Analyzing NASPInet Data Flows

Ragib Hasan, Rakesh Bobba and Himanshu Khurana
University of Illinois at Urbana Champaign

***Abstract*—One of the missions of the North American SynchroPhasor Initiative (NASPI) is to create a robust, widely available and secure synchronized data measurement infrastructure, dubbed NASPInet, that will improve reliability of the power grid. Phasor Measurement Unit (PMU), a GPS clock synchronized measurement device capable of measuring the current and voltage phasors in the power grid, is the main measurement device that NASPInet envisions to support. While the dataflow, latency and to some extent security requirements for individual PMU applications that depend on the measurement infrastructure have been characterized, this work undertakes the challenge of characterizing the collective dataflow, latency and security requirements of the measurement infrastructure when using different network architectures and when multiple PMU applications simultaneously utilize NASPInet. For our analysis we focus on a case study where we model a scalable scenario in NASPInet for a part of the North American Power Grid, the western interconnect, using Network Simulator v2 (NS-2).**

***Index Terms*—PMU SynchroPhasor NASPInet**

I. Introduction

THE North American electric power grid is a highly interconnected system hailed as one of the greatest engineering feats of the 20th century. However, increasing demand for electricity and an aging infrastructure are putting increasing pressure on the reliability and safety of the grid as witnessed in recent blackouts [1], [2]. Furthermore, deregulation of the power industry has moved it away from vertically integrated centralized operations to coordinated decentralized operations. Reliability Coordinators (RCs) such as Independent System Operators (ISOs) or Regional Transmission Operators (RTOs) are tasked by Federal Energy Regulation Commission (FERC) and North American Electric Reliability Council (NERC) with overseeing reliable operation of the grid and providing reliability coordination and oversight over a wide area. Balancing Authorities (BAs) are tasked with balancing load, generation and scheduled interchange in real-time in a given Balancing Authority Area (BAA). BAA is a geographic area where a single entity balances generation and loads in real-time to maintain reliable operation. BAAs are the primary operational entities that are subject to NERC regulatory standards for reliability. Every generator, transmission facility, and end-use customer is in a BAA.

In order to improve the reliability of the power grid while meeting the increased power demand, the industry is moving towards wide-area measurement, monitoring and control. The Department of Energy (DOE), NERC and electric utility companies formed the North American SynchroPhasor Initiative (NASPI) (www.naspi.org) with a vision to improve the reliability of the power grid through wide area measurement, monitoring and control. NASPI's mission is to create a robust, widely available and secure synchronized data measurement infrastructure with associated monitoring and analysis tools for better planning and reliable operation of the power grid. NASPI envisions deployment of as many as tens to hundreds of thousands of Phasor Measurement Units (PMUs) across the grid that pump data at 30 samples/second to hundreds of applications in approximately 140 BAAs across the country. PMUs are clock synchronized (through GPS) sensors that can read current and voltage phasors at a substation bus on the transmission power network. The delivery of sensed PMU data to applications is envisioned to be achieved via a distributed digital communication network called NASPInet. The distributed and open nature of NASPInet makes it necessary to provide adequate security for the PMU data traversing it; e.g., to prevent unauthorized modifications [3].

Research and development efforts are underway that aim to develop accurate PMU data sensing, NASPInet and storage systems that deliver data from PMUs to applications, and novel applications that utilize PMU data. However, to a large extent these efforts focus on individual PMU applications utilizing data from a few selected PMUs spread over a small geographic area. There are good reasons for doing so as that in itself poses significant challenges. In contrast, we focus on studying the data flow, latency and security properties for the communication and storage systems at scale; e.g., what kind of bandwidth is needed when tens of thousands of PMU generate data that is consumed by hundreds of applications? Via simulation we study how different networking and storage architectures as well as security mechanisms can scale to adapt to the needs of a large-scale PMU system. We use the Network Simulator v2 (NS-2) whereby simulation models are developed and run with varying parameters to allow us to answer the following specific questions among a set of general data flow and latency related ones:

1) What are the bandwidth requirements on links between PMUs and entities on NASPInet such Phasor Gateways (PGWs) and Phasor Data Concentrators (PDCs)?
2) What latencies can be supported on dedicated point-to-point communication links over short and long distances?
3) How do various security mechanisms affect the bandwidth requirements and latency guarantees computed above?
4) What are the storage requirements at PDCs?

To address these questions we simulate a case study that

The authors are with the Computer Science Department, National Center for Supercomputing Applications and Information Trust Institue respectively at the University of Illinois at Urbana-Champaign. E-mail:{rhasan,rbobba,hkhurana}@illinois.edu

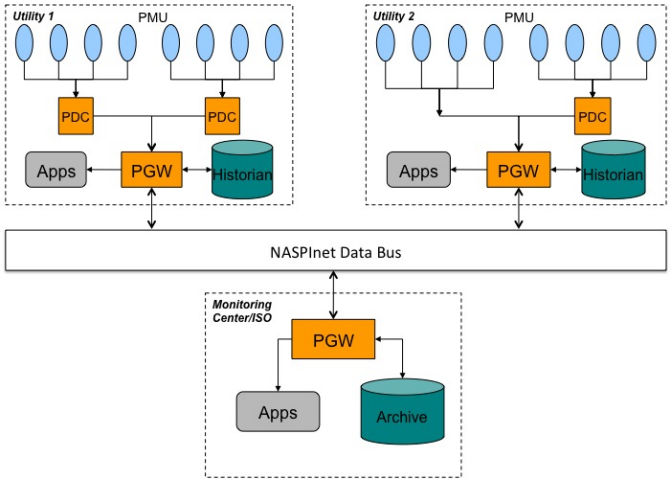

Fig. 1. Proposed NASPInet Architecture

represents a scalable scenario in the western interconnect characterized by 1) large number of PMUs, 2) extensive PMU data sharing and 3) point-to-point communication links.

The rest of this work is organized as follows. In Section 2 we provide background on PMU data sharing and NASPInet application classes. In Section 3 we describe our simulation framework. In Section 4 we describe the simulation case study and we conclude in Section 5.

II. Background

Currently, sensor readings from transmission substations are sent via Supervisory Control And Data Acquisition (SCADA) systems in the local BA that controls the system and to the regional RC that oversees reliable operation of the system. There are operations taking place at various time granularities to keep the power system stable and reliable. Among the frequent operations, protection and control mechanisms at substation operate at the granularity of milliseconds, state estimators and contingency analysis in BAs and RCs operate at the granularity of minutes and hourly and day ahead power markets run by RCs operate at the granularity of hour and day respectively.

PMUs can provide direct access to the state of the grid at higher frequencies than current approaches of having to estimate the state of the grid every two minutes or so. In addition, there are a range of emerging applications that collectively demonstrate a strong benefit of sharing PMU data widely. Sharing PMU data widely will help in operating the grid safely and reliably and in avoiding overloading, outages, brown-outs and blackouts [1], [2]. Sharing PMU data will also help in planning, post disturbance/event analysis [2] and for research and development purposes. Applications envisioned to utilize PMU data are classified into four classes based on their data requirements as shown in Table I[1]. Typically feedback control applications like transient stability control fall into Class A, open loop control applications like state estimation fall into Class B, post event analysis applications like disturbance analysis fall into Class C and visualization and monitoring applications like situational awareness fall into Class D [4], [5]. Currently two pilot deployments each with about 75 PMUs exist in Eastern [6] and Western [7] Interconnects.

There is need for a framework that provides for scalable, secure, robust and flexible data sharing before a wide area full scale deployment of PMUs can be realized [4]. NASPI envisions NASPInet to be this flexible framework. Figure 1 shows a high-level architecture for the framework as envisioned in NASPI's RFP [5]. PMU's send their data either to a Phasor Data Concentrator (PDC) at a substation or at the local BA or the Phasor Gateway (PGW) at the local BA. Data received at the PDC is time aligned before it is forwarded to local applications, archives and the PGW. PGWs are to mediate all access to the NASPInet Data Bus, that is, they are to mediate data sharing between utilities. They are envisioned to provide authentication, access control, QoS service based on service classes and security for all the data and control traffic through them. The data bus is envisioned to route and deliver data while meeting the quality of service, latency, throughput and security requirements of the data. Efforts to design NASPInet are underway [8].

TABLE I
PMU Application Classes

	Class A	Class B	Class C	Class D
Low Latency	Critical	Fairly Important	Not Very Important	Somewhat Important
Reliability/ Availability	Critical	Somewhat Important	Fairly Important	Not Very Important
Data Accuracy	Critical	Somewhat Important	Critical	Not Very Important
Time Alignment	Critical	Critical	Not Very Important	Somewhat Important
Message Rate	Critical	Somewhat Important	Critical	Somewhat Important
Sample Application	Out of step protection	State Estimation	Disturbance Analysis	Real Time Compliance Monitoring

III. Framework

In this work we argue that there is a need to study scalability aspects on NASPInet. To allow for an extensive design space exploration, we chose to use a simulation based framework for the study. We implemented our framework using the Network Simulator ns-2 [9]. ns-2 is a discrete event simulator widely used for simulating networked systems. Written in C++ and OTcl, ns-2 is open source, extensible, and allows addition of custom-designed packet headers, protocols, and applications.

Model: For simulating the NASPInet in ns-2, we designed the following components:

- **UDP Agent for PMU Data:** We added a header for PMU data, which tracks the source of the PMU data and a sequence number for the source. This header helps us track end-to-end latency experienced by the PMU data. We then modified the existing UDP Agent, that models the UDP protocol in ns-2, to accept this header data from the application that is using this agent to transmit PMU data and return this header to the application that is using

[1]Table I is based on NAPSI's Phasor Application Classification and Taxonomy http://www.naspi.org/resources/dnmtt/dnmttresources.stm

the agent to receive PMU data. This allows us to capture complete data flows.

- **PMU Application:** This component simulates a PMU device and generates phasor readings along with the necessary PMU header. We modified ns-2's constant bit rate (CBR) traffic generator to simulate data generation in a PMU. Each component generates PMU data packets at a configurable rate and sends them to the PDC for that PMU after invoking configured security operations.
- **PDC Application:** This component simulates a PDC. It receives the packets from the PMU applications, time-aligns these packets and then forwards the packets to the PGW after integrity protecting them if configured to do so. Time alignment is modeled by having the PDC wait for PMU data with the same sequence number from all the PMUs it is connected to. Necessary security operations are performed as configured.
- **PGW Application:** This component collects the time-aligned packets sent by the local PDC applications and forwards them to other PGWs, with which it is sharing data with after performing necessary security operations. It also receives PMU data from other PMUs that are sharing data with it. Again, necessary security operations are performed as configured.

In all of the application components above, *i.e.*, PMU, PDC and PGW, security operations are modeled by adding appropriate computation delay and byte overhead to each packet. For example, to model integrity protection using a Message Authentication Code (MAC) like HMAC-SHA1 a 20 byte overhead is added to each packet and a 1μs computation delay is added at both the sender and receiver of the packet.

The components were written as C++ classes. Each component has one or more agents that handle the transfer of data packets to other components. In addition, we enhanced the tracing capability of the simulator to trace the PMU header as the packet crosses multiple nodes to be able to track the end-to-end latencies of PMU data. We wrote necessary scripts to analyze the trace file and obtain end-to-end latencies, bandwidth usage, etc.

We designed our components and simulation scripts to be highly configurable. All the components can be configured to simulate any cryptographic processing overhead or latency for common security mechanisms. This allows us to test end-to-end or hop-by-hop security models with little effort. Simulation scripts can be configured to tune the bandwidth and latencies in network links to match a simulation scenario. The topology for the NASPInet is loaded from a configuration file where different simulation parameters can be set. Visualization of a simulation is provided by Network Animator (*nam*) a visualization tool available for ns-2 [9].

IV. Case Study: Western Interconnect

Our simulation framework is capable of studying a range of scalability issues that will arise in NASPInet as it grows. This includes adequate bandwidth provisioning to ensure data delivery without packet loss as well as ensuring latency guarantees for various application classes. And it also includes estimating data generation, storage and processing capabilities to support data sharing by applications based on where the applications are located and when they need the data. We conducted one such case study where we assume strong growth of PMUs and corresponding applications over a period of time. Focusing on the western interconnect, the scenario we simulate has the following salient characteristics:

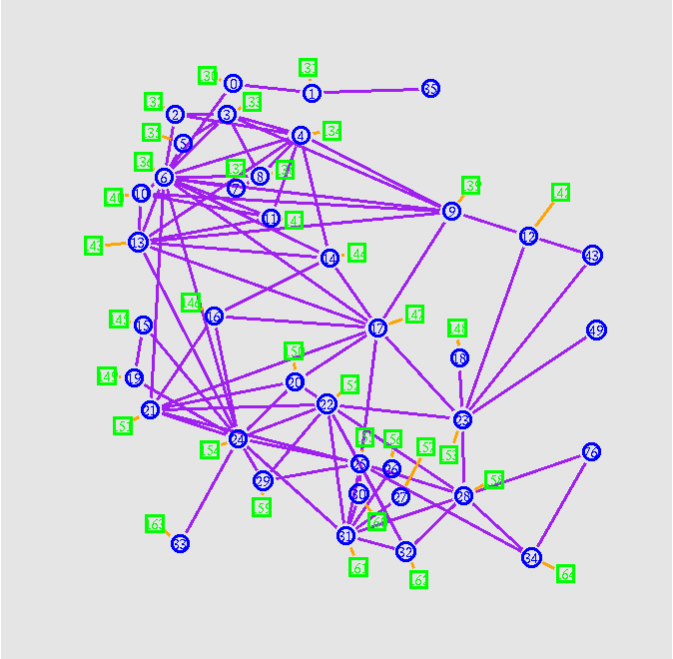

Fig. 2. Western Interconnect Topology as shown in Network Animator (*nam*). LEGEND: Circles (in Blue) represent PGWs. Squares (in Green) represent PDCs. Purple and Orange lines connecting them represent communication links.

- *Large number of PMUs*. We estimate that there are 35 BAs and western interconnect. We varied the number of PMUs in each BA from 150 to 250 giving rise to 5250 to 8750 PMUs in the interconnection. For each BA we assume one logical PDC device and one logical PGW devices that supports data distribution.
- *Extensive data sharing*. We assume that each BA shares data with all other BAs with which it shares a tie line. While this need not necessarily capture all data sharing needs in the future we believe this to be an effective metric for our study.
- *Point-to-point communication links*. In our envisioned scenario we imagine that this growth of PMUs and data sharing is organic in that when and where needed communication links are setup to enable data sharing. Therefore, we simulated point-to-point communication links across the interconnect. Clearly, more efficient network architectures such as multicast ones might be more appropriate and we believe that simulation studies like ours will truly motivate the need for such advanced network architectures and help quantify their requirements.

A. Setup

In this case study we analyzed the bandwidth and data storage requirements for PMU data sharing and the end-to-end latency guarantees that can be met in the above Western Interconnect scenario both with and without security. Specifically, we had 35 PGWs representing the 35 BAs in the Western Interconnect. Each BA shared its PMU data with only those BAs with whom it shares a tie line. The number of PMUs in each BA is varied from 150 to 250. All the PMUs in a BA sent their data to a single PDC that time aligned it and forwarded it to the local PGW to be shared with other BAs. PMUs communicated with their local PDC using dedicated point-to-point leased lines. PDCs communicated with their local PGW over a gigabit link intended to model Ethernet. PGWs shared PMU data with other BAs, here represented by PGWs, over independent point-to-point leased lines. Figure 2 shows the topology of the resulting PGW network as seen in NAM. The circles represent the PGWs and the squares represent the PDCs. PMUs are omitted in the figure for clarity.

Communication links are characterized by their capacity, *i.e.*, the number of bits they can transmit per second, and their delay, *i.e.*, the time it takes for a bit to travel from the source to the destination after it is transmitted. The bandwidth of dedicated communication links between a PMU and a PDC is set based on PMU sample size and the sampling rate and any security overhead. Since we intend to model leased lines which may traverse multiple physical communication links underneath, the delay on them includes the propagation delays as well as delays added by intermittent nodes such as switches. This delay on the Internet today is typically within a factor of two of the propagation delay. Thus if the leased line is between two entities 1000 miles apart the delay will be less than $16ms$[2]. For our study, the delay on the leased lines between PMUs and PDCs is picked using a normal distribution with a mean of $4ms$ corresponding to a mean distance of 250 miles and standard deviation of $0.5ms$.

For our study each PMU generated data at a rate of 30 samples per second with each sample size being 128 bytes, which is approximately the rate required to transmit a bus voltage and two line currents in the 16-Bit integer format using IEEE PC37.118 data format [10] as described in [11]. This data is transmitted to the local PDC using the connectionless UDP protocol to minimize overhead. The PDC waits for the data sample for a given instant, here modeled by a sequence number, from all the PMUs it is connected to and forwards the bundled time-aligned data to the local PGW over a gigabit link characterized with a $1ms$ delay. The PGW then forwards this data to PGWs in all the BAs that it is sharing data with using UDP protocol over point-to-point dedicated lines. The bandwidth of communication links between PGWs is set based on the number of PMUs in the BA. It is important to note that the delay on these lines is set based on the distance between the BAs in the Western Interconnect. Thus the delay on the line between PGW in Los Angeles Department of Water and Power (LDWP) and the PGW in Bonneville Power Administration (BPAT) which are about 1000 miles apart is set to be $16ms$. The simulations were run on a machine running Ubuntu Linux on a Intel Dual Core Xeon $2.5Ghz$ processor with $12GB$ of RAM.

[2]Assuming the speed of light in the carrier is 66% the speed of light in vacuum

B. Results and Analysis

TABLE II
BANDWIDTH REQUIREMENTS ON COMMUNICATION LINKS BETWEEN PGWS

Number of PMUs	Bandwidth needed	T1 lines needed
100	$3.4Mbps$	3
150	$5.1Mbps$	4
200	$6.8Mbps$	5
250	$8.5Mbps$	6

Bandwidth The links between PMUs and PDC need to have at least $34.7Kbps$ bandwidth without authentication and at least $39.4Kbps$ bandwidth when using a 20 byte MAC for authentication. Thus we set the bandwidth on links between PMUs and PDCs to be $56Kbps$ which is next commonly available modem speed that can accommodate the above data rates. Table II shows the bandwidth needed (inlcuding UDP/IP overhead) on communication links connecting PGWs as the number of PMUs in a BA is increased from 150 to 250. The last column in the table shows the number of T1 lines ($1.544Mbps$) that a BA has to provision to meet the need. The total inter-PGW bandwidth (simplex) needed in western interconnect when there are 200 PMUs per BA when using point-to-point links is $1.3Gbps$, *i.e.*, an average of $38Mbps$ per BA.

End-to-end Latency To illustrate end-to-end latency in the interconnect we picked communication links spanning typical distances in the interconnect. Specifically, we picked 300 mile link to represent all the links spanning 251 to 350 miles which constituted about 20% of the total links. Similarly we picked links spanning 400, 500 and 600 miles to represent ranges each of which constituted more than 10% of the links. We picked 50 mile link as it is the shortest length link we used and it constituted about 16% of the total links. We also picked the 1000 mile link between PGWs at LDWP and BPAT as it is the longest in the interconnect.

TABLE III
AVERAGE END-TO-END LATENCY WHEN DATA IS TIME-ALIGNED AT SOURCE BA (VARIANCE ~ 0)

Link Distance	End-to-End Latency	End-to-End Latency with Auth.
50	$56.4ms$	$59.2ms$
300	$61.1ms$	$64.1ms$
400	$62ms$	$64.9ms$
500	$63.6ms$	$66.5ms$
600	$65.9ms$	$68.9ms$
1000	$72ms$	$74.9ms$

Table III shows the average end-to-end latency of PMU data both with and without authentication when PMUs per PGW is set to 200, link bandwidth between PMUs and PDC is set to $56Kbps$ and link bandwidth between PGWs is set to the equivalent of 5 T1 lines as seen in Table III. That is, it shows the average delay between when a PMU sample in a source BA

for a given time[3] is generated and when the time-aligned PMU data for that time is received at the PGW in the destination BA. When authentication is simulated 20 byte overhead and a $1\mu s$ computation delay are added for each packet at PMU, PDC and PGW applications respectively to model HMAC-SHA1 authentication scheme. The average latency for PMU data sent varied between $56.4ms$ and $72ms$ without authentication and between $59ms$ and $74.6ms$ with authentication. Since the links are dedicated and adequately provisioned in terms of bandwidth the deviation from average is almost negligible. Note that authentication only added a delay of about $3ms$ which it turns out is essentially the time needed to transmit the extra 20 bytes from PMUs to PDCs. Computation delays are negligible when compared to communication costs and the time to transmit the extra 20 bytes on inter-PGW links is also negligible.These latencies meet the requirements of even Class A applications some of which can only tolerate latencies in the order of $100ms$. These results indicate that when the PMU data is time-aligned before it is shared with other BAs it may be possible to use public-key based digital signatures at a PGW instead of MACs to integrity protect data with out increasing the latency by more than a few milliseconds. For example, an RSA signature takes only about $2ms$ to compute and the additional time needed to transmit the signature, 128 bytes, will be negligible on inter PGW links.

Note that the bandwidth provisioned on inter-PGW links in the above case, $7.72Mbps$ ($\sim$ 5 T1 lines), is more than that is needed, $6.8Mbps$ as seen in Table III. In order to see how using shared links affects the latency we sent additional non PMU traffic on inter-PGW links. We used Pareto on/off distribution available in ns-2 to generate enough self similar traffic to use the additional $0.9Mbps$ bandwidth available on the link. The pareto parameters we used are, pareto shape parameter of 1, packet size of 1500, equal burst and idle times (5 seconds each for a total simulation time of 10 seconds) and a rate of 1000000 to achieve total bandwidth usage of $7.68Mbps$ on average with a standard deviation of $0.098Mbps$. Table IV shows the average end-to-end latency obtained in this case. While the average end-to-end latency only increased by $6ms$, standard deviation is no longer negligible but is $3.2ms$. The second column in Table IV shows the minimum and maximum latencies observed and the maximum latency observed is almost $12ms$ more than that observed when using dedicated link. This indicates that when using the inter-PGW links for additional traffic, that traffic should be carefully characterized to bound PMU latencies.

TABLE IV
AVERAGE END-TO-END LATENCY WHEN USING SHARED LINK

Link Distance	End-to-End Latency (Avg. /Std. Dev.	End-to-End Latency (min. /max.)
50	$62.3/3.2ms$	$56.8/68.5ms$
300	$67/3.2ms$	$61.1/73.2ms$
400	$67.9/3.2ms$	$61.9/74.1ms$
500	$69.5.2/3.2ms$	$63.7/75.7ms$
600	$71.8/3.2ms$	$65.9/78ms$
1000	$77.9.2/3.2ms$	$72/84.1ms$

[3]For a given sampling rate, PMUs generate data samples evenly spaced through each second with the first sample coinciding with the UTC second roll over.

Table V shows the average end-to-end latency of PMU data both with and without authentication when it is not time aligned at the source BA but is time-aligned at the destination BA. That is, it shows the average delay between when a PMU sample in a source BA for a given time[3] is generated and when the sample from the last PMU for that time is received at the PGW in the destination BA. Time aligning data at destination improves end-to-end latency by about $28ms$ both with and without authentication. Thus time aligning at destination seems a better option when point-to-point communication architecture is used for connecting PGWs and MACs are used for authentication. However using time-alignment at destination might become very expensive in terms of bandwidth if public-key based cryptographic primitives need to be used.

TABLE V
AVERAGE END-TO-END LATENCY WHEN DATA IS TIME-ALIGNED AT DESTINATION BA (VARIANCE ~ 0)

Link Distance	End-to-End Latency	End-to-End Latency with Auth.
50	$28.7ms$	$32ms$
300	$33.6ms$	$36.8ms$
400	34.8	$38.1ms$
500	$35.8ms$	$39.1ms$
600	$38.7ms$	$41.8ms$
1000	$44ms$	$47.2ms$

Storage PMUs generate tremendous amount of data everyday. With 200 PMUs generating 30 samples a second a BA would have 768000 bytes of data generated every second. While some of this data is header information that need not be stored it is indicative of the amount of data a BA has to manage. Even storing just one year's worth of locally generated data would mean storing $22TB$. If data received from other BAs is also locally stored then the storage requirements would increase 5 times on an average in western interconnect. At some BAs like BPAT the storage will increase 15 fold as they are highly connected. Thus good data compression techniques and storage strategies need to be designed and developed to manage this data. One strategy might be to store only locally generated data but make it available to other BAs in a secure manner when needed.

V. RELATED WORK

Tomsovic *et. al.* [12] discuss the need for a real-time wide area communication network for large power systems and present Gridstat [13] a QoS managed publish/subscribe overlay network as a possible solution. Johnston *et. al.* [14] presents Gridstat as means to disseminate PMU data. While Gridstat is shown to add only a $0.1ms$ latency per hop over that of underlying network the latency and scalability characteristics of the underlying network itself have not been analyzed. Existing deployments of PMUs in Eastern [6] and Western [7] Interconnects are small scale and centralized. Efforts to produce a specification for NASPInet are underway [8]

VI. CONCLUSION AND FUTURE WORK

In this work we have designed and demonstrated a flexible framework that can be used to analyze the scalability of NASPInet. In the future, we will use this framework to analyze

1) how multi-hop network architectures like multicast affect the guarantess that NASPInet can provide, 2) how connection based protocols like TCP affect the guarantess that NASPInet can provide and 3) how NASPInet for the full north american grid scales.

ACKNOWLEDGMENT

The authors would like to thank the entire Trustworthy Cyber Infrastructure for Power Grid project team for valuable discussions. Part of the second and third authors' work is supported by National Science Foundation under Grant No. CNS-0524695. First author's work and part of second and third authors' work is supported by Office of Naval Research under Grant No. N00014-07-1-1173. Any opinions, ndings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reect the views of the National Science Foundation or the Office of Naval Research.

REFERENCES

[1] U.S.-Canada Power System Outage Task Force, "Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations," April 2004.

[2] J. Dagle, "Postmortem analysis of power grid blackouts - the role of measurement systems," *Power and Energy Magazine, IEEE*, vol. 4, no. 5, pp. 30–35, Sept.-Oct. 2006.

[3] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *IFIP WG 11.10 International Conference on Critical Infrastructure Protection*. Springer, March 2008.

[4] J. E. Dagle, "North american synchrophasor initiative," in *Hawaii International Conference on System Sciences*, 2008.

[5] "DNMTT Resources http://www.naspi.org/resources/dnmtt/dnmttresources.stm."

[6] M. Donnelly, M. Ingram, and J. R. Carroll, "Eastern interconnection phasor project," in *Hawaii International International Conference on Systems Science (HICSS-39 2006)*, January 2006.

[7] J. Cai, Z. Huang, J. Hauer, and K. Martin, "Current status and experience of wams implementation in north america," *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, pp. 1–7, 2005.

[8] "Statement of Work - Specification for North American Synchrophasor Initiative (NASPI)," http://www.naspi.org/resources/dnmtt/quanta_sow.pdf, May 2008.

[9] K. Fall, K. Varadhan *et al.*, "The ns Manual," Online at http://www.isi.edu/nsnam/ns/doc/index.html, 2002.

[10] "IEEE Standard for Synchrophasors for Power Systems," *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*, pp. 1–57, 2006.

[11] R. Moxley, "Synchrophasors in the real world," White Paper, http://www.selinc.com/techpprs/TP6194_Synchrophasors_Moxley_20060118.pdf, January 2006.

[12] K. Tomsovic, D. Bakken, V. Venkatasubramanian, and A. Bose, "Designing the next generation of real-time control, communication, and computations for large power systems," *Proceedings of the IEEE*, vol. 93, no. 5, pp. 965–979, May 2005.

[13] K. Gjermundrød, "Flexible qos-managed status dissemination middleware framework for the electric power grid flexible qos-managed status dissemination middleware framework for the electric power grid," Ph.D. dissertation, Washington State University, 2006.

[14] R. A. Johnston, C. H. Hauser, K. H. Gjermundrød, and D. E. Bakken, "Distributing time-synchronous phasor measurement data using the gridstat communication infrastructure," in *Hawaii International International Conference on Systems Science (HICSS-39 2006)*, January 2006.

Ragib Hasan is currently a PhD candidate at the Department of Computer Science of the University of Illinois at Urbana-Champaign. He received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 2003, and M.S. in Computer Science from the University of Illinois at Urbana-Champaign in 2005. His research interests include Secure Provenance, Remembrance-capable systems, Storage System security, and Computer-supported collaboration.

Rakesh Bobba is a Security Engineer at National Center for Supercomputing Applications (NCSA), University of Illinois, Urbana-Champaign. He received his B.S. from Birla Institute of Technology and Science (BITS), Pilani, India (2000) and M.S. from the University of Maryland (2007). His research interests are in network and distributed system security including access control, key management, applied cryptography among others. He has been part of a number of research and development activities and is currently involved with design and development of secure communication infrastructures for the next generation Power Grid as part of NSF funded Trustworthy Cyber Infrastructure for the Power Grid Center (TCIP).

Himanshu Khurana is a Principal Research Scientist at the Information Trust Institute, University of Illinois at Urbana-Champaign and will serve as the Principal Investigator for this project. He received his Ph.D. from the University of Maryland, College Park in 2002. His research interests are in distributed system security, and he has published over 20 papers in this field. Currently, he leads projects dealing with security and trust for email systems, next-generation messaging, and control systems. His work on secure email list services received an honorary mention in ComputerWorld Magazine 2005 Horizon Awards. He is currently serving as Principal Scientist for the Trustworthy CyberInfrastructure for the Power Grid Center (TCIP).