2015 Georgia Tech Annual
Fault & Disturbance Analysis Conference
April 27-28, 2015 Atlanta, GA

Open Source Software for Phasor Data Transport Security and Resiliency

Fred L. Elmendorf; F. Russell Robertson
Grid Protection Alliance
Chattanooga, TN USA
felmendorf@gridprotectionalliance.org

Abstract— **Changes in technology and infrastructure over the past five years have resulted in a new high precision data source, with high value, that is largely untapped at this point. Phasor measurement units (PMUs) record phasor quantities with high precision time stamps (synchrophasors) to allow the synchronized analysis of measurements from geographically separated measurement points. This data was initially envisioned as the raw material for new techniques in monitoring and visualizing wide area system phenomena such as power flows, oscillations, among others. While the use of synchrophasor data in wide area solutions has been extremely valuable, new techniques are emerging to take advantage of this data in solving local system problems. Building robust and resilient real-time systems using synchrophasor data exposes new data challenges, such as cybersecurity and data availability.**

This paper describes new open source software (OSS) opportunities that can increase the density of synchrophasor measurements using new functionality from existing substation intelligent electronic devices (IEDs as well as the communication and transport challenges that are inherent in using this relatively new data source. The paper presents an overview of open source software (OSS) tools available for managing, archiving, and securely sharing synchrophasor data, and a detailed description of an OSS project to explicitly address the unique challenges of cybersecurity and data resiliency of synchrophasor data streams at the substation level.

Keywords—operational data, security, open source software

I. THE PHASOR DATA MANAGMENET PROBLEM

There has been rapid growth of synchrophasor data systems and infrastructure over the last four years in the United States with most systems being designed and implemented based on traditional information technology best practices [i]. It is becoming increasingly apparent that new technology is needed to effectively collect, manage, analyze synchrophasor data in real-time and using off-line tools. Specifically, the following elements seem to be emerging as common areas for improvement in existing synchrophasor data infrastructure:

- Simplified implementation of end-to-end configuration and change management
- Identification and management of bad data
- Capability to easily integrate with existing legacy systems
- Data management and storage designed for phasor data volume and speeds
- Analytic applications that extract valuable information from synchrophasor data

In addition, security is becoming an increasingly important concern both for the implementation of genuine, best-practice security elements and to create the documentation necessary for security compliance.

II. OSS AND THE ELECTRIC INDUSTRY

Open source network and system monitoring tools have been a significant part of the tool kit for assuring information security for some time. However, there is a limited amount of OSS specifically targeted for electric power system applications. This list of applications is growing as the benefits of OSS are becoming better understood by the utility industry. These benefits include:

- Better code and application quality
- Development process improves security and performance
- Freedom from vendor lock in
- Accelerates collaboration and promotes innovation

In the April 2014 Future of Open Source Survey[ii], Black Duck Software determined that the debate over the appropriateness of use of open source software was over.

- 30% of participating corporations “make it easy” for employees to participate in OSS projects

- 50% see OSS as a means to retain competitive advantage
- 68% see OSS as a way to lower cost and improve efficiency
- 72% of those selecting OSS say that "many eyes" makes OSS more secure

In the late 2014, EPRI conducted a survey of electric utilities. Preliminary results suggest that of the respondents:

- Over half agree that the U.S. industry overall has embraced and is using OSS
- A significant majority don't believe that OSS has been largely embraced in U.S. electric industry
- More than half view OSS as a strategy to build utility partnerships
- Very few know where to find OSS designed for electric utility use

To help address this last bullet, on March 23, 2015 NASPI announced[iii] the creation of the "Phasor Software Exchange" which is a library of software, much of it free, open source software, for the collection, management, analysis and visualization of synchrophasor data.

III. New Sources for Synchrophasor Data

The initial deployment of synchrophasor devices consisted of dedicated single purpose PMUs which introduced additional burden on power systems to design, deploy, and maintain additional substation hardware. More recently, many IED manufacturers have added PMU functionality to their existing devices including PQ monitors, DFRs, and solid state relays. In addition to providing new multi-function hardware offerings, some manufacturers provide software or firmware upgrades that can be applied in the field to a power company's installed infrastructure. These new hardware, software, and firmware solutions make it possible to dramatically increase the density of synchrophasor data, which also increases the demand for communications bandwidth and cybersecurity measures. To take advantage of the potential value in more ubiquitous synchrophasor data, communication networks and data management tools must be carefully designed and managed.

IV. The Gateway Exchange Protocol

The Gateway Exchange Protocol is an open source measurement-based publish/subscribe transport protocol used to securely exchange time-series data and automatically synchronizing meta-data between two applications. The protocol supports sending real-time and historical data at full or down-sampled resolutions. When sending historical data the replay speed can be controlled dynamically for use in visualizations to enable users to see data faster or slower than recorded in real-time. GEP streaming communication speed can be set to "as-quickly-as-possible" as is typically desirable for system-to-system communication.

For phasor data, use of GEP overcomes the scaling limits imposed by frame-based protocols. By their nature, configuration frames in frame based protocols are much larger than data frames and can quickly exceed the practical 32K UDP data packet size limit. Synchrophasor implementations often use a combination of TCP and UDP to help postpone frame size limitations using TCP to send the configuration frame and UDP to send data frame; however, frame based protocols have a fixed maximum number of values that can be transmitted in one frame even when the transport is TCP. Use of a measurement-based protocol, like GEP, overcomes these issues.

Additionally, using GEP provides the following benefits over frame-based protocols:

- Automatic exchange of authorized metadata information between GEP appliances (the publisher approves subscriber data access based on data filters – which can be broad or as specific as measurement level approval.)
- Reduces bandwidth required for communicating data using lossless compression techniques.
- Allows the subscribing GEP appliance to start and stop the data stream as needed.
- Allows the subscribing GEP appliance to dynamically change the measurements points which are streaming to it.
- Reduced latency for most synchrophasor data architectures since data is communicated "on receipt" without the need for data concentration into time-based frames.

To test the impact of sending data in small packets rather that large (often very large) frames, GEP has been tested using UDP over the internet with a set of approximately 125 million phasor measurements using 4 different protocols: IEEE C7.118-2005, IEC 61850-90-5 with no frame retransmission, IEC 61850-90-5 with one frame retransmission, and GEP. The results shown in Figure 1 below show the advantage of the small GEP packet size on reducing data loss.

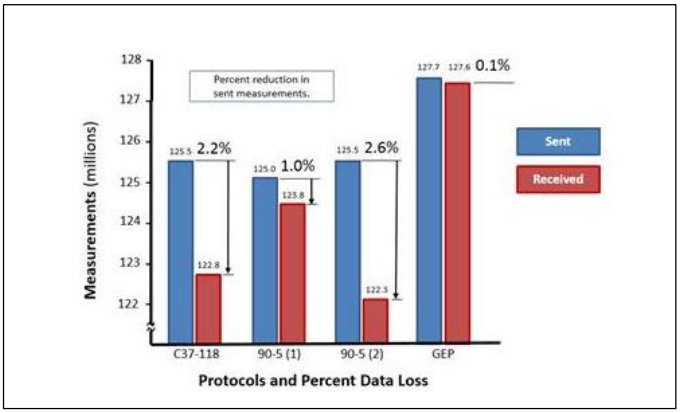

Figure 1. GEP Data Loss vs. Framebased Protocols

The wire protocol employed by GEP implements a publish/subscribe data exchange model using a simple

command driven service with tightly compressed, fast binary serialization of time-series values. The protocol does not require a predefined or fixed configuration – that is, the time-series values arriving in one data packet can be different than those arriving in another. Each packet data consists of a collection of time-series values; each time-series value is a structure containing an ID, a time-stamp, a value and associated flags.

The GEP is implemented using a TCP/IP command channel (for actions such as subscription) and optionally a UDP/IP data channel (the actual data to be transmitted). The TCP command channel is used to reliably negotiate session specific required communication, state and protocol parameters. It is used to authenticate with other GEP communications appliances, exchange metadata on points with them, and request points for subscription. The UDP data channel is used to send compact, binary encoded packets of identifiable measured values along with a timestamp accurate to one ten-millionth of a second (e.g., a tick) and flags that can be used to indicate time and data quality. When the UDP data channel is not used, data is transmitted on the TCP channel.

Subscriber Command Format -- The subscriber command consists of a payload marker, payload length, command code and actual payload bytes. To easily separate multiple commands on the wire, the first four bytes in the command wire format is a payload marker, specifically: 0xAA, 0xBB, 0xCC, 0xDD. The next four bytes (a 32-bit integer) represent the total size of the payload including the command code. The next byte is the subscriber command code.

Publisher Response Format -- The publisher response consists of a response code, an in-response-to command code, payload length and actual payload bytes. The first byte is the response code. The second byte is command code for which this response applies. It is required even if the response is unsolicited. The next four bytes (a 32-bit integer) represent the payload length. The actual bytes of the payload follow, if any, and are specific to the actual response. For example, in the case of a data packet response, the payload will contain serialized measurements.

The serialized measurements, "the data", are constructed in GEP to be easy to parse so that third-party systems can easily consume and use data. The data structure is a repeated binary encoding of an ID, timestamp, measured value and flags (e.g., time and data quality).

- Point ID – A 128-bit GUID identifier provided in the command channel. This value is compressed to 2 bytes on the wire using run-time cache.
- Time – A 64-bit integer based timestamp in "ticks" (100's of nanoseconds). This value is also compressed to 2 bytes on the wire using an offset to an absolute time value which is updated frequently.
- Value – Typically a 32-bit floating point real number (4 bytes). GEP supports other data payload sizes.
- Flags – For synchrophasor data these flags are embedded in a 16-bit integer representing the IEEE C37.118 data quality flags.

GEP and Grid Solutions Framework Improves Security

GEP can be implemented with or without its security features. GEP enables implementation of both strong access control and encryption. For GPA's products, security is managed through components in the Grid Solutions Framework[iv] of which GEP is a part. These features include:

- Administrator access control where multiple role-based options are available. This access control can be implemented to integrate with existing enterprise authentication (e.g., Microsoft AD, Kerberos, or local) as well as existing enterprise authorization (e.g., Microsoft AD, RBAC). The GSF also provides the capability for multi-factor authentication strategies using hardware/software tokens and/or biometrics.
- Authentication / access control for data communication includes strong authentication of trusted appliances through the out of band exchange of symmetric keys. Publishers have a fine-grained mechanism to control access to specific data by authenticated partner (or trusted) GEP appliances.
- Integrity-protected logging for operating logs and configuration logs as well as remote log storage capability for additional security. The GSF leverages standardized logging to the OS so that errors and events can be captured through enterprise log integration systems.
- Key Management – The GSF is configurable to allows use and manage private keys in a highly isolated environment. The GSF is capable of utilizing key management services that offer X.509 identity certificates for authentication. In the absence of that infrastructure, GSF is able to use self-signed X.509 identity certificates that are securely communicated out-of-band. The GSF also includes NIST-approved cryptographic algorithms and parameters (e.g., key length, hash size) for tenability
- Failure Management – The GSF is able to recover from basic failure and includes automatic restarting of the service on failure. In addition there is service and application visibility to external systems is available to support monitoring and early warning of failure.

The GSF facilitates full compliance with NERC CIP standards with complete logging of configuration changes and administrative actions.

As seen in Table 1 below, GEP is an alternative to the use of VPN for securing communication of streaming utility operating data.

VPN Approach	GEP Approach
• Security managed at a network interface level	• Security managed at the application layer, with fully flexible pairwise security
• Traffic only protected once it reaches the VPN tunnel, susceptible at every previous level	• Traffic is protected from the very beginning, protecting it directly in the application which eliminates exposure via other apps on the system
• VPN failure can result in either unencrypted data being sent, or complete blockage of transmission until the network issue can be resolved	• Connection failure results in retried connections, renegotiating the key at each try
• If network issues are present, the connection may require intervention to either start or stop once the network issues are corrected	• If network issues are present, the connection will be re-established without intervention once they are corrected

Table 1. Comparison of VPN and GEP

GEP API's are available[v] in C/C++, .NET (including Mono and Unity 3D) and Java to enable these applications to easily be integrated with the applications that use GEP and avoid limitations imposed by use of frame-based protocols. An open source application, the GEP Subscription Tester[vi] is a multi-platform graphical application that can be used to verify connectivity to applications implementing a GEP data publisher.

V. OSS Products Derived from GSF

The Grid Protection Alliance has developed a suite of open source products built from the Grid Solutions Framework and which leverage the GEP for secure transport.

SIEGate

SIEGate[vii] is a security-centric appliance designed from the ground up to reliably exchange the information necessary to support real-time control room operations. As seen in Figure 2 below, using GEP SIEGate can exchange measurement data, alarm and notification data as well as batch or file-based data at low latency. As compared to current utility practices, SIEGate significantly improves security while reducing the administrative burden and cost to exchange grid data among control rooms. Additional SIEGate features are:

- High availability and reliability
- Can bridge multiple namespaces and indifferent to registry use
- Easy publication and subscription configuration
- Rapid extensibility to support new protocols
- Detect and alarm on communication or data

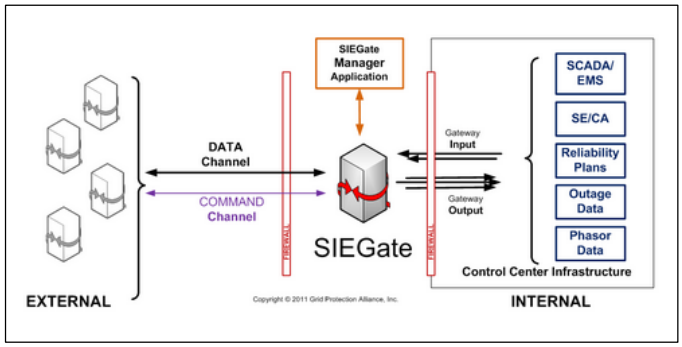

Figure 2. SIEGate

For use with phasor data, SIEGate accepts all frame-based synchrophasor protocols. There include: IEEE C37.118 R1 & R2 (up to 120 samples per second), IEEE 1344. BPA Stream, Macrodyne G and M, SEL Fast Message, 61850-90-5 (PG&E Implementation).

substationSBG

The substationSBG leverages the functionality provided by GEP and the features of GPA's openPDC and SIEGate to form a purpose-built, high-availability data gateway for use in substations. For phasor data, it is both a substation PDC with a local data historian and a gateway to enable the secure, reliable communication of synchrophasor data from the substation to the control center. It has been tested on fan-less substation computers for both 32 and 64 bit processors using either Windows or Linux operating systems.

The benefits of the substationSBG include:

- Provides security isolation between trusted internal systems and untrusted external ones
- Presents a small external attack surface

- Creates highly trusted gateway-to-gateway associations
- Makes it easy to publish or subscribe to data
- Overcomes scaling and latency limitations with frame-based protocols
- Minimizes communication bandwidth

As shown in Figure 3 below, phasor data that moves through the substationSBG is persisted locally in a short-term rolling archive. Following any communications outage between the substation and control center, data archived locally by the substationSBG is transmitted (at lower priority than real-time data) back to the control central to ensure that the central archive-of-record is complete.

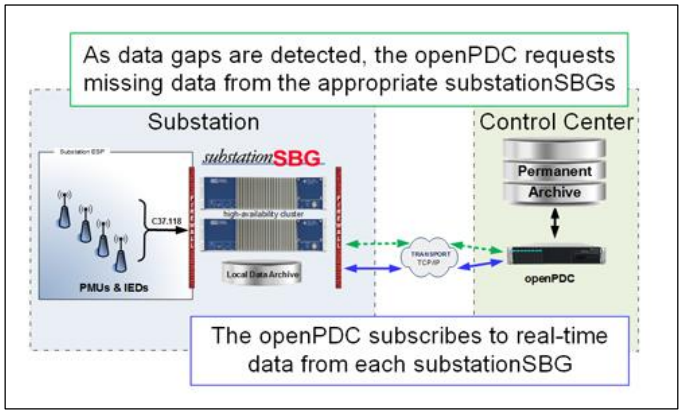

Figure 3. substationSBG

Phasor Data Quality Tracker

PDQTracker [viii] is a high-performance, real-time data processing engine designed to raise alarms, track states, store statistics, and generate reports on both the availability and accuracy of streaming synchrophasor data. PDQ Tracker will work with any vendor's PDC and synchrophasor data infrastructure; measure and automatically produce periodic reports on phasor data completeness and correctness; alarm and/or create emails in real-time if data quality problems are detected.

The purpose of PDQ Tracker is (1) to measure phasor data quality, (2) to disseminate data quality information to assure data quality awareness and facilitate data quality problem resolution, and (3) to provide a platform that can be extended to provide comprehensive data quality analytics including phasor data correction

ARMORE

ARMORE is a substation appliance being developed by the Grid Protection Alliance and the University of Illinois[ix] that will perform inspection of network packets and alarm on communications that does not comply with the specified ARMORE policy. ARMORE will be capable of blocking traffic based on deep inspection of common substation communications protocols, such as DNP3. ARMORE will also be capable of encapsulating and encrypting legacy communications and resiliently exchanging this information among ARMORE nodes.

Bro, an open source network analysis platform, is the core engine to be used by ARMORE to inspect network packets. Bro conducts semantic analysis of network traffic in process control and other networks. With Bro integrated within it, ARMORE will provide the ability to collect statistics, inspect relevant traffic, and apply policies to that traffic to help secure critical infrastructure from attackers all while securely communicating both known and unknown protocols to their intended destination via a fault tolerant middle-ware.

VI. Conclusions

A growing number of effective OSS solutions designed to for phasor data transport, security, and resiliency are now available to facilitate applications that range from monitoring data traffic at the substation device level all the way to real-time data exchange at the operations center level. These solutions have been developed in collaboration with major US utilities, and some have been deployed around the world. Building on this established base of phasor data tools, and leveraging the OSS environment to foster continued innovation and collaboration, phasor data systems can continue to support the operation and modernization of the electric power grid.

[i] Synchrophasor Applications in Transmission Systems https://www.smartgrid.gov/recovery_act/program_impacts/applications_synchrophasor_technology

[ii] 2014 Future of Open Source http://www.slideshare.net/blackducksoftware/2014-future-of-open-source-survey-results

[iii] NASPI software exchange https://www.naspi.org/synchrophasorsoftware

[iv] GSF http://gsf.codeplex.com/

[v] GEP API http://gsf.codeplex.com/wikipage?title=Subscriber%20API%20%28C%2b%2b%29&referringTitle=Documentation

[vi] The GEP Subscription Tester. http://openpdc.codeplex.com/wikipage?title=GEP Subscription Tester

[vii] SIEGate https://www.controlsystemsroadmap.net/ieRoadmap%20Documents/SIEGate-011613.pdf

[viii] PDQ Tracker http://gridprotectionalliance.org/docs/products/PDQTracker/highlevelrequirements.pdf

[ix] ARMORE http://gridprotectionalliance.org/products.asp#ARMORE