees, and isolated from the hard real-time mechanisms.

Guideline 2. *Optimize for rate-based sensors.* WAMS-DD can be made with higher throughput and robustness if not over-engineered. General-purpose publish-subscribe systems offer a wide range of traffic types, because they are designed to support a wide range of applications [EFG+03]. However, in an WAMS-DD, the vast majority of the traffic will be rate-based.

Guideline 3. *Provide per-subscriber QoS+*. It is crucial that different subscribers to the same sensor variable be able to have different guarantees in terms of latency, rate, and criticality/availability. If not, then a lot of bandwidth will be wasted: all subscribers will have to be delivered that sensor's updates at the most stringent QoS+ that any of its subscribers requires.

Guideline 4. *Provide efficient multicast.* In order to achieve the highest throughput possible, it is imperative to avoid unnecessary network traffic. Thus, never send an update over a link more than once. Also, as a sensor update is being forwarded through the network, if it is not needed downstream in the multicast tree (e.g., those subscribers require it at a lower rate than other subscribers), the update message should be dropped. This can be implemented using a *rate down-sampling* mechanism as is done in GridStat [TBV+05,GBH+09].

These first four guidelines add up to a need for multicast routing heuristics that provide multiple, disjoint paths to each subscriber with each path meeting the subscriber's latency requirement. A family of heuristics developed for this multi-cast routing problem [IH05, Ira06] confirms the feasibility of the approach at the anticipated scale (see **IG10**) if routing decisions are made statically (**IG11**). GridStat's route selection mechanisms are based on these four guidelines, as described in Section 8.2.

[5] We use the term "message" rather than "packet," because in many cases we are describing middleware-layer mechanisms above the network and transport layers.

Table 3: Implementation Guidelines and the Delivery Requirements that Mandate Them

DR1: Hard E2E WAN guarantees	DR2: Future-Proofing	DR3: Multicast	DR4: Wide Range of QoS+…	4A: Latency & Rate	4B: Criticality/Afailability	4C: Cyber-Security	DR5: Ultru-Low Latencies…	5A: Per-update & predictable	5B: Tolerating failures	5C: Tolerating Cyber-Attacks	DR6: High Throughput	IG*x* Prerequisites	Summary of Implementation Guideline IG*x*
X								X	X				**IG1: Avoid post-error recovery mechanisms**
X				X				X	X	X	X		**IG2: Optimize for Rate-Based Sensors**
		X									X		**IG3: Provide Per-Subscriber QoS+**
		X									X		**IG4: Provide efficient multicast**
												2,3	**IG5: Provide Synchronized Rate Down-Sampling**
X					X			X			X		**IG6: Don't depend on priority-based "guarantees"**
X	X			X	X	X							**IG7: Provide end-to-end interoperability across different/new IT**
X								X			X		**IG8: Exploit *a priori* knowledge of traffic**
X								X	X	X		8	**IG9: Have systematic, quick internal instrumentation**
X								X					**IG10: Exploit smaller scale of the WAMS-DD**
X								X				8-10	**IG11: Use static, not dynamic, routing**
X								X	X	X			**IG12: Enforce complete perimeter control**
X								X	X	X	X	12	**IG13: Reject unauth. messages quickly & locally**
											X	2,8	**IG14: Provide only simple subscription criteria**
								X			X	2	**IG15: Support transient, not persistent, delivery**
								X			X		**IG16: Don't over-design consistency & (re)ordering**
											X	2,8,14-16	**IG17: Minimize forwarding-time logic**
X	X			X	X	X							**IG18: Support multiple QoS+ mechanisms for different operating**
								X			X	17	**IG19: Inspect only message header, not payload**
X								X			X		**IG20: Manage aperiodic traffic**

Guideline 5. *Provide synchronized rate down-sampling*. In providing rate down-sampling, it is import to not down-sample in a way that destroys the usefulness of some data. For example, synchrophasors are used to take a direct state measurement at a given microsecond. If some subscribers require only a small fraction of the updates for a set of synchrophasor sensors, it is important that the updates that reach the subscriber at each interval carry the same timestamp. For example, if a subscriber only requires a tenth of the updates from two different variables, then it would not be useable to get updates {#1, #11, #21, …} from one synchrophasor and updates {#2, #12, #22} from another synchrophasor, because the given measurements (e.g., #1 vs #2) do correspond to the same time (they are not the same snapshot), which is the main point of synchrophasors.

Guideline 6. *Don't depend on priority-based "guarantees".* Publish-subscribe delivery systems typically offer a way to specify a priority, so if the traffic gets too heavy less important traffic can be dropped. However, this does not provide a hard end-to-end guarantee to subscribing applications, and even applications that are not of the highest criticality still need their DRs to be met. Instead of priorities, mechanisms must be used that exploit the characteristics of WAMS-DD (as outlined in these guidelines) to provide each subscriber firm assurances that its guarantees will be met so long as the design criteria in terms of kind and numbers of failures (**DR5B**) and cyber-attacks (**DR5C**) is not violated.

Guideline 7. *Provide end-to-end interoperability across different/new IT technologies (providing multicast, latency, rate, etc).* A grid-wide WAMS-DD will *ipso facto* have to span many utility and network organizations. It is unlikely that the same mechanisms will be present across all these organizations. And, even if they are today, if the WAMS-DD gets locked into the lower-level APIs and semantics of a given multicast or QoS mechanism, it will be difficult to "ride the technology curve" and utilize newer and better mechanisms that will inevitably become available over the long lifetime of the WAMS-DD. This is a stated goal of the GridWise community, for example [Gr06]. Fortunately, it is possible to use middleware to span these different underlying technologies in order to provide guarantees that the diversity of the underlying networks that must be spanned; indeed, this is one of the main reasons for the development of middleware over the last three decades.

Guideline 8. *Exploit a priori knowledge of predictable traffic.* Internet routers cannot in general make assumptions or optimizations based on the characteristics of the traffic that they will be subjected to, because they are intended to be general-purpose and support a wide range of traffic types. WAMS-DD, however, has traffic that is not just rate-based, but is almost all known months ahead of time (e.g., when an engineering survey is made of a new power application). This common case can be optimized, as described in later IGs below.

Guideline 9. *Have systematic, quick internal instrumentation.* In order to provide end-to-end guarantees across a wide area despite failures and cyber-attacks, **IG8** must be exploited to provide systematic and fast instrumentation of the WAMS-DD. This allows much quicker adaptations to anomalous traffic, whether accidental or malicious in origin. Finally, this instrumentation should exploit the pervasive presence of GPS clocks in substations and in likely sites for WAMS-DD backbone mechanisms.

Guideline 10. *Exploit smaller scale of the WAMS-DD.* This is a crucial if the challenging delivery requirements are to be met over a wide area with reasonable cost. However, this requires rethinking the conventional wisdom in networking research and commercial middleware products.

NnDB will be orders of magnitude smaller in scale than the Internet at large[6], so it is feasible for the entire configuration to be stored in one location for the purposes of (mostly offline) route selection. Additionally, academic computer science researchers historically consider something that is $O(N^2)$ for path calculation with N routers or forwarding engines to be infeasible; see for example [BHK+10]. However, this assumption ignores two key factors for NnDB. First, N is not in the neighborhood of 10^8 as in the Internet, but rather is more likely $\sim 10^3$ at least for the next 5-10 years; even $O(N^2)$ algorithms are feasible at this scale. Second, as a rule, power engineers do not decide that they need a given sensor's values seconds before they really need it, due in part to the fact that today's data delivery infrastructure requires them to recode hard-coded socket programs and then recompile. Rather, power engineers plan

[6] For example, in the entire USA there are approx 3500 companies that participate in the grid [Blackout2003]. We thus believe that the number of router-like forwarding engines that would be required for a NnDB backbone (at least in the case of broker-based publish-subscribe; defined later) is at most 10^4 and likely only around 10^3.

their power contingencies (and what data they will need in them) months ahead of time with detailed engineering studies, and similarly for their monitoring, protection, control, and visualization needs. Thus, the routing/forwarding decisions involved in path selection can be done offline well ahead of time, while still allowing for handling a modefst number of subscription requests at runtime.

It is also feasible for router-like forwarding engines to store state for each flow. Having a router keep per-flow state has long been considered a bane to networking researchers, because it is considered to be prohibitively unscalable. However, with the much smaller scale, and the much more limited type of applications for a WAMS-DD, storing per-flow state is not only feasible but it is a requirement for providing IG3 (per-subscriber QoS+) with IG4 (efficient multicast); this is something that our GridStat project has been advocating for many years [GDH+03]. However, recently networking researchers are realizing the necessity of storing per-flow state to provide any reasonable kind of QoS [Rob09]. Other recent efforts with roughly similar approaches include as CHART [BMR+09] and PHAROS [RHJ+09].

Guideline 11. *Use static, not dynamic routing and naming.* Much stronger latency guarantees can be provided when using complete knowledge of topology (**IG10**) coupled with static routing. Complete topology knowledge is a reasonable assumption in an NnDB, given that it will be a carefully managed critical infrastructure with complete admission control. Also, almost all of the sensors and power applications will be known well ahead of time, so optimizations for static (or slowly-changing) naming can potentially be useful and can be done while still providing more flexible and dynamic discovery services at a much lower volume. We note that networking and security researchers generally assume that the membership of multicast groups (or a set of subscribers) may change rapidly; see for example [BHK+10]. However, as noted above, that is not the case with WAMS-DD.

Guideline 12. *Enforce complete perimeter control*. All traffic put onto an WAMS-DD must pass admission control criteria (permissions based on rules for both cyber-security and resource management) via a management system: the publisher registering a sensor variable (at a given rate) and the subscribers asking for a subscription with a given rate and end-to-end latency. This is essential to provide guarantees at a per-message granularity. It also enables quicker adaptations.

Guideline 13. *Reject unauthorized messages quickly and locally*. Messages that have gone around the admission control perimeter should be rejected as soon as possible, ideally at the next WAMS-DD forwarding engine, rather than going most or all the way across the WAMS-DD consuming resources along the way. Detection of such unauthorized packets is an indicator of anomalous traffic and hyence evidence of a failure or cyber-attack that needs to be reported to the management infrastructure. When sufficient evidence over sufficient time is collected, an appropriate adaptation can occur.

Guideline 14. *Provide only simple subscription criteria.* This is exactly the opposite of what is usually done with general purpose publish-subscribe in either academic research or commercial products: both tend to favor complex subscription criteria which are expensive to evaluate as each update is forwarded through the system (think of complex “topics”) [EFG+03]. For example, in GridStat, the subscription criteria are latency, rate, and number of paths, and, as noted below, the forwarding decision is done completely based on rate, with static routing. Note also that the lower-level ID of a sensor variable could still be looked up through a complicated discovery service; this guideline is concerned with avoiding complex forwarding logic.

Guideline 15. *Support only transient delivery, not persistent delivery*. Most publish-subscribe systems offer persistent delivery, whereby if an event cannot be immediately forwarded it is stored for some time and then the delivery retried. This harms throughput, however, as well as potentially the per-packet predictability (because it requires storing the data). In our experience it is completely unnecessary for real-time visualization, control and protection, due to the temporal redundancy inherent in rate-based update streams: the next update will be arriving very soon anyway, so the usefulness of a given update decays very quickly. Thus, it is inadvisable to complicate delivery mechanisms to support persistent delivery (though it can be provided "on the side" by other mechanisms). Furthermore, in the power grid, historian databases are already required for archiving data, so there is no reason to complicate the design or otherwise bog down the fastest and highest availability mechanisms of WAMS-DD to delivery historical data[7].

Guideline 16. *Don't over-design for consistency and (re)ordering*. Research in fault-tolerant multicast tends to provide different levels of ordering between updates from the same publisher, or between different clients of the same server, as well as consistency levels between different replicas or caches of a server [CKV01,DSU03][8]. There is no need for anything like this in an WAMS-DD: present data delivery software provides no kind of consistency (or any other advanced properties) at all, so, in our experience, power applications assume nothing in terms of consistency and ordering. The only requirement for such consistency that we have found is reflected in **IG5** for synchrophasors, and the only ordering of any kind is where a PDC combines updates from different PMUs into one message to pass onwards. With devices such as synchrophasors that have accurate GPS clocks the order of events can be directly known and no delivery ordering mechanism is required other than that which is done by a PDC.

Guideline 17. *Minimize forwarding-time logic*. In order to provide the highest throughput, the forwarding logic that decides how a packet or update is to be forwarded on should be kept as simple as possible. On the GridStat project, forwarding decisions are made based solely on the subscription rate of subscribers downstream in the multicast tree [GDH+03,GBH+09]. Given that the traffic is rate-based (**IG2**) and known ahead of time (**IG8**), and that subscription criteria are kept simple (**IG14**), and only transient delivery is supported (**IG15**), and that there are no consistency semantics (**IG 16**), much logic can be pushed off to subscription setup time or even offline. This reduces the logic necessary when an update arrives at a forwarding engine (or P2P middleware mechanisms at an edge) and hence greatly increases throughput and decreases latency.

Guideline 18. *Support multiple QoS+ mechanisms for different runtime conditions*. A given mechanism that provides guarantees of latency and security, for example, will not be appropriate for all the runtime operating conditions in which a long-lived WAMS-DD may have to operate. This is because different implementations of a given QoS+ mechanism can require very different amounts of lower-level resources such as CPU, memory, and bandwidth [ZBS97]. This will be particularly important as WAMS-DD deployments span areas that cannot be controlled nearly as closely as the core backbone can; for example as described in Section 7.2.

Guideline 19. *Inspect only packet header, not payload*. In order to provide the highest throughput and lowest latency, ensure that subscription criteria and consistency semantics allow a forwarding decision to be based solely on a packet header. This is not possible for publish-subscribe middleware that has complicated subscription topics as is typical with commercial and research systems. For them, data fields in the payload also have to be inspected.

[7] We note that such post-event historical data can be delivered by the same network links as the fast traffic with traffic isolation mechanisms; indeed, this is one of the main traffic categories for the emerging NASPInet.

[8] We note that simpler systems, or many other things obeying the "KISS Principle", are not very publishable!

Guideline 20. *Manage aperiodic traffic*. Any traffic that is aperiodic (i.e., not based on rate but on a condition) must be isolated from rate-based periodic traffic and managed accordingly. This can be done deterministically, for example with (OSI Layer 1) optical wave division multiplexing (OWDM) hardware. Further, aperiodic traffic should be aggregated intelligently—ideally based on updateable policies rather than hardcoded settings—instead of sending all alarms/alerts to the next level up for processing.

Regarding the IGs above, and data delivery properties in general, it is important to recognize that you can't have the highest level of all the properties described in the Design Requirements for every sensor variable. As noted in [BST09]:

1. Different properties inherently must be traded off against others.
2. Different mechanisms for a given property are appropriate for only some of the runtime operating conditions that an application may encounter (especially a long-lived one).
3. Different mechanisms for the same non-functional property can have different tradeoffs of lower-level resources (CPU, bandwidth, storage)
4. Mechanisms most often can't be combined in arbitrary ways

Even if you somehow could have them all at once, it would be prohibitively expensive. Given these realities, and the fact application programmers rarely can be expert in dealing with the above issues, middleware with QoS+ properties supported in a comprehensive and coherent way is a way to package up the handling of these issues and allow reuse across application families, organizations, and even industries.

Similarly, it is important to note that meeting **IG3** (and others) requires the data delivery system to be provided at the middleware layer. This is because network-level mechanisms know about packets and IP addresses, not middleware-layer sensor variables and the power applications that subscribe to their updates. There is thus no way that network-level mechanisms can provide different subscribers to the same sensor variable with different QoS+ guarantees, which is mandated by efficient multicast (**IG4**).

Finally, because of length constraints it is not possible in this paper to fully discuss the cyber-security issues that arise in a WAMS-DD; however, they are overviewed in Section 6. Clearly, a WAMS-DD providing universal connectivity creates cyber-security challenges beyond those arising in a conventional, single-utility SCADA system. Cyber-security also interacts with DRs and IGs: for example, techniques used for message confidentiality and authentication must not impose too much additional latency, yet the multicast requirement appears to limit use of symmetric-key cryptography for authentication. But, finally, we note that of the traditional "CIA" cyber-security properties (confidentiality, integrity, and availability), many power practitioners consider availability to be the most important for WAMS-DD (see, for example, [Nor09]). Availability of sensor updates is of course discussed throughout this paper.

Table 4: Coverage of Delivery Requirements and Implementation Guidelines by Existing Technologies

IP	TCP, UDP, SCTP	IPv6 Flow Labels	IP Multicast	MPLS	VLANs & VPNs	PGM & RDM	Spread	ATM	SOSCOE	BB COTS Pub-sub	P2P COTS Pub-Sub	Streaming SQL/CEP	Military RT Apps	SOA/web services	IEC 61850, OPC UA	DNP3, MMS	GridStat	Delivery Requirement or Design Guideline
—	—	?	—	—	—	—	—	Y	D	D	D	—	D	—	—	—	Y	DR1: Hard E2E WAN guarantees
—	—	S	—	—	—		S	—	L	Y	Y	D	Y	Y	—	—	Y	DR2: Future-Proofing
—	—	?	Y	Y	?		Y	—	Y	Y	Y	D	Y	S	—	—	Y	DR3: Multicast
																		DR4: Wide Range of QoS+:
—	—	—	—	Y	?	S	φ	Y	?	Y	Y	—	Y	—	—	—	Y	4A: Latency & Rate
—	—	—	—	—	—	—	Y	—	L	Y	Y	—	Y	—	—	—	Y	4B: Criticality/Availability
S	S	—	S	S	?	—	Y	—	L	Y	Y	—	Y	S	—	—	S	4C: Cyber-Security
																		DR5: Ultra-Low Latencies:
—	—	—	—	—	—	—	—	Y	D	D	D	—	D	—	—	—	Y	5A: Per-message & predictable
φ	φ	φ	φ	φ	φ	—	φ	?	D	D	—	—	—	S	—	—	Y	5B: Tolerating failures
φ	φ	φ	φ	φ	φ	—	φ	—	D	?	?	D	S	—	—	—	F	5C: Tolerating Cyber-Attacks
Y	Y	Y	Y	Y	?	Y	Y	Y	D	Y	Y	Y	Y	—	—	—	Y	DR6: High Throughput
Y	—	φ	—	—	—	—	—	Y	D	?	—	D	?	—	—	—	Y	IG1: Avoid post-error recovery mechanisms
—	—	—	—	—	—	—	—	—	D	?	?	D	?	—	—	—	Y	IG2: Optimize for Rate-Based Sensors
φ	φ	—	—	D	—	—	—	—	?	?	?	?	?	D	—	—	Y	IG3: Provide Per-Subscriber QoS+
φ	φ	φ	Y	?	—	S	S	—	D	?	D	D	?	—	—	—	Y	IG4: Provide efficient multicast
—	—	—	—	—	—	—	—	—	D	—	—	D	—	—	—	—	Y	IG5: Provide Synch'd Rate Down-Sampling
φ	φ	—	φ	—	—	Φ	—	—	D	?	?	D	?	—	—	—	Y	IG6: Don't count on priority "guarantees"
—	—	—	—	—	—	S	S	—	L	Y	Y	S	Y	Y	—	—	Y	IG7: Provide E2E interoperability across diff./ new IT technologies (multicast, QoS+)
—	—	?	—	?	—	—	—	—	D	?	?	L	?	—	—	—	Y	IG8: Exploit a priori knowledge of traffic
—	—	φ	—	?	—	—	S	—	?	?	?	D	P	—	—	—	F	IG9: Have systematic, quick internal instrumentation
—	—	—	—	—	—	—	—	—	D	?	?	D	?	—	—	—	Y	IG10: Exploit smaller scale of the WAMS-DD
—	—	—	—	—	—	—	—	—	D	?	?	D	?	—	—	—	Y	IG11: Use static, not dynamic, routing
—	—	—	—		—	S	—	—	D	—	—	D	—	—	—	—	Y	IG12: Enforce complete perimeter control
—	—	?	—		—	φ	—	—	?	?	?	D	?	S	—	—	S	IG13: Reject unauth. packets quickly & locally
—	—	Y	—	—	—	φ	Y	—	D	D	D	—	D	—	—	—	Y	IG14: Provide only simple subscription criteria
Y	—	Y	Y	—	—	Y	—	—	?			D		—	—	—	Y	IG15: Support transient, not persist., delivery
Y	Y	Y	Y	Y	Y	Y	—	Y	?	?	?	D	?	S	—	—	Y	IG16: Don't provide unnecessary consistency
Y	Y	Y	Y	Y	Y	Y	S	Y	D	?	?	—	?	—	—	—	Y	IG17: Minimize forwarding-time logic
—	—	—	—	—	—	—	S	—	L	?	?	—	?	—	—	—	Y	IG18: Support multiple QoS+ mechanisms for different operating conditions
Y	Y	Y	Y	Y	Y	Y	Y	Y	?	—	—	—	—	—	—	—	Y	IG19: Inspect only packet header, not payload
—	—	—	—	—	—	S	—	—	L	D	D	—	D	—	—	—	F	IG20: Manage aperiodic traffic

5.4 Analysis of Existing Technologies for WAMS-DD

We now analyze how existing technologies and standards meet the above DRs and IGs. Table 4 summarizes this coverage, which we explain next. The columns of this table are existing networking and middleware technologies, while the rows are the DRs and IGs outlined previously in this section. The table cells denote how well the given technology meets the given IG or DR. These have the following values: '**Y**': yes; '—': no; '**S**': some; '**L**': likely (but not confirmed or unconfirmable); '**?**': unknown; 'D': doubtful (but not confirmed or unconfirmable); 'F': future plans (architected for this); '**φ** ': Not applicable (and does not provide). We provide details below, but please note two things about Table 4. First, ***we note that the DRs and IGs are covered very poorly by traditional network-level technologies, power protocols, and some commercial middleware***. Second, some of the values are not confirmed, because it is extremely difficult to glean detailed information about whether or not a commercial middleware product provides a given DR or IG (or even details of a particular mechanism or tradeoff), despite any marketing claims. In these cases, if we believe that it does or does not (for example, based on its intended domain or other information), this is indicated by 'L' or 'D', respectively.

5.4.1 Technologies and Standards at the Traditional Network Layers

Traditional network protocols, including the OSI-2 (“Data Link”) layer (e.g., Ethernet), OSI-3 (“network”) layer (e.g., IP) and the OSI-4 (“transport”) layer (e.g., TCP, UDP, SCTP) ***do not provide any kind of end-to-end QoS+ guarantees or multicast*** (see for example [BCH+05,HRW+09]), and can be ***a significant security risk*** [Kro06]. We now examine these protocols, and extensions to them. to see how they meet the requirements and guidelines. We do not consider experimental or emerging network technologies such as CHART [BMR+09], PHAROS [RHJ+09], and Anagram’s Flow routers [Rob09]. Such technologies may someday be helpful in providing QoS guarantees across parts of a WAMS-DD; evaluation of this is future research. However, they are unlikely to cover a significant portion of such infrastructures for a decade or more. Further, often one of their stated goals of such research is to provide a “fair” transport (not discriminate between flows), which may be fine for the general-purpose Internet but is a very bad idea for WAMS-DD.

IPv6 Flow Labels: IPv6 flow labels [RFC-3697] associate each “reservation” with an application-to-application network socket connection, which would contain many different sensor update streams with a wide range of required QoS+. Packets are processed in a flow-specific manner by the nodes that have been set up with flow-specific state. The nature of the specific treatment and the methods for the flow state establishment are out of scope of the specification.

IP Multicast: IP Multicast provides efficient multicast for a single, non-replicated flow. However, if multiple IP multicast groups are used as a replication mechanism there is no guarantee that the corresponding multicast trees will be disjoint, which is important not only for efficient multicast (**IG4**) but also for providing low latencies in the face of failures (**DR5B**). It also does not, by itself, have other end-to-end capabilities that are necessary for WAMS-DD, as shown in Table 4.

MPLS: MPLS is designed to give ISPs a set of management tools for bandwidth provisioning, not to provide fine-grained (per-update) QoS [Far09]. Its guarantees are very weak compared to the needs of a critical infrastructure: it gives aggregate economic guarantees over user, location, and protocol, not hard guarantees (**DR1**) for each update (**DR5A**), for example. Further, different ISPs can implement MPLS in different ways, and there are no facilities for combining flows across different ISPs (as would be required in WAMS-DD) and being able to reason about what the end-to-end predictability of delays will be.

MPLS has some fault tolerance mechanisms (fast re-route feature, detour merging, and end-to-end path protection). However, this provides a minimum latency of about 50 msec, which is far more than what is needed for emerging SIPS and Transient Stability applications described above. We are aware of no MPLS providers that will guarantee anything close to these requirements (**DR5**).

VLANs and VPNs: Virtual local area networks (VLANs) and virtual private networks (VPNs) intrinsically meet none of the DRs listed above as their purposes are orthogonal to the DRs. A VPN or VLAN could be part of WAMS-DD but VPN and VLAN technologies alone do not meet the requirements, and in fact can add greatly to latency and decrease throughput.

Pragmatic General Multicast (PGM) is a transport-layer multicast experimental protocol [Iet01] whose implementation by Microsoft is known as Reliably Delivered Messages (RDM). PGM runs over a datagram multicast protocol such as IP multicast to provide basic reliable delivery by use of negative acknowledgements (NACKs). PGM uses a rate-based transmission strategy to bound the bandwidth consumed. However, it does not provide any real-time guarantees.

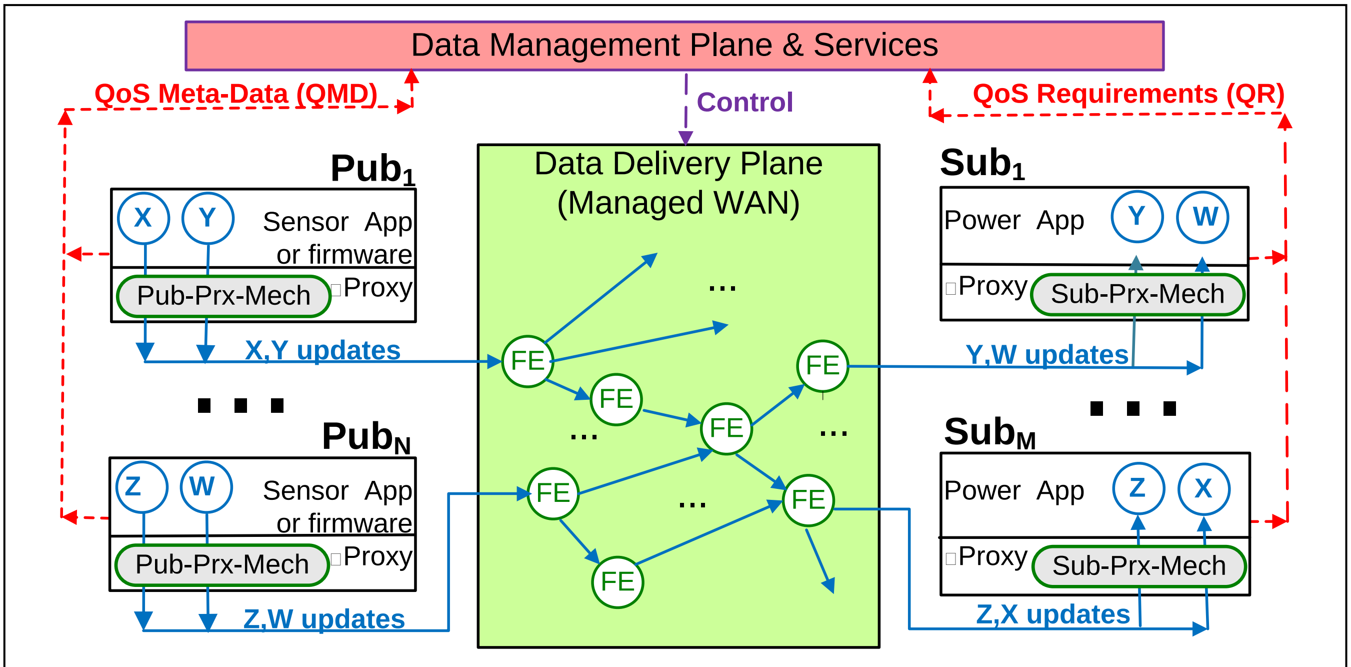


Figure 15: Broker-Based WAMS-DD

Spread [ADS00,Spr10] can be considered as a high-level multicast protocol that provides a range of ordering strengths across a wide-area network. It supports ordered delivery (and resulting consistency) even in the face of network partitions, and is used largely for replicating databases. It has no real-time mechanisms.

ATM: Asynchronous Transfer Mode (ATM) is a networking technology sometimes employed in wide area networks. It offers very strong latency guarantees on a per-message basis. However, it does not support multicast (DR3) and multiple disjoint paths (DR4B), and it does not follow any of the IGs other than IG1. Thus, ATM is not an end-to-end solution for WAMS-DD. However, given its strong latency guarantees at the right granularity, it can be part of a WAMS-DD that overlays ATM and other kinds of lower-level networking technologies.

5.4.2 Commercial Middleware Technologies and Standards

There is a wide range of commercial, off-the-shelf (COTS) middleware frameworks providing different kinds of services with some relevance for WAMS-DD.

We first consider middleware supporting the publish-subscribe paradigm. There are two distinct architectures for publish-subscribe middleware, each with advantages and disadvantages.

5.4.2.1 Publish-Subscribe

We now overview the two main architectures for implementing publish-subscribe systems, broker-based and peer-to-peer. We describe their mechanisms as well as their advantages and disadvantages. In both, the WAMS-DD Cloud described earlier is called a Data Delivery Plane (DDP), though with different typical scopes. Both types share some characteristics: e.g., they can potentially enforce complete perimeter control (**IG12**)

Broker-Based WAMS-DD

Broker-based (BB) pub-sub systems rely upon an infrastructure of broker nodes to forward messages towards subscribers. The Data Delivery Plane is, for WAMS-DD (though not necessarily commercial systems), a managed wide-area network (WAN), e.g., it implements **IG12** (complete perimeter control).

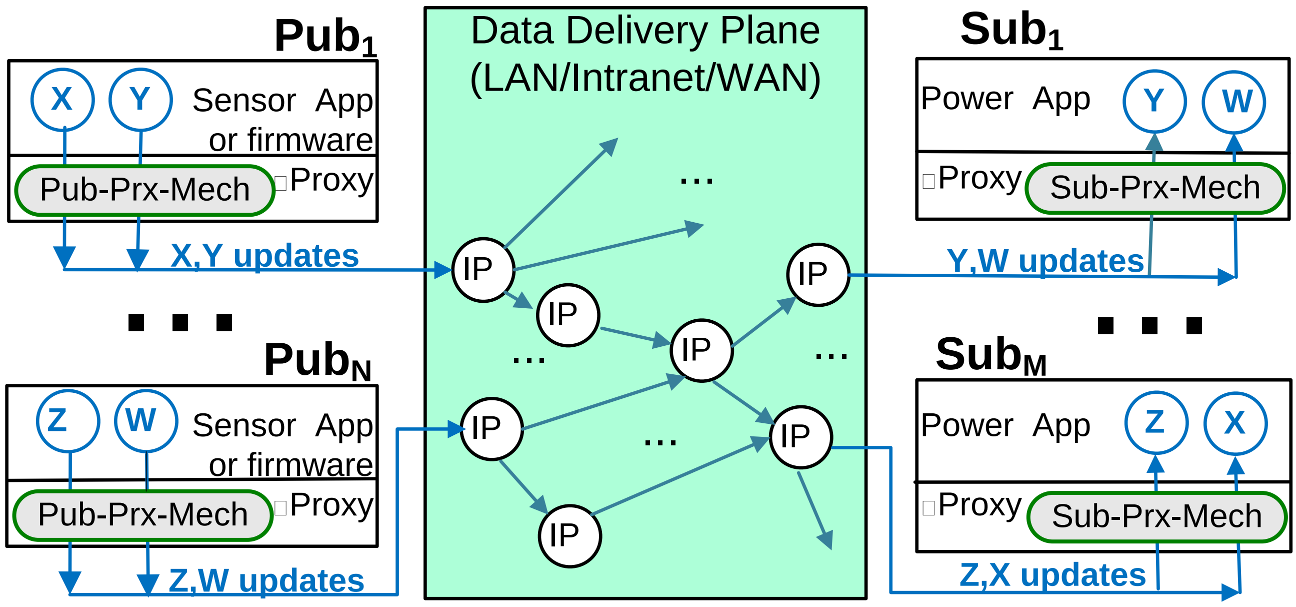


Figure 16: Peer-to-Peer WAMS-DD

BB pub-sub WAMS-DD is depicted in Figure 15. A node in the DDP is called a Forwarding Engines (FE), and is a device specialized for the particular BB pub-sub framework. We depict the mechanisms that a BB WAMS-DD system can exploit in green: these consist of the proxies and the FEs.

In BB WAMS-DD systems intended for mission-critical applications there is often a separate "plane" for managing the data[9] and providing services. This is depicted in red in Figure 15; it is shown here as a single entity but is often distributed. Publishers provide the data management plane with basic QoS Meta-Data (QMD) about their publications; for example, the rate at which they will output updates. Subscribers provide QoS Requirements (QR) including rate, latency, etc. The Data Management Plane then exerts control over the Data Delivery Plane (depicted in purple) in order to provide the delivery guarantees, e.g. by updating a forwarding table for an FE.

BB pub-sub systems require a broker/server infrastructure to be installed: you can't just buy an IP router from Cisco or others. This can be a disadvantage, which often for small and medium scales cannot be amortized over enough applications to be justified. BB pub-sub systems have an advantage, however, in that they place intelligence inside the network, not just at the edges. This enables, for example, efficient multicast (IG4) and rate down-sampling throughout an NGDDS, not just at the edges. It also allows the potential to reject unauthorized packets at their next "hop" through the system (**IG13**).

Additionally, BB can exploit mechanisms in the graph of FEs in order to provide many more of the IGs described in Section 5.3. For example, such an FE can be used to provide per-subscriber QoS+ (**IG3**), provide synchronized rate downsampling (**IG5**), exploit a priori knowledge of traffic (**IG8**), exploit the smaller scale of the WAMS-DD (**IG10**; e.g. it can contain per-subscriber state: a forwarding table entry for every subscription that it forwards updates for).

An example of a BB WAMS-DD is GridStat, described in Section 8.

[9] In telecommunications parlance this is often called the "Control Plane", hence our usage of the term "plane".

Peer-to-Peer WAMS-DD

Peer-to-peer (P2P) publish-subscribe systems place mechanisms for reliability and filtering only at the edges of an infrastructure. A canonical architecture for a P2P pub-sub configuration of WAMS-DD is given in Figure 16. For the Data Delivery Plane, P2P systems typically rely on a combination of IP multicast and Ethernet broadcast to be as efficient as possible. Note that, in this figure, we omit any data management plane (which is often not present as a separate core entity in P2P systems, the edge mechanisms collectively implement it).

One other thing to note in Figure 16 is that the controllable mechanisms for affecting traffic lie at the edges, in the proxies. Certainly P2P WAMS-DD will exploit IP multicast as much as possible, but this has its limits, as described previously in Section 5.4.1. Because its control mechanisms are at the edges, both QMD and QR are communicated to other proxies that collectively provide the delivery guarantees. Similarly, the only WAMS-DD specific mechanisms are in the proxies, so control messages also must go there. (In Figure 16, to help aid understanding, the red and purple traffic is omitted, mainly to keep it readable. In practice, this traffic would be delivered via the Data Delivery Plane, i.e. via IP routers.)

P2P pub-sub systems have an advantage in smaller and medium sized deployments, but for larger scales the lack of mechanisms in the backbone core for rate down-sampling and fault tolerance limit their abilities to achieve extremely low latencies in the presence of failures.

Federated Broker-Based and Peer-to-Peer WAMS-DD

A **federated combination** of P2P and BB publish-subscribe systems has the potential to offer much of the best of both worlds. Here, near the edges (e.g., within a single utility, or sometimes within an ISO), P2P pub-sub is employed. Between utilities (or ISOs), BB pub-sub is used in order to support higher throughputs and the lowest possible latencies over distance. A federated amalgamation of P2P would feature a globally unique namespace for variables and utilities, and could seamlessly pass messages with standardized wire and message formats [BST09].

Such a federation is overviewed in Section 7.2, after NASPInet is introduced.

5.4.2.2 Business-to-Business and Web Services

Another middleware category called *streaming queries* (also known as streaming SQL or complex event processing) consists of a network of computer nodes that manipulate data streams through continuous queries in order to selectively propagate data, merge streams with existing data, or store data in a distributed database. Such systems are not designed to provide hard end-to-end WAN guarantees (DR1) with per-message granularity (DR5A) while tolerating failures (DR5B). Given their intended application domain—often finanancial markets—they also do not follow most of the IGs.

More recently a number of vendors are offering middleware based on web technologies such as HTTP, XML, and “web services” for use in the power grid. We note that scalability and throughput of such systems is highly questionable due to the many integration layers they typically add to make it possible to glue together just about any application to another [Bir04]. Ken Birman, a leading expert in reliable distributed computing (and founder of multiple companies) notes in [Bir06] (***emphasis*** is ours):

> *It doesn’t take an oracle to see that the mania for web services, combined with such rampant online threats, contains the seeds of a future debacle. We’re poised to put air-traffic control, banking, military command and control, electronic medical records, and other vital systems into the hands of a profoundly insecure, untrustworthy platform* **cobbled together** *from complex legacy software components.*

Unfortunately, one can add the smart grid to this list: a number of utilities and organizations see web services as a key enabling technology for the smart grid (for example [PJM07,IEEE-1547.3,Kle09,Ope10]).

5.4.2.3 Other Miscellaneous Middleware

There are a number of other middleware systems that have been developed and deployed (though, as noted earlier, very few in the power sector [BST09]), and a good number have been commercialized for decades. It is thus impossible to cover this space comprehensively (or close!), other than to repeat our observation that, while there may be superficial similarities between such products and WAMS-DD, the requirements for ultra-low latencies, wide geographic scope, very high throughput, and other factors make WAMS-DD fairly unique. This is not to say that existing middleware product families cannot be extended to meet these requirements. Rather, to the authors, it seems unlikely that such products "out of the box" would meet them, given that there is no market for them as of yet.

One middleware system to note is System of Systems Common Operating Environment [Sys10]. SOSCOE is being used in a "smart grid" project involving Consolidated Edison, PJM, and other partners [And09].

SOSCOE is a sophisticated, multi-tiered middleware framework designed to enable "interoperability between and within common military and commercial command and control systems" [Boe10]. As such, it is a sophisticated integration framework ("glue" middleware) meant to integrate many different lower-level middleware systems. It also is designed for military environments where computing and network resources can vary widely (e.g., see [ZBS97]). Such glue middleware can, in our opinion, have many uses in integrating disparate systems. However, given its many layers and its intended environment, we believe it is highly unlikely to be able to support ultra-low latencies (**DR5**), extremely high throughput (**DR6**), hard guarantees over a wide area (**DR1**), etc.

5.4.3 Existing Power Technologies and Standards

Existing communication protocols in the power grid have, with few exceptions, been designed by engineers unfamiliar with the state of the art and practice (or even the history of) network protocol design, distributed computing systems, and other supporting areas. Further, middleware is rarely used in today's power grid, despite being considered a "best practice" in many other industries for a few decades [BST09]. It is not surprising, then, that there seem to be no networking technologies developed for the power grid that meet any of the DRs above. Part of this is limitation is because commonly-used power technologies are intended for a substation scope, with the only QoS+ "mechanism" being massive over-provisioning of bandwidth. Unfortunately, the reality is that, when moving from a LAN to a WAN environment, there are many issues that arise, and often implicit design decisions, that cannot possibly be solved by merely layering a new "WAN-appropriate" API over existing LAN-based protocols [ZBS97]. We now overview some of the more common power protocols and standards related to communications.

OPC-UA [MLD09] was designed for a substation scope and is fairly crude. It uses TCP, which was not designed for predictable latency and does not support multicast. Subscribers and publishers "ping" each other to verify if the other is up, which not only does not scale but also ignores best practices for publish-subscribe systems. OPC-UA also supports SOAP and HTTP for a transport, but both are worst than TCP in any mission critical setting. Further, OPC was for many years a Microsoft-only software system, which casts doubt on its ability to reliably interoperate with systems based on other operating system families, as well as its overall security and reliability[10].

IEC 61850 was also designed for a substation scope, including having messages mapped directly onto an Ethernet frame. Based on the first two author's extensive experience with real distributed computing systems and middleware, IEC 61850's ability to be reasonably extended beyond a substation LAN is thus highly questionable

[10] The SEL authors of this paper have no public opinion on this subject, this judgment is from the WSU authors.

in terms of data delivery mechanisms[11]. However, its CIM can potentially be of great use in a WAMS-DD, especially when (and if) the harmonization with C37.118 is completed, in particular in helping automate QoS+ settings and perhaps adaptation strategies for a wide variety of sensors and applications that use them. And, if the 61850 GOOSE APIs were successfully extended to the WAN, then 61850 may well be able to successfully use a WAMS-DD transport, if those APIs were freed of LAN assumptions and structuring.

IEEE C37.118 is a standard for synchrophasors that includes standard message formats. Unfortunately, its present version has no separation between these formats and data delivery mechanisms for them. C37.118 is being revised to allow different data delivery mechanisms to be used. If successful, then C37.118 synchrophasor updates should easily be deliverable by any WAMS-DD transport.

MMS also does not have any data delivery mechanisms. It can map onto the OSI protocol stack (which never got adopted in practice) and TCP/IP (which of course has severe limitations for WAMS-DD; see [BCH+05,HRW+09]).

An information architecture for the power grid is proposed in [XMV+02], which contains an analysis of 162 disturbances between 1979 and 1995 and indicates that information systems have an impact on power grid reliability, and points out major deficiencies in the current communications scheme. The paper contains proposals for different ways to structure interactions between control centers and substations, and reliability analyses of different schemes. However, it does not propose any communications mechanisms, and relies on off-the-shelf network technology which are shown above to not meet many of the DRs and IGs.

6. Overview of Cyber-Security Issues Specific to WAMS-DD

This paper so far has concentrated mainly on delivery performance and availability issues. It specifically does not discuss the details of cyber-security issues and tradeoffs for WAMS-DD, a complex subject of its won. In this section, however, we provide an overview cyber-security issues that are different for WAMS-DD than for more general distributed computing applications. For more information see [DoE06,DFH+07,HBD+07,GDG+10].

6.1 Overview of Cyber-Security in the New (Electric Grid) World

Security of SCADA systems today is characterized by efforts to prevent intrusions into control systems and to maintain security on communication links between substations and control centers. So-called “bump-in-the-wire” security devices are being deployed to address the latter issue. The associated technical challenges include limited available bandwidth along with a need to avoid increasing reporting latency. Products such as the SEL-3021 Serial Encrypting Transceiver are available in this space, and research and specifications such as [Aga06,TS08] suggest techniques for minimally increasing latencies and bandwidth requirements.

For the applications described above and the communication system that supports them, the security issues are considerably more complicated, although their solution is also less constrained by the communication system's capabilities: a high-bandwidth, low-latency communication service is already assumed. In moving away from point-to-point and star topologies, however, the need for end-to-end security across multiple network hops arises. Particularly, the system must ensure that each application using data from a particular sensor is guaranteed that the data actually was produced by that sensor. This is an integrity and authenticity requirement.

[11] The SEL authors of this paper have no public opinion on this subject, this judgment is from the WSU authors.

The communication service must also ensure that data is timely delivered so that control decisions in turn may be made timely. Communication service capabilities for delivery over multiple paths, admission control, and ability to quickly reroute around down links and edges address this part of the issue. Overall system design and control system design must also contemplate the question of what will happen if expected data is not available and provide safe, if less optimal, operations in this case.

The increased complexity of security in the new system arises from (three) primary sources: abandoning the star topology in favor of an any-to-any multi-hop approach; inter-organizational communication paths; and use of multi-cast to achieve efficient use of communication resources when a single data stream is needed by multiple applications at different locations.

6.2 End-to-end message integrity and authenticity

A standard cryptographic technique for message integrity and authenticity uses a hashed message authentication code (HMAC) along with a key known only to the sender and receiver to ensure that messages are received as sent and that they really come from the purported originator. The originator sends *(M, H(M,K))*. Based on the received *M*, the receiver independently computes *H(M,K)* and compares it to the received value. Authentication is also needed in the subscription process to *authorize* the subscriber. This is a decision that may be made by the publisher itself or by a one of its broker ancestors. Regardless of who makes the decision, key management is needed between the authorizer and the subscriber; and if the publisher does not do its own authorization, between the authorizer and the publisher.

At the operational level, whether based on symmetric techniques or public-key techniques, the problem is then how do publishers, and subscribers acquire the keys they need to authenticate one another. And how do they know that the keys are actually those of the entities to whom it is purported that they belong. The classical answer to this latter question is that the knowing relies on the security of the channel over which the keys are learned. In the setting of the power grid, key distribution is simplified by the relatively controlled nature of the environment: new devices and applications are introduced with forethought by designated individuals rather than the self-introduction process that is typical in the public Internet. On the other hand, geographic isolation of many substations means that there is strong incentive for distributing keys to devices only at the time they are commissioned.

As for scale, the number of entities needing authentication keys is likely to be on the order of 10s of thousands for the North American grid – not an insurmountable number.

To reduce the computational cost of signing, and to reduce the exposure of keys due to use for encrypting or signing large amounts of nearly-similar traffic, usual practice involves generating and using a *session key* for use over a limited lifetime between two entities. The entity authentication process is embedded in the session-key generation process. Thereafter, entity authentication is implicit in the message authentications that take place for each received msg.

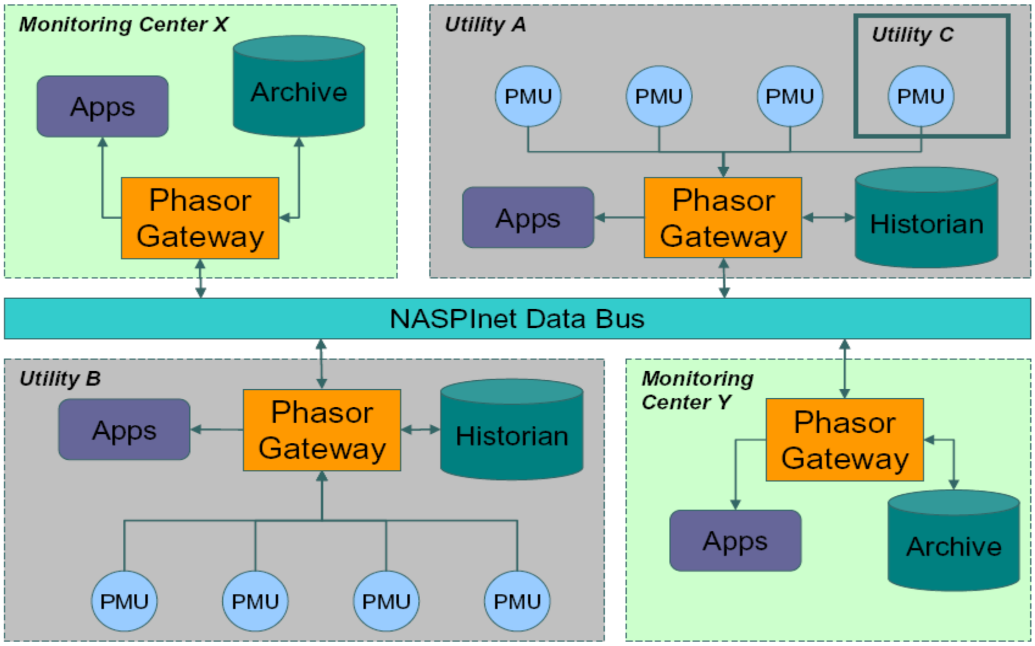


Figure 17: NASPInet Conceptual Architecture

7. Overview of NASPInet

The North American Synchrophasor Initiative (NASPI) is a government-industry consortium that is, in our opinion, the world leader on deploying synchrophasors. (Its predecessor was the Eastern Interconnect Phasor Project (EIPP).) Further, it is the only effort that we are aware of—and we would very likely be so aware had one existed—that is dealing with E2E WAMS-DD issues at a more than superficial level.

The NASPInet architecture envisioned to support the use of such sensors over a wide area is given in Figure 17 [Nor08][12]. It conceptually has two main components, the Phasor Gateway (PGW) and the Data Bus (DB). The PGW has the APIs for integrating data historians and is the edge of NASPInet. The NnPG also serves a useful political purpose (reassurance to utilities): it serves as a demarcation beyond which NASPnet cannot change a utility's communications infrastructure (unless the utility wants this). It also provide, via the APIs, the necessary mechanisms with which to send and receive NASPInet data over the DB, much like a router provides mechanisms for an application to access the general-purpose Internet.

7.1 NASPInet Data Bus

The NASPInet Data Bus concept was fairly undefined in its original form, with less implementation detail than Is provided in the generic WAMS-DD from Figure 14. Its main purpose is to [Nor08]:

- Provide connectivity between Phasor Gateways and other elements of the NASPInet
- Provide Quality of Service (QoS) guarantees for reliable and redundant delivery fo real-time operational data
- Provide QoS conformance monitoring over NAPSInet for Service Classes
- Enforce conformance with cyber security and access control policies.

[12] The diagram in Figure 17 was created in a summer meeting circa 2007 at Pacific Northwest National Lab, but [Nor08] is the first document or even URL we can find of it.

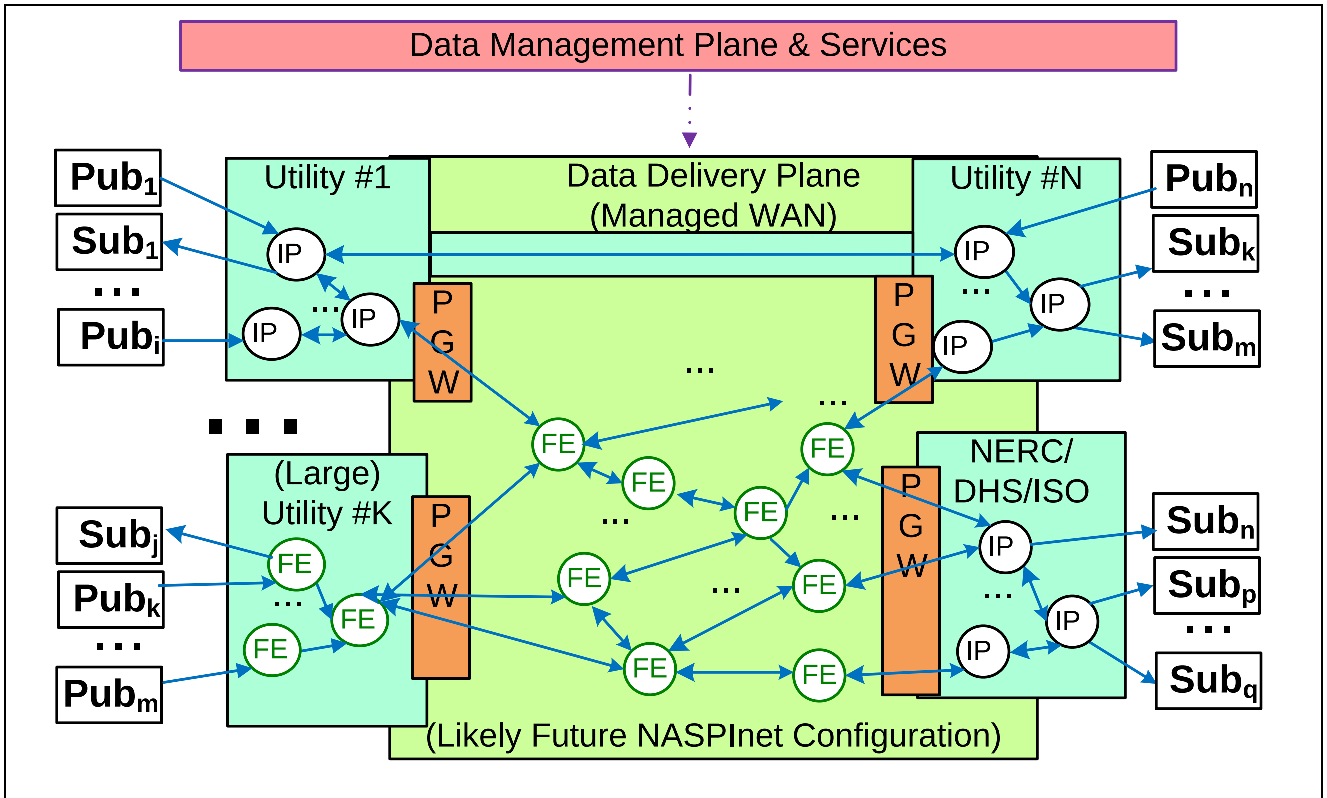


Figure 18: NASPInet Federated Broker-Based and Peer-to-Peer WAMS-DD

The actual implementation constraints and choices were left fairly unspecified in [Nor08] (the creation of which the first author was heavily involved) and its follow document [Hu09]; per these documents, it could conceivably be implemented with IP multicast or publish-subscribe, whether BB or P2P.

7.2 Federated NASPInet Data Bus

We believe that, in practice, NASPInet is going to evolve in a number of ways that are not as simple or "clean" as depicted in figures earlier in this document (or in [Nor08,Hu09] for that matter). One main complication is likely to be is this: even if desired (which we believe it is), it may not be practical to install FEs as shown in Figure 15 everwhere that NASPInet had to deliver data; or, at the least, it may be much more expedient to have portions of inter-utility data delivered by P2P mechanisms, given that they do not have the burden of installing FEs.

We thus think that a more likely "mixed" NASPInet configuration in 4-5 years may look like that in Figure 18. Even more, the P2P mechanisms used by different utilities (or at the edges of NASPInet) may not be the same. For example, one could be from the multi-vendor Data Distribuion Service (DDS) [OMG07], for example from RTI [RTI10a,RTI10b], which is being deployed in Grand Coulee Dam and a dozen other Army Corps dams. Another could be an internal utility data delivery system. Further, there are a number of things to note about Figure 18:

- While almost all inter-utility traffic is carried by Broker-Based FEs, Utility #1 and #N communicate via P2P mechanisms.
- Utility #K, being quite large, has deployed NASPInet-compliant FEs inside its utility, to give it enhanced control over the QoS+ properties. (The utility can still maintain control over its own network, i.e. it does not cede control of its internal FEs to NASPInet or any other entity.)
- Publishers and subscribers of NASPInet are not directly connected to a PGW, but rather the updates traverse through a utility's internal communications infrastructure. We believe that this will come

increasingly common as the logic of power applications becomes less centralized and more distributed, for example with distributed control, microgrids, and other emerging technologies.

This likely federated NASPInet, involving combinations of BB and P2P mechanisms, highlights the importance of carefully investigating how QoS+ and security properties compose across such a federated system. This is work we have initiated between the GridStat project at WSU (described in Section 8) and the multinational INSPIRE project from Europe [AKR+08,AKR+09,Ins10,GKS10,KJD+10,GDG+10].

7.3 NASPInet Service Classes

As a conceptual exercise to help convey the notion that there will be different kinds of traffic with different delivery requirements for NnDB, five initial service classes have been identified [Nor08]:

A. Feedback Control (e.g., small signal stability)

B. Feed-forward Control[13] (e.g., enhancing state estimators with synchrophasors)

C. Post-Event (post-mortem event analysis)

D. Visualization (for operator visibility)

E. Research (testing or R&D)

These classes are further distinguished by varying qualitative requirements for such properties as low latency, availability, accuracy, time alignment, high message rate, and path redundancy. This is depicted in Table 5 (adapted from [Nor08]). While distinguishing the classes in this way is an important first step, it is nowhere what is needed to plan, design, and manage a WAMS-DD, as we discuss next. It also considers the lowest required latency to be 100msec, which as we have explained above in Sections 3.4 and Section 3.6 and is nowhere near low enough for a long-lived WAMS-DD. Further, as described in Section 3 and depicted in Table 4 , much more granularity of specifying traffic attributes is required than given in Table 5.

Table 5: Traffic Attributes of NASPInet Service Classes

Traffic Attribute	CLASS A: Feedback Control	CLASS B: Feed-Fwd Control	CLASS C: Post-Event	CLASS D: Visualization	Class E: Research
Low Latency	4	3	1	2	1
Availability	4	2	3	1	1
Accuracy	4	2	4	1	1
Time Alignment	4	4	1	2	1
High message rate	4	2	4	2	1
Path redundancy	4	4	1	2	1

Table key: 4≡critically important, 3≡important, 2≡somewhat important, 1≡not very important.

[13] We note that this term is not in standard use among power researchers, and may be misleading to some. This is because there is virtually no control being done in grids today using feed-forward techniques, due to well-known stability issues.

7.4 Resource Management of NASPInet Service Classes

A common misconception, in our experience, is that an entity (utility, ISO , RTO, etc) can simply specify that it wants, for example, a "Class A" network, and this is all that is needed. However, this will not necessarily result in a WAMS-DD meeting the requirements: more is needed. For example, if too much traffic of "easier" classes is on the network, then you will not get Class A guarantees (or Class B if that is what was ordered).

Rather, to provide the DRs described in Section 5.2, one needs to do significant *resource management* with NASPInet (or any other WAMS-DD). That is, all traffic must be accounted for when an application asks to subscribe to a new sensor feed with given QoS+ guarantees. This is embodied in a number of IGs, including **IG8** (exploit traffic knowledge), **IG9** (systematic, quick internal instrumentation), **IG12** (complete perimeter control), **IG13** (reject unauthorized packets quickly and locally), and **IG20** (manage aperiodic traffic).

8. GridStat WAMS-DD Middleware Framework

8.1 Overview

GridStat is a WAMS-DD middleware framework being developed at WSU. It has been designed to meet the DRs described in this paper, and has to date implemented almost all of the IGs. We now overview elements of GridStat that are different from commercial and research publish-subscribe systems; these are ways that we have designed it to exploit the nature of a more controllable WAMS-DD in order to achieve its extreme requirements.

8.2 Route Selection

GridStat route selection is done at the time subscriptions are created. This is generally offline, far ahead of time, though GridStat does support runtime subscription creation. When a subscriber asks for a subscription, GridStat calculates the path(s) that will meet the delivery requirements; the application programmer does not have to concern itself with this.

Routes of course need to be chosen to meet real-time requirements for the subscriptions; furthermore, multiple routes using disjoint sets of resources are needed in order to maximize availability of delivered data (i.e., the *disjoint paths problem*). Thus route selection algorithm is part of meeting **IG3** (per-subscriber QoS+).

However, the existence of two disjoint paths each meeting a latency bound is an NP-hard problem even for unicast. But GridStat needs even more than this: route selection heruristics for multi-cast trees, not merely point-to-point paths.

To make best use of available resources and avoid unnecessary *blocking*, routes would need to be selected with global knowledge of all the current subscriptions. This leads to additional NP-hard problems.

Because a number of fundamentally hard (NP-hard) problems in network path selection can be reduced to optimal route selection for GridStat we have developed heuristics for GridStat route selection with emphasis on simplicity and correctness. Whether the heuristics are successful or not in selecting routes to meet the requirements of a particular set of subscriptions will depend on the available resources. Our research suggests that in practical settings if the network is modestly over-provisioned, relative to the absolute minimum requirements needed to satisfy a set of subscriptions, the heuristics will usually succeed in finding routes.

For more on route selection in GridStat, see [IH05,Ira06].

8.3 Forwarding Engine Mechanisms and their Performance

GridStat forwarding engines map incoming messages onto outgoing links based on the publication ID and timestamp contained in the message[GBH+09,Gje06,Hel07]. The routing tables in each forwarding engine are maintained by the management plane interface of the FE in cooperation with the management plane QoS brokers. A message is forwarded on a link if and only if there is a downstream subscriber whose subscription rate indicates that the message is of interest. Thus the forwarding engines implement both a multicast mechanism to meet DR3 and different delivery rates for different subscribers to a particular publication **IG3** [JHG+06].

In order to ensure bounded end-to-end latency the forwarding engines must provide local, per-subscription latency bounds that can be incorporated into the route selection decisions overviewed in Section 8.2. To this end, not only do the FEs choose the outgoing links for each message, they also order the outgoing queues to meet each packet's latency bound. (Note that this is only possible because admission control limits the number of publications using a link to what it can support.)

GridStat forwarding engines are implemented as user-level processes running on a commodity operating system (Linux). The FE implementation in Java achieved about 160 μ s average processing latency on 2008-era desktop hardware under heavy CPU load but worst-case latency was in exceeded 11 msec and best-case was 107 μ s. A C implementation on 2008-era the same hardware delivered worst-case latency of 71 μ s (average 24 μ s) and can scale to approx 20K forwards/sec [Mut08]. Application layer code is not able to manage the order of packets in outgoing queues using standard socket OS interfaces. The problem is that the OS implements the outgoing queues and largely hides them from the application layer. In conjunction with the C FE implementation we also incorporated a new queueing mechanism for Linux sockets [XCN04] and developed additional system calls to communicate ordering information from the application layer code to the queueing. Thus, application-level FE is responsible for determining the order of queued packets and the kernel-level mechanism sends the packets in the proper order. [Mut08]

In addition to the Java- and C-based forwarding engines we have prototyped a FE implementation for the Intel IXP series of network processors [Swe09]. The prototype demonstrated the potential of network processor technology to provide even lower forwarding latency (about 10 μ s on 2003 vintage hardware) and higher throughput due to the network processor's parallelism, and can handle approx. 1 million forwards/sec. However, limitations of the network processor hardware such as lack of division operations mean that while they are potentially interesting in the future they are likely not currently viable as a replacement for general purpose processors in the FE role. We also note that implementing on the network processors is a difficult and time-consuming task and they are quite expensive. Their use would only be justified in high-traffic situations where a general-purpose processor could not meet the performance needs.

Finally, we note that the GridStat mechanisms for forwarding support synchronized rate down-sampling (**IG5**).

8.4 Cyber-Security

As noted earlier, the cyber security requirements for WAMS-DD emphasize integrity and availabiity over confidentiality among the usual cyber-security requirements. For GridStat, we have also considered the effects of the long system lifetime, **DR2**, on the cyber security mechanisms. The GridStat *Security Management Service* (SMS), a part of the management plane, supports stackable and replaceable modules in the publisher output path and the subscriber input path. For the data plane, the module stack is configured by the SMS during publication and subscription set-up, including, if necessary, delivering keys and modules to the publishers and subscribers [Sol07,SHC+09]. Thus, as cryptographic software becomes obsolete new algorithms can be deployed throughout the data plane. The module stack allows, on a per-publication basis, configuration of message encryption, integrity, authentication, and obfuscation functionality.

For the management plane the major issue is authentication of management plane entities to one another. GridStat uses the replaceable modules concept here as well in order to meet the long lifetime requirement. [CHB10,Cha09]

The real-time and scale requirements for WAMS-DD lead to some difficult challenges for cyber security leading to inevitable trade-offs between security requirements and performance requirements. For example, the computational resource requirements of public-key cryptography make its use for verifying message integrity in the data plane at best unpalatable if not infeasible in many use cases. The issue is not merely added latency but also the consumption of computational cycles on limited-capability devices that are needed for other things. Time-based signature schemes such as TESLA [PCS+05] and TVOTS [WKH+09] have lower computational cost but have as large or larger latency impacts. The modular security approach of GridStat can make any of these mechanisms available, but users of WAMS-DD will need guidance about which one to use in different situations. For example; without public-key techniques, non-repudiation is not available for recipients; with time-based techniques sender authentication is available but latency may be high; with a single symmetric-key-based signature a message can be authenticated as having come only from one of the multicast group (sender or recipient) not only the designated publisher; or per-subscriber symmetric-key signatures could be used at the cost of additional computation by the publisher and additional network traffic.

Finally, the scale envisioned for WAMS-DD leads to challenges for authenticating the binding of received data to the proper problem-domain (power grid) data source. Today, utilities are beginning to get a handle on this naming problem for their own equipment and substations but WAMS-DD increases the name binding scope to an entire grid and raises the additional question of how to ensure that the bindings are securely and accurately communicated.

8.5 Advanced Mechanisms

8.5.1 Mode Change Mechanisms for Fast, Systemic Adaptation

It was realized early on in the design of GridStat that, because the route selection algorithms are computationally very complex, that having to handle the addition of a large number of subscriptions during a crisis would actually constitute an effective denial-of-service attack on the subscription service. Fortunately, system operators do not typically choose random sensors to subscribe: they use a checklist planned far in advance for that particular contingency (**IG8**: exploit *a priori* knowledge of traffic).

GridStat thus implements the emergency modes described in Section 3.8 (and mentioned, but not defined in [Nor08,Hu09]); for more details see [Ger06,AGB+09]. A mode is simply a label, and a mode set is a collection of such labels (e.g., the DHS National Threat Advisory {`Green, Blue, Yellow, Orange, Red`}). Corresponding to each mode is a “bundle” of subscriptions, each with its own QoS+ requirements, represented in a rate-based forwarding table in each of the GridStat Forwarding Engines (FE). These modes can be switched quickly, with a choice of two algorithms: one that does a quick flooding on the data plane, and the other that provides stronger consistency via a multi-level commit mechanisms that utilizes the hierarchy in GridStat’s management plane, one that is designed to map naturally onto the hierarchy in the power grid [AGB+09].

Implicit in this design is that a mode implements coarse-level resource allocation by its forwarding tables (and, specifically, the rate each subscriber gets delivered at). Further, an FE can have multiple mode sets active, each with its own scope [AGB+09]. For example, the DHS set could be defined at the top and be given a certain percentage of capacity (perhaps 10%) in that FE, while a mode set defined at the ISO level could be given another portion of the capacity (another 10% or so), and a utility could have one or mode sets within its domain with the rest of the capacity. Even though an FE may be inside a utility(or at least an ISO), the entity that defines that mode set would be allowed to change its value (e.g., DHS decrees that we have moved from Orange to Red). This would affect resource allocation at all levels in the GridStat WAMS-DD.

Such mode change mechanisms are potentially useful for responding to power contingencies, as outlined in Section 3.8. However, when coupled with quick internal instrumentation and the other IGs, it enables very fast, coordinated system-level adaptations to be made in the face of an IT failure or cyber-attack. This is opposed to the uncoordinated per-flow adaptations that other technologies overviewed in 5.4 can only provide, because they are not tailored for a critical infrastructure and utilizing the IGs described above.

8.5.2 Actuator Remote Procedure Call with Safety

Publish-subscribe is inherently one-way communication, but sometimes a two-way roundtrip exchange is required. GridStat has such a two-way delivery mechanism called Ratatorkr[14], which is built on top of its QoS-managed, one-way delivery mechanisms. The capabilities of this mechanism has already been overviewed in Section 3.5.3; we now provide further details.

Remote procedure calls are a long-known technology [Iet76,BN84]. These mechanisms make a call to a procedure or routine on a remote computer look as similar to a local call as possible.

Ratatoskr is a remote procedure call mechanism tailored not only to exploit GridStat's delivery mechanisms but also to the needs of power applications. It is highly tunable by the application, and offers three distinct mechanisms for redundancy, offering tradeoffs between worst-case deadlines, use of network resources, and resiliency towards a variety of network failures.

Ratatoskr supports pre-conditions and post-conditions. A pre-condition is a predicate over live GridStat variables that resides in the server's proxy and can terminate a request from the client (caller) to the server (called routine) if appropriate in the face of safety concerns. For example, when the message with the client's request arrives at the server, the pre-condition is checked before the call is delivered to the server (which may be firmware for an actuator). This can be used, for example, if the sensor variables show that a line seems to be energized even though operators and control center software does not believe it is so. This is something that has killed utility technicians in the past, so in this case the call would be aborted (and an exception returned to the calling application) rather than carrying out the potentially dangerous actuator change command.

Further, even though a reply from a server comes back, it is possible that the actuator hardware has failed. Ratatoskr 's post-condition mechanism can help detect this. It allows the client's proxy to specify a predicate over live GridStat variables. This predicate is evaluated after the reply has returned and a application-specified delay is over. The typical case for such a post-condition is to specify a predicate over a few sensor variables that should have changed if the hardware actuator (for example, a relay) actually did what the client's call commanded it to do. If the predicate fails, the application will be notified.

For more information on Ratatoskr, see [VBG+10].

[14] In Norse mythology, Ratatorkr is a squirrel climbing around the great world tree Yggdrasil, ferrying messages and gossip between the mythological creatures living in its braches.

8.6 Status

The interdisciplinary GridStat project began in 1999 with two of the authors of this paper, power researcher Bose and applied computer scientist Bakken, and was joined in 2001 by author Hauser, another researcher in applied computer science. Rather than the usual academic approach of devising a clever (and publishable) solution then finding a problem to claim it solved, the project started out by a careful analysis of how much better communications could help the power grid (for example, see [BBB 00,BEB01,BBD+02]). Furthermore, it was clear that what was needed was not yet another "point solution" algorithm, no matter how publishable, but rather a more comprehensive middleware framework (no matter how less publishable). Towards that end, design of GridStat began in 2001, a first prototype demonstrated to project sponsor NIST in 2002, and live data from regional utility Avista started in 2003 (and is still in place as of 2010).

Since then, different prototypes of portions of GridStat have been done in the C++ and Java languages as well as with specialized network processor hardware [Swe09]; not every language is presently supported, and not all features developed by GridStat researchers are available in every language or system (or are presently supported). GridStat was deployed across multiple US national energy labs in a cyber-security assessment project 2008.

In 2010, the decision was made to offer to make GridStat open source and royalty-free, to serve as a reference implementation for the NASPInet Data Bus[15], which is hugely underspecified in our opinion[16]. However, that will require funding to transition it (including doing future work overviewed below in Section 10.2). That is, as a candid reality check, to "toss over the fence" a complex middleware framework without adequate infrastructure would be a bad experience for all involved. For users, trying to configure and monitor such a complex system without a set of sophisticated software tools and support staff would almost certainly be a disaster. For the GridStat project, releasing it so, from Day One, anyone anywhere could download it and ask questions and report bugs without adequate support infrastructure would be a huge waste of time.

9. Related Work

Much related work was already discussed in the context and structure of Section 5.4. Here we summarize other pertinent research.

LIPSIN is a publish-subscribe multicast forwarding fabric; [Jac+09] gives results of some small-scale simulations and an initial forwarding engine prototype. Like GridStat, is assumes that the network topology is fully known. It names the links in the network with large (~250 bit names) in which 1-bits are sparse. This allows multicast packets to be source-routed by a Bloom filter carried in each packet. With this design, forwarding engines need only logically AND each of their outgoing link names with the Bloom filter in a received packet in order to decide on which links the packet should be sent. The paper discusses mitigations for inevitable issues such as false positives and cycles. The basic idea of LIPSIN operates at a lower level than GridStat's mechanisms for redundant real-time, multi-cast and possibly would allow implementation of higher-performance GridStat forwarding engines.

Much more background information on WAMS-DD and its broader context (including the need for it) can be found in [BHG+07]. Background on how the power grid's structure in Europe is different from the USA, and ongoing research there (as of 2006) can be found in [BBH06].

[15] As Internet pioneer David Clark of MIT famously said in 1992, concerning developing (coherent and useable) new protocols: "We reject: kings, presidents, and voting. We believe in: rough consensus and running code."

[16] The SEL authors of this paper have no public opinion on this subject, this judgment is from the WSU authors.

10. Discussion

10.1 Summary

The electric power grids in the US, Europe, and elsewhere are all enjoying a "smart grid" renaissance that has the potential to make grids be more resilient, efficient, and eco-friendly. However, to date, most of the energy, discussion as well as investment from government and industry concerning "smart grids" has been on the distribution side, not involving the bulk power system (generation and transmission). We believe that this imbalance is most unfortunate, because, in addition to the distribution side, great improvements to the bulk power system can be made, but it requires sophisticated data delivery services that can be relied on. While much can be learned from existing commercial and research middleware systems, many aspects of these systems are different from the bulk power system, including real-time requirements, geographic scope, and the security threat profile.

In this paper, we have provided a detailed and holistic analysis of how the bulk power system can be improved by better data delivery services. We first overviewed fundamental problems facing today's bulk power systems: namely reliability, efficiency, and integration of renewable energy sources. We then described a wide range of techniques for improving the bulk power system using coherent, real-time data. In Section 4 we normalized the space of pertinent QoS+ requirements for data delivery, namely the following: latency, rate, criticality, quantity of data, geographic scope of delivery; and, for bulk data transfers, deadline. We then summarized the QoS+ requirements for the previously-described power techniques and solutions in terms of these normalized parameters, in order to allow "apples-to-apples" comparisons of implementation difficulty between said parameters.

The paper transitioned to details about requirements and implementations for WAMS-DD in Section 5. There, we described and justified the baseline data delivery requirement for WAMS-DD and provided detailed implementation guidelines that are helpful (and, in many cases, necessary) to meet the most stringent of WAMS-DD applications. We then analyzed existing networking and middleware technologies in terms of how well they covered the WAMS-DD delivery requirements and implementation guidelines. We then provided overviews of cyber-security issues for WAMS-DD, NASPInet, and GridStat.

10.2 Conclusions

Major conclusions from this paper include:

- WAMS-DD must support a very wide range of QoS+ properties.
- Network protocols developed by internet researchers, as well as power protocols, have very limited coverage of WAMS-DD delivery requirements and implementation guidelines.
- There is no commercial or military market for data delivery systems meeting the most extreme of WAMS-DD requirements, especially ultra-low latencies with hard guarantees over a wide area.
- Not surprisingly, given this lack of market, commercial middleware systems almost always have some gaps in terms of delivery requirements or implementation guidelines, though certainly some seem close and may well be able to be transitioned to fill these gaps.
- Just because a technology
    - is considered best practices in one industry
    - seems superficially similar to what is needed for WAMS-DD

  does not mean that is not a really bad idea for WAMS-DD. Case in point: web services.

- Despite being used as a WAMS-DD in some recently funded projects, MPLS and IP Multicast are very deficient (even when used together) compared to the actual requirements of WAMS-DD.
- GridStat is an existence proof that all of the delivery requirements, and almost all of the implementation guildelines, can be achieved with careful design and a good understanding of the requirements of leading-edge power application programs.

And, finally, one meta-conclusion: there are many pragmatic and important (but not academically publishable!) details and issues that any real WAMS-DD system has to address. We have gone to great lengths to explain them in this paper, because if WAMS-DD is not done right (as is quite plausible for a number of reasons), power grids worldwide, and the societies that depend upon them, may suffer unnecessarily.

10.3 Future GridStat Work

There are many areas of short-term, applied research and development which we hope to pursue with GridStat. These include:

- A set of software tools for configuring, testing, monitoring, and managing a GridStat deployment.
- Systematic internal ***instrumentation*** (**IG9**) that exploits the properties of a WAMS-DD in order to provide very fast detection of a wide range of benign failures, cyber-attacks, publishers emitting more updates than promised (and planned for by GridStat), etc.
- ***Policy language support*** for avoiding hard-coding the many things that tend to be hard-coded in today's power applications. This includes all manner of reusable policies for resource usage, access control, criticality of various subscriptions and application families, etc.
- Supporting a fully-functional management hierarchy, including muti-level policy language support, policy-driven aggregation of both internal instrumentation and key power indicators, etc.
- ***Adaptation strategies*** that are not on the level of a single subscription variable but rather can be specified (by reusable policies) at the level of an entire grid, portions of a grid, a utility, or portions of a utility. This has significant advantages to uncoordinated adaptations at the network or middleware layer, which is the only other choice barring development and support of such strategies. Adaptations would be to the various instrumentation targets outlined above, including cyber-attacks and benign failures.
- Systematic support for managing different kinds of ***aperiodic traffic*** from both the power domain (alarms/alerts, condition-based updates. etc) and the IT domain (instrumentation alerts, etc).
- Formal ***verification*** of deployed GridStat implementation to ensure it is meeting its DRs as long as its design constraints[17] are not violated.

Acknowledgements

We are very grateful to a number of key individuals who have provided guidance, support, and encouragement to this research, especially Jeff Dagle, Don Kopzcynksi, and Phil Overholt. We are also very grateful to the many people who have offered comments and insights on this research and the problem space of WAMS-DD. These include (in random order) Helen Gill, Damir Novosel, Vahid Madani, Mahendra Patel, Dave Hilt, Alison Silverstein, Dmitry Kosterev, Dan Brancaccio, Hank Kenchington, Dave Norton, Floyd Galvan, Lisa Beard, Supreet Oberoi, Stan Schneider, Gabriella Ciocarlie, Paul Myrda, Kris Koellner, Stan Johnson, Doug Maughan,

[17] The reader with a background in dependable computing may have correctly surmised that the informal term "design constraint" is really incorporating the notions of a failure model [ALR+04], and such verification necessarily involves application of a number of techniques including "assumption coverage" [Pow92]; we felt that this simplification was more appropriate given the interdisciplinary nature of this paper.

Carl Landwehr, Rick Schantz, Jessica Ascough, Ken Birman, Jim Thorp, Ken Hopkinson, Deb Frincke, Klara Nahrstedt, Neeraj Suri, Majid Khelil, Carl Landwehr, Daniel Germanus, Bob Thomas, Tony Johnson, Jay Giri, Kevin Tomsovic, Vijay Vittal, Sakis Meliopolis, David McKinnon, Dave Chassin, Ritchie Carroll, Ken Martin, Mladen Kezunovic Dave Anderson, Mani Venkatasubramanian, the members of the NASPI Data & Network Management Task Team, all of the present and past GridStat students, and anyone else who has provided useful inputs but whose name we inadvertently omitted.

This research has been supported in part by grants CNS 05-24695 (CT-CS: Trustworthy Cyber Infrastructure for the Power Grid(TCIP)) and CCR-0326006 from the US National Science Foundation; and by grant #60NANB1D0016 (Critical Infrastructure Protection Program) from the National Institute of Standards and Technology, in a subcontract to Schweitzer Engineering Labs Inc. We thank the US Dept. of Energy and Dept. of Homeland Security for their part in funding the NSF TCIP center. This research was also funded in part by Department of Energy Award Number DE-OE0000097 (TCIPG).

Disclaimer: "This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."

References

[ADS00] Yair Amir, Claudia Danilov, and Jonathan Stanton. "A Low Latency, Loss Tolerant Architecture and Protocol for Wide Area Group Communication", in *Proceedings of the First International Conference on Dependable Systems and Networking (DSN 2000),* IEEE, New York, June 2000.

[AE04] A. Abur and A. G. Exposito, *Power System State Estimation,Theory and Implementation*. New York: Marcel Dekker, 2004.

[Aga06] American Gas Assoc. "*Cryptographic Protection of SCADA Communications,*" 2006.

[AGB+09] Stian F. Abelsen and Harald Gjermundrød and David E. Bakken and Carl H. Hauser. "Adaptive Data Stream Mechanism for Control and Monitoring Applications". In Proceedings of 1st International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE'09), Athens, Greece, November 2009, 86–91.

[AKR+09] S. D'Antonio, A. Khelil, L. Romano, and N. Suri, "Increasing Security and Protection of SCADA Systems through Infrastructure Resilience," International Journal of System of Systems Engineering (IJSSE), vol. 1, no. 4, pp. 401–413, 2009.

[ALR+04] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr. "Basic Concepts and Taxonomy of Dependable and Secure Computing", *IEEE Transactions on Dependable and Secure Computing*, 1(1), January 2004, 11–33.

[And09] Roger Anderson, "The Con Edison/Boeing/Columbia/NYC Economic Cprt. Smart Grid Project in New York City and Suburbs", http://eesc.columbia.edu/files/uploaded/file/AndersonSmartGrid.pdf.

[ARK+08] S. D'Antonio, L. Romano, A. Khelil, and N. Suri, "INcreasing Security and Protection through Infrastructure REsilience: the INSPIRE Project," in Proceedings of The 3rd International Workshop on Critical Information Infrastructures Security (CRITIS'08), October 2008.

[Bak09] David Bakken. *Quality of Service Design Considerations for NASPInet*. Presentation to the North American Synchrophasor Initiative (NASPI) Work Group meeting, Scottsdale, AZ February 4, 2009.

[BBB00] Bakken, D. and Bose, A. and Bhowmik, S. "Survivability and Status Dissemination in Combined Electric Power and Computer Communications Networks", in *Proceedings of the Third Information Survivability Workshop* (ISW-2000), CERT, October, 2000, Boston, MA.

[BBD+02] Bakken, D, Bose, A., Dyreson, C., Bhowmik, S, Dionysiou, I., Gjermundrod, H. and Xu, L. "Impediments to Survivability of the Electric Power Grid and Some Collaborative EE-CS Research Issues to Solve Them", In *Proceedings of the Fourth Information Survivability Workshop*, IEEE, Vancouver, Canada, March 2002.

[BBH06] Bakken, D, Bose, A., and Hauser, C. *EC Efforts in SCADA-Related Research: Selected Projects.* Technical Report EECS-GS-008, Washington State University, 20 October, 2006. Available via http://www.gridstat.net/publications/EC-SCADA-CIP-Report.pdf.

[BCH+05] Kenneth P. Birman, Jie Chen, Kenneth M. Hopkinson, Robert J. Thomas, James S. Thorp, Robbert van Renesse, and Werner Vogels, "Overcoming Communications Challenges in Software for Monitoring and Controlling Power Systems", *Proceedings of the IEEE*, 9(5), May 2005.

[BEB01] Bakken, D., Evje, T., and Bose, A. "Survivable Status Dissemination in the Electric Power Grid", in Proceedings of the Information/System Survivabilty Workshop, in Supplement Proceedings of the International Conference on Dependable Systems and Networks (DSN-2001), IEEE/IFIP, Göteberg, Sweden, July 2001.

[BHG87] P. Bernstein, V. Hadzilacos, and N. Goodman. Concurrency Control and Recovery in Database Systems, Addison-Wesley, 1987.

[BHG+07] D. Bakken, C. Hauser, H. Gjermundrød, and A. Bose. *Towards More Flexible and Robust Data Delivery for Monitoring and Control of the Electric Power Grid***,** Technical Report EECS-GS-009, Washington State University, May 2007.

[BHK+10] Rakesh Bobba, Eric Heine, Himanshu Khurana, and Tim Yardley. "Exploring a Tiered Architecture for NASPInet", in *Proceedings of the IEEE Conference on Innovative Smart Grid Technologies*, Gaithersberg, MD, January 2010.

[Bir04] Ken Birman, "Like it or not, Web Services are Distributed Objects!", *Communications of the ACM*, 47:12, 60–62

[Blackout2003] U.S.-Canada Power System Outage Task Force. Final Report on the August 14th, 2003 Blackout in the United States and Canada, 2004. https://reports.energy.gov/

[Bir06] Ken Birman, "The Untrustworthy Web Services Revolution". *IEEE Computer*, 39:2, Feb., 2006, 98-100.

[BMR+09] Jack Brassil, Rick McGeer, Raj Rajagopalan, Puneet Sharma, Praveen Yalangadula, Sujata Banerjee, David P. Reed, Sung-Ju Lee, Andy Bavier, Larry Peterson, Stephen Schwab, Larry Roberts, Alex Henderson, Bob Khorram, Shidong Zhang, Soonyong Sohn, Brian Mark, John Spies, Nicki Watts, The CHART System: A High-Performance, Fair Transport Architecture Based On Explicit-Rate Signalling, *ACM SIGOPS Operating Systems Review*, January 2009.

[BN84] Andrew Birrell and Bruce Nelson, Implementing Remote Procedure Calls, *ACM Transactions on Computing Systems*, 2(1), February 1984, 39-59.

[Boe10] Boeing Defense, Space, and Security. "System of Systems Common Operating Environment", http://www.boeing.com/bds/soscoe/SOSCOE_overview.pdf.

[BST09] David E. Bakken, Richard E. Schantz, and Richard D. Tucker. “Smart Grid Communications: QoS Stovepipes or QoS Interoperability”, in *Proceedings of Grid-Interop 2009*, GridWise Architecture Council, Denver, Colorado, November 17-19, 2009. Available http://gridstat.net/publications/TR-GS-013.pdf. **Best Paper award** for “Connectivity” Track.

[BTB04] Sudipto Bhowmik, Kevin Tomsovic, and Anjan Bose. “Communication models for third party load frequency control.” IEEE Transactions on Power Systems, 19:1, February 2004, 543–548.

[Cha09] Chakravarthy, Rasika. *Long-lived authentication protocols for critical infrastructure process control systems*. Masters Thesis, Washington State University, 2009.

[CHB10] R. Chakravarthy, C. Hauser, and D. Bakken. Long-lived authentication protocols for critical infrastructure process control systems, *Fourth IFIP WG 11.10 Int’l Conf. on Critical Infrastructure Protection*”, Washington, D.C., March, 2010.

[CKT+09] S. Chakrabarti, E. Kyriakides, B. Tianshu, C. Deyu Cai, and V. Terzija. “Measurements Get Together”. *Power and Energy Magazine*, IEEE, 7(1), January 2009, 41–49.

[CKV01] Chockler, Gregory V. and Keidar, Idit and Vitenberg, Roman, “Group Communication Specifications: A Comprehensive Study”, ACM Computing Surveys, 33(4), December 2001,1–43.

[DFH+07] Ioanna Dionysiou, Dehorah Frincke, Carl Hauser, and Dave Bakken, “An Approach to Trust Management Challenges for Critical Infrastructures”, *Lecture Notes in Computer Science 5141*, Springer, Berlin, 2007.

[DoE06] US Dept. of Energy, *Roadmap to Secure Control Systems in the Energy Sector*, January 2006.

[DSU03] Xavier Défago and André Schiper and Péter Urbán, “Totally Ordered Broadcast and Multicast Algorithms: Taxonomy and Survey”, 36(4):372-421, December 2004.

[Eco04] *The Economist*, “Building the Energy Internet”, 11 May 2004 (Technology Quarterly section).

[EFG+03] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. “The Many Faces of Publish-Subscribe”. *ACM Computing Surveys*, Vol. 35, No. 2, June 2003, pp. 114–131.

[Ele04] Electric Power Research Institute (EPRI), The Integrated Energy and Communication Systems Architecture, Vol. IV: Technical Analysis, 2004.

[Far09] Adrian Farrel (ed*). Network Quality of Service Know it All*, Elsevier, 2009.

[HNM+08] Horowitz, S. Novosel, D. Madani, V. Adamiak, M. “System-Wide Protection”, *IEEE Power & Energy Magazine*, 6(5), September 2008, 34-42.

[GBH+09] K. Harald Gjermundrød, David E. Bakken, Carl H. Hauser, and Anjan Bose. “GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid”, *IEEE Transactions on Power Delivery*, 24(1), 2009, 136–143.

[GDH+03] K. Harald Gjermundrød, Ioanna Dionysiou, Carl Hauser, Dave Bakken, and Anjan Bose “Flexible and Robust Status Dissemination Middleware for the Electronic Power Grid”. *Technical Report EECS-GS-003*, School of Electrical Engineering and Computer Science, Washington State University, September 2003.

[GGC+04] Christopher Gill, Jeanna Gossett, David Corman, Joseph Loyall, Richard Schantz, Michael Atighetchi, and Douglas Schmidt. Integrated Adaptive QoS Management in Middleware: A Case Study. 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004), Toronto, Canada, May 25-28, 2004.

[GDG+10] Daniel Germanus, Ionna Dionysiou, Harald Gjermundrød, Abdelmajid Khelil , Neeraj Suri , David E. Bakken, and Carl H. Hauser. "Leveraging the Next-Generation Power Grid: Data Sharing and Associated Partnerships", In *Proc. of The first IEEE Conference on Innovative Smart Grid Technologies (ISGT) Europe*, Göteberg, Sweden, October 2010, *to appear*.

[Gje06] K. Harald Gjermundrød. *Flexible QoS-managed status dissemination middleware framework for the electric power grid.* PhD Dissertation, Washington State University, August 2006.

[GKS10] D. Germanus, A. Khelil, and N. Suri, "Increasing the Resilience of Critical SCADA Systems Using Peer-to-Peer Overlays," in ISARCS 2010, 1st International Symposium on Architecting Critical Systems, ser. LNCS, no. 6150. Springer, June 2010, pp. 161–178.

[Gr06] GridWise Architecture Council, *Interoperability Constitution Whitepaper (v1.1),* December 2006.

[HBD+07] Carl H. Hauser, David E. Bakken, Ioanna Dionysiou, K. Harald Gjermundrod, Venkata S. Irava, Joel Helkey, and Anjan Bose. "Security, Trust and QoS in Next-generation Control and Communication for Large Power Systems." *International Journal of Critical Infrastructures (Inderscience),*2007.

[Hel07] Helkey, J. Achieving end-to-end delay bounds in a real-time status dissemination network. Masters Thesis, Washington State University, 2007.

[HPR10] S. Horowitz, A. Phadke, and B. Renz. "The Future of Power Transmission", *IEEE Power and Energy Magazine*, 8(2), March-April 2010, 34-40.

[HRW+09] K. Hopkinson, G. Roberts, X. Wang, and J. Thorp, Quality of Service Considerations in Utility Communication Networks, *IEEE Transactions on Power Delivery*, 24(3), July 2009.

[Hu09] Hu, Yi. Data Bus Technical Specifications for North American Synchrophasor Initiative Network (NASPInet). North American Synchrophasor initiative, May 2009.

[IEEE-1547.3] IEEE Guide for Monitoring, Information Exchange, and Control of Distributed Resources Interconnected with Electric Power Systems. IEEE Std 1547.3^{TM}-2007.

[Iet01] PGM Reliable Transport Protocol Specification, RFC 3208, IETF, 2001.

[Iet76] A High-Level Framework for Network-Based Resource Sharing, RFC 707, IETF, 1976.

[IH05] V.S. Irava, and C. Hauser, "Survivable low-cost low-delay multicast trees", *in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM),* Nov. 2005.

[Ins10] "INcreasing Security and Protection through Infrastructure REsilience",http://www.inspire-strep.eu/.

[Ira06] V.S. Irava, "Low-cost delay-constrained multicast routing heuristics and their evaluation," PhD Dissertation, Washington State University, August 2006.

[JAC+09] Petri Jokela, András Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, Pekka Nikander. "LIPSIN: line speed publish/subscribe inter-networking". in *Proceedings of SIGCOMM 2009*, ACM, August 2009, Barcelona, 195–206**.**

[JHG+06] Johnston, R.A., Hauser, C.H., Gjermundrod, K.H., Bakken, D.E. "Distributing Time-Synchronous Phasor Measurement Data Using the GridStat Communication Infrastructure." In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06),* 2006.

[JTT+07] Anthony Johnson, Robert Tucker, Thuan Tran, Dan Sullivan, Chris Anderson and Dave Whitehead, "Static Var Compensation Controlled via Synchrophasors", Western Protective Relay Conference, Spokane, WA, 2007.

[KJD+10] A. Khelil, S. Jeckel, D. Germanus, and N. Suri, "Towards Benchmarking of P2P Technologies from a SCADA Systems Protection Perspective," in Proceedings of The 2nd International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT), 2010.

[KKK+01] Krishnamurthy Y, Kachroo V, Karr DA, Rodrigues C, Loyall JP, Schantz RE, Schmidt DC. Integration of QoS-Enabled Distributed Object Computing Middleware for Developing Next-Generation Distributed Applications. In *Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems (OM 2001),* June 18, 2001, Snowbird, Utah.

[Kle09] Stanley Klein, *Methods and System to Manage Variability in Production of Renewable Energy*, US Patent 20090216387, 2009.

[KRK91] M. Klein, G. J. Rogers, and P. Kundur, “A fundamental study of inter-area oscillations in power systems,” *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 914–921, August 1991.

[Kro06] Kropp, T. “System Threats and Vulnerabilities”, IEEE Power & Energy Magazine, 4(2), March/April 2006, 46–50.

[LBS+98] Loyall JP, Bakken DE, Schantz RE, Zinky JA, Karr DA, Vanegas R, Anderson KR. QoS Aspect Languages and Their Runtime Integration. *Lecture Notes in Computer Science*, Vol. 1511, Springer-Verlag. Proceedings of the Fourth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers (LCR98), 28-30 May 1998, Pittsburgh, Pennsylvania.

[MC08] K. Martin and J. Carroll, “Phasing in the Technology”, *Power and Energy Magazine*, IEEE, 6(5), September 2008, 24-33.

[MJG+06] E. Martínez, N. Juárez, A. Guzmán, G. Zweigle, and J. León, “Using synchronized phasor angle difference for wide-area protection and control,” in *33rd Annual* Western Protective Relay Conference, Spokane, WA, October 17–19, 2006.

[MLD09] W. Mahnke, S. Leitner, and M. Damm. *OPC Unified Architecture*, 4 May 2009.

[MPA+04] Roy Moxley PE, Chuck Petras PE, Chris Anderson, and Ken Fodero II. Display and Analysis of Transcontinental Synchrophasors, *Western Power Delivery and Automation Conference*, 2004.

[MSM+09] J. Mulhausen, J. Schaefer, M. Mynam, A. Guzmán, and M. Donolo, “Anti-islanding today, successful islanding in the future,” in 36th Annual Western Protective Relay Conference, Spokane, WA, October 20–22, 2009.

[Mut08] Muthuswamy, S. System implementation of a real-time, content based application router for a managed publish-subscribe system. Masters Thesis, Washington State University, 2008.

[Nor08] North American Synchrophasor Initiative, “Quanta Statement of Work”.

[Nor09] Dave Norton, Security Strategy for NASPI Implementation at Entergy, presentation at NASPI meeting, February 5, 2009, Scottsdale, AZ.

[NKM+07] J. Nutaro, P.T. Kuruganti, L. Miller, S. Mullen, M. Shankar. Integrated Hybrid-Simulation of Electric Power and Communications Systems, in *Proceedings of the 2007 Power Engineering Society General Meeting*, 2007. IEEE , 1-8, June 2007.

[OMG07] Object Management Group, Data Distribution Service for Real-Time Systems v1.2, 2007.

[Ope10] Open Secure Energy Control Systems, LLC (OSECS), www.osecs.com.

[PCS+05] Perrig A, Canetti R, Song D, Tygar D, Briscoe B (2005) Timed efficient stream loss-tolerant authentication (TESLA): multicast source authentication transform introduction. http://www.ietf.org/rfc/rfc4082.txt .

[PM08] A. Phadke and R. de Moraes, “The Wide World of Wide-Area Measurement”, *Power and Energy Magazine*, IEEE, 6(5), September 2008, 52-65.

[Pow92] David Powell, “Failure Mode Assumptions and Assumption Coverage”, in *Proceedings of the Twenty Second International Symposium on Fault-Tolerant Computing (FTCS-22)*, IEEE, Boston, MA, July, 1992, 386–395.

[PT08] A. G. Phadke and J. S. Thorp. *Synchronized Phasor Measurements and Their Applications*. Springer, 2008.

[PJM07] PJM, *PJM 2007 Strategic Report*, April 2, 2007, http://www2.pjm.com/documents/downloads/strategic-responses/report/20070402-pjm-strategic-report.pdf.

[PLS+00] Pal PP, Loyall JP, Schantz RE, Zinky JA, Shapiro R, Megquier J. Using QDL to Specify QoS Aware Distributed (QuO) Application Configuration. Proceedings of ISORC 2000, In *Proceedings of the 3rd IEEE International Symposium on Object-Oriented Real-time distributed Computing*, March 15–17, 2000, Newport Beach, CA.

[RHJ+09] Kristin Rauschenbach, Regina Hain, Alden Jackson, John Jacob, Will Leland, John Lowry, Walter Milliken, Partha Pal, Ram Ramanathan, Cesar Santivanez, "Dynamic provisioning system for bandwidth-scalable core optical networks," in *Proceedings of MilCom 2009*.

[Rob09] Lawrence Roberts. “A Radical New Router”. *IEEE Spectrum*, July 2009, 35–39.

[RFC-3031] E. Rosen, A. Vishanathan, R. Callon. RFC-3031: Multiprotocol Label Switching Architecture. The Internet Society, 2001. http://datatracker.ietf.org/doc/rfc3031/

[RFC-3697] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering. RFC3697: IPv6 Flow Label Specification. The Internet Society, 2004. http://www.faqs.org/rfcs/rfc3697.html

[RTI10a] RTI Data Distribution Service, http://www.rti.com/products/dds/.

[RTI10b] RTI, “Getting Started Guide”, http://research.rti.com/docs/pdf/RTI_DDS_GettingStarted.pdf .

[SS02] Richard E. Schantz and Douglas C. Schmidt. *Research Advances in Middleware for Distributed Systems: State of the Art*. IFIP World Computer Congress, August 2002, Montreal, Canada.

[SLA+02] Schantz RE, Loyall JP, Atighetchi M, Pal PP. Packaging Quality of Service Control Behaviors for Reuse. Proceedings of ISORC 2002, The 5th IEEE International Symposium on Object-Oriented Real-time distributed Computing, April 29 - May 1, 2002, Washington, DC.

[SHC+09] E. Solum, C. Hauser, R. Chakravarthy, and D. Bakken. Modular over-the-wire configurable security for long-lived critical infrastructure monitoring systems, *Proc. of the 3rd ACM Int’l Conf. on Distributed Event-Based Systems (DEBS 2009)*, Nashville, TN, July 2009.

[Sol07] Solum, Erik. *Achieving over-the-wire configurable confidentiality, integrity, authentication and availability in GridStat's status dissemination.* Masters Thesis, Washington State University, 2007.

[Spr10] www.spreadconcepts.com.

[SW07] E. O. Schweitzer, III and D. E. Whitehead, “Real-time power system control using synchrophasors,” in 34th Annual Western Protective Relay Conference, Spokane, WA, October 16–18, 2007.

[SW08] E. O. Schweitzer, III and D. Whitehead, “Real-world synchrophasor solutions,” in 35th Annual Western Protective Relay Conference, Spokane, WA, October 21–23, 2008.

[SW70] F. C. Schweppe and J. Wildes, “Power System Static-State Estimation, Part I: Exact Model,” *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-89, pp. 120–125, Jan. 1970

[Swe09] K. Swenson. Exploiting network processors for low latency, high throughtput, rate-based sensor updated delivery. MS Thesis, Washington State University, 2009.

[SLA+02] Schantz RE, Loyall JP, Atighetchi M, Pal PP. Packaging Quality of Service Control Behaviors for Reuse. Proceedings of ISORC 2002, *in Proceedings of the 5th IEEE International Symposium on Object-Oriented Real-time distributed Computing*, April 29 - May 1, 2002, Washington, DC.

[SRC84] J. Saltzer, D. Reed, and D. Clark. “End-to-End Arguments in System Design”. *Transactions on Computer Systems*, Association of Computing Machinery (ACM), 2(4), November 1984, 277–288.

[STB86] Richard E. Schantz, Robert H. Thomas, Girome Bono. “The Architecture of the Cronus Distributed Operating System”. In *Proceedings of the 6th International Conference on Distributed Computing Systems (ICDCS86),* IEEE Computer Society Press, Cambridge, Massachusetts, USA, May 19-13, 1986, 250-259.

[Sys10] www.soscoe.com

[SWZ+09] Edmund O. Schweitzer III, David Whitehead,Greg Zweigle, Krishnanjan Gubba Ravikumar, “Synchrophasor-Based Power System Protection and Control Applications, Western Protective Relay Conference, Spokane, WA, 2009.

[TAB+08] J. Thorp, A. Abur, M. Begovic, J. Giri, and R. Avila-Rosales, “Gaining a Wider Perspective”, *Power and Energy Magazine*, IEEE, 6(5), September 2008, 43-51.

[TBV+05] K. Tomsovic, D. Bakken, M. Venkatasubramanian, and A. Bose, Designing the Next Generation of Real-Time Control, Communication and Computations for Large Power Systems. In *Proceedings of the IEEE* (Special Issue on Energy Infrastructure Systems), 93(5), May 2005.

[TPZ+08] Daniel J. Trudnowski, John W. Pierre, Ning Zhou, John F. Hauer, and Manu Parashar, “Performance of Three Mode-Meter Block-Processing Algorithms for Automated Dynamic Stability Assessment, *IEEE Transactions on Power Systems*, 23(2), May 2008.

[TS08] Tsang, P.P. and Smith, S.W., 2008, in IFIP International Federation for Information Processing, Volume 278; in *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 445–459.

[VBG+10] Erlend Viddal, David. Bakken, K. Harald Gjermundrød, and Carl Hauser. “Wide-Area Actuator RPC over GridStat with Timeliness, Redundancy, and Safety”, in 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'10), February 2010, Krakow, Poland, 17–24.

[WKH+09] Wang, Q., Khurana, H., Huang,Y. Nahrstedt, K. “Time Valid One-Time Signature for Time-Critical Multicast Data Authentication,” IEEE InfoComm 2009, pp. 1233-1241.

[WEB10] Jan Åge Walseth, Jan Eskedal and Øyvind Breidablik. "Analysis of Misoperations of Protection Schemes in the Nordic Grid 1st of December 2005." *Protection, Automation and Control World*, March 2010.

[XCN04] Yuan Xue, Kai Chen, and Klara Nahrstedt. “Achieving proportional delay differentiation in wireless LAN via cross-layer scheduling”. *Wireless Communications and Mobile Computing*, 4(8): 2004, 849-866.

[XMV+02] Xie, Z., Manimaran, G, Vittal, V., Phadke, A., and Centeno, V. An Information Architecture for Future Power Systems and Its Reliability Analysis, *IEEE Transactions on Power Systems*, 17:3, August 2002, 857–863.

[YSB09] T. Yang, H. Sun, and A. Bose, "Two-level PMU-based Linear State Estimator," in *Proceedings of the IEEE PES Power Systems Conference & Exposition (PSCE)*, Seattle, Washington, March 15-18, 2009, pp. 1-6.

[ZBS97] John A. Zinky, David E. Bakken, and Richard E. Schantz, “Architectural Support for Quality of Service for CORBA Objects”. Theory and Practice of Object Systems, April 1997.

Bio Briefs

Dr. David E. Bakken is an Associate Professor of Computer Science at Washington State University. His expertise includes designing, implementing, and deploying middleware frameworks supporting multiple QoS/security properties for wide-area networks. He is Co-Editor-in-Chief, IEEE Communications Society Smart Grid Vision (SGV) workshop, October 2010, and the follow-on report. Dr. Bakken has been working closely with WSU power researchers since 1999 on rethinking the grid's limited communications and developing the GridStat middleware framework. GridStat has had a large impact on the shape of NASPInet. Prior to joining WSU, he was a scientist at BBN Technologies where he was an original co-inventor of the Quality Objects (QuO) framework. QuO has been fielded in various demonstrations and evaluations, and its foundational paper [ZBS97] has been cited over 500 times as of 2010. Dr. Bakken has consulted for Amazon.com, Network Associates Labs (formerly Trusted Information Systems), Real-Time Innovations, Harris Corp., TriGeo Network Security, and others; he has also worked as a software developer for Boeing and has served as member of the Board of Directors for TriGeo.

Dr. Anjan Bose is a Regents Professor and the Distinguished Professor of Power Engineering at Washington State University, where he also served as the Dean of the College of Engineering & Architecture from 1998 to 2005. He has over 35 years of experience in the power industry and academe. His pioneering work in developing and implementing real time analysis software for power grid control centers was cited in his election to Fellow of the Institute of Electrical & Electronics Engineers (IEEE) and and as a Foreign Fellow of the Indian National Academy of Engineering. His work in the development of real time simulators, which are used around the world for training grid operators, was cited in his election to the US National Academy of Engineering. He was also recognized by the IEEE with their Outstanding Power Engineering Educator Award, the Third Millenium Medal and the Herman Halperin Electric Transmission & Distribution Award. He has consulted on power system operation for numerous companies and governments all over the world.

Dr. Carl H. Hauser is an Associate Professor of computer science with the School of Electrical Engineering and Computer Science, Washington State University (WSU), Pullman. His research interests include concurrent programming models and mechanisms, networking, programming language implementation, and distributed computing systems. Prior to joining WSU, he worked at the Xerox Palo Alto Research Center and IBM Research for a total of more than 20 years and was a coauthor of a seminal paper on epidemic multicast algorithms.

Dr. Edmund O. Schweitzer, III is recognized as a pioneer in digital protection and holds the grade of Fellow of the IEEE, a title bestowed on less than one percent of IEEE members. In 2002, he was elected a member of the National Academy of Engineering. He is the recipient of the Graduate Alumni Achievement Award from Washington State University and the Purdue University Outstanding Electrical and Computer Engineer Award. In September 2005, he was awarded an honorary doctorate from Universidad Autónoma de Nuevo León in Monterrey, Mexico, for his contribution to the development of electric power systems worldwide. He has written dozens of technical papers in the areas of digital relay design and reliability and holds more than 30 patents pertaining to electric power system protection, metering, monitoring, and control. Dr. Schweitzer served on the electrical engineering faculties of Ohio University and Washington State University, and in 1982, he founded Schweitzer Engineering Laboratories, Inc. to develop and manufacture digital protective relays and related products and services. Today, SEL is an employee-owned company, which serves the electric power industry worldwide, and is certified to the international quality standard ISO-9001. SEL equipment is in service at voltages from 5 kV through 500 kV, to protect feeders, motors, transformers, capacitor banks, transmission lines, and other power apparatus.

David E. Whitehead, P.E., is the vice president of research and development at SEL. Prior to joining SEL, he worked for General Dynamics Electric Boat Division as a combat systems engineer. He received his BSEE from Washington State University in 1989, his MSEE from Rensselaer Polytechnic Institute in 1994, and is pursuing his Ph.D. at the University of Idaho. He is a registered professional engineer in Washington and Maryland and a senior member of the IEEE. Mr. Whitehead holds seven patents with several other patents pending. He has worked at SEL since 1994 as a hardware engineer, research engineer, and a chief engineer/assistant director and has been responsible for the design of advanced hardware, embedded firmware, and PC software.

Gregary C. Zweigle received his master of science in electrical engineering and master of science in chemistry degrees from Washington State University. He also received a bachelor of science in physics from Northwest Nazarene University. He is presently a research and engineering manager at SEL. He previously worked as a principal research engineer in the research group and as a senior software developer at SEL. He has been responsible for phasor measurement unit signal processing algorithms, embedded system architectures, and synchrophasor-based power system solution designs. Mr. Zweigle holds seven patents and is presently pursuing a Ph.D. at Washington State University focusing on energy systems. He is a member of the IEEE and the American Chemical Society.