Comparison of Publish-Subscribe Network Architectures for Smart Grid Wide Area Monitoring

Kiran Kumar[1], Muthukumar Radhakrishnan[1], Krishna M. Sivalingam[1], Deva P. Seetharam[2] and M. Karthick[1]
[1]Department of CSE, IIT Madras, Chennai 600036, India
[2]IBM Research India Labs, Bangalore 560045, India
Email: *cheriviralakiran@gmail.com, rmuthu@cse.iitm.ac.in, skrishnam@cse.iitm.ac.in, dseetharam@in.ibm.com*

***Abstract*—The future of the electrical power grid is the "Smart Grid" that uses a combination of digital sensing and actuation technologies. For example, Phasor Measurement Units (PMU) can be used to accurately measure the quality of power and help improve the reliability and efficiency of the grid. A specially designed network architecture is required to transport the PMU data to the data processing systems. The NASPInet framework specifies requirements (such as latency and reliability) to be met for different types of applications that will process the PMU data. The objective of this paper is to analyze suitable network architectures that can meet all the requirements of NASPInet and thus be useful for the Smart Grid. An architecture based on the publish-subscribe paradigm and a distributed hash table technique for storing meta-data about the publishers is presented. This architecture is compared to GridStat, PSIRP and Source Specific Multicast approaches, using discrete event simulation models based on OMNET++. The delay performance of these systems has been studied for different network parameters.**

I. INTRODUCTION

The modernization of the electrical power grid along many dimensions using several different technologies is collectively referred to as the *Smart Grid* [1], [2]. One of the important capabilities of the *Smart Grid* is the ability to operate reliably and recover without manual interventions as much as possible. This is possible using *Phasor Measurement Units (PMU)* that can provide precise measurements known as synchrophasors or time-synchronized phasors [3]. PMUs measure voltage, current and frequency at high speeds of even up to 60 – 120 time-stamped observations per second, compared to conventional technologies (such as SCADA) that measure once every 4 seconds. Thus, measurements taken by PMUs in different locations or with different owners can be synchronized and time-aligned. Monitoring and analysis of these measurements let observers identify changes in grid conditions for improved grid maintenance and reliability. A synchrophasor system includes phasor measurement units (PMUs), local data concentrators and higher-capability phasor data concentrators (PDC). The PDCs feed the consolidated data to analytical applications such as wide-area visualization, state estimation, and alarm processors and to archival systems.

There was an imminent need for designing networking architectures that efficiently interconnect the PMUs and the user applications that process the PMU data. The North American Synchro-Phasor Network (NASPInet) initiative was started to provide a reference networking architecture, shown in Fig. 1 [4]. In this reference architecture, the PDCs forward the information to their respective Phasor Gateways (PG). Thus, the PG is the entry point to the communication network similar to gateways in the traditional Internet. The supervisory systems that monitor the health of the power grid are connected to the PG. The term "Data Bus" (DB) is used to refer to the inter-PG communication infrastructure.

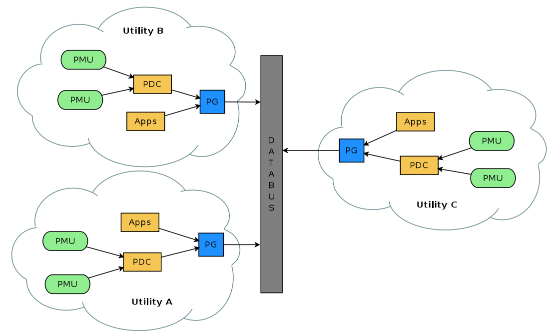


Fig. 1. NASPInet conceptual architecture.

The network architecture of the DB will have a significant impact on the overall performance of the system. An efficient network architecture is required to ably transfer the data from the PG to all the supervisors while meeting the NASPInet recommended data delivery constraints. The data generated in NASPInet is classified into five classes, of which are relevant to this work: Class A data used for high performance feedback control and Class B data for state estimation or feed-forward control. The key objectives are *low latency* and *high reliability*. A unicast-based architecture will not be efficient in meeting these goals. Hence, multicast based systems, especially publish-subscribe network architectures, have been deemed to be more appropriate [5], [6].

The focus of this paper is to present a comprehensive network-level performance evaluation of three different publish-subscribe architectures: (i) the GridStat middleware framework that provides publish-subscribe at the middleware level (above the transport layer); (ii) the Publish Subscribe Internet Routing Paradigm (PSIRP) [7], a clean-state Internet architecture that employs a native publish-subscribe paradigm at the routing layer, and (iii) the proposed publish-subscribe architecture at the routing layer that uses distributed hash table

978-1-4673-0911-0/12/$31.00 ©2012 IEEE

techniques a virtual-circuit approach for data forwarding.

II. Background and Related Work

This section presents some of the related work on network architecture design and related performance studies for Smart Grid systems [1].

The Internet has traditionally been based on the unicast paradigm, where flows (e.g. TCP, MPLS) are established between a source and a destination. The Internet routing protocols were also primarily designed for unicast routing. Mutlicast communications can be enabled by adding native support in the Internet routers which requires modification in these routers. An alternative is to establish *multicast overlay* networks that utilize the unicast routing support of the underlying network and realize multicast functions using application level routers.

The ability to meet low latency, availability and path redundancy depend on the network architecture. For example, the delay constraint for NASPINET Class A packets is 8–16 ms in the U.S. The network architecture designed for the *Smart Grid* has to take care of these Quality of Service (QoS) attributes. Accuracy, time alignment and high message rate depend on the devices instead of the network architecture. Proposing a network architecture which satisfies the above mentioned QoS attributes is challenging.

With the explosive growth of the Internet, the publish-subscribe paradigm emerged. In this paradigm, there are *publishers* who publish data on a regular basis and *subscribers* who subscribe to one or more publishers. This paradigm can be realized using overlay networks that use the underlying Internet's services to provide application specific network-related functionality. It can use both multicast and unicast routing support to efficiently support communications. It is derived from the "message queue" paradigm and is often supported using specialized publish-subscribe middleware. Scalability, QoS and reliability are some of the major concerns of this approach.

Recently, there have been efforts to natively support the publish-subscribe model at the router level. For example, the "Line speed publish/subscribe inter-networking" (LIPSIN) architecture is based on efficiently combining multicasting with guaranteed QoS [8]. In [9], methods to avoid global broadcasts in a publish-subscribe network are described. It also discussed several techniques for fault tolerance mechanisms to improve robustness of the system. The Publish Subscribe Internet Routing Paradigm (PSIRP) [7] is a clean state Internet architecture based on the publish subscribe paradigm. The emerging "Content-Centric Networking" (CCN) paradigm may also be considered for supporting the *Smart Grid* and is left for future study. In terms of designing a Smart Grid network architecture, the publish-subscribe paradigm seems natural to adopt.

There are other architectural frameworks present, in addition to the NASPINET framework, including EPRI's IntelliGrid^{SM} initiative [10], NIST's Smart Grid Interoperability Panel (SGIP) for creating standards and interoperability frameworks [11], and Europe's "Smart Grids European Technology Platform for Electricity Networks of the Future" (*Smart Grid*ETP) [12].

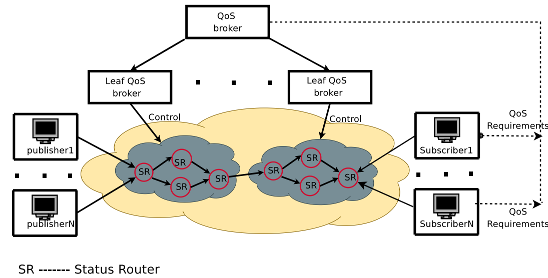


Fig. 2. GridStat architecture.

III. Publish-Subscribe and Multicast

In this section, we present two existing publish-subscribe architectures and a multicast based protocol.

A. GridStat

The *GridStat* project [13] has defined a status dissemination middleware that supports reliable delivery of status messages. The network architecture is based on the publish-subscribe paradigm realized using an overlay network that utilizes the underlying network's Quality-of-Service (QoS) support. It also uses a multicast approach for better utilization of network resources.

The GridStat architecture is shown in Fig. 2. The entities in GridStat architecture are: (i) **Publisher:** the Phasor Gateway (PG) acts as the publisher; the publishers connect to the network through edge status routers. (ii) **Subscriber:** Subscribers are destinations for data in the publish-subscribe paradigm, and are normally power vendor companies. (iii) **Status Router:** Status routers are the routers that run GridStat middleware applications. Status routers receive and forward status messages. Status routers that connect to either publishers or subscribers are called edge status routers. An area consists of several status routers that are interconnected and the GridStat architecture consists of several areas. Status routers that connect two or more areas are called border status routers. (iv) **Leaf QoS Broker:** A leaf QoS broker manages a set of status routers, publishers and subscribers. It maintains information about the topology and also QoS information that falls under it. The QoS broker sees whether any node under it is publishing the required status variable. If it finds one, then it will establish connection if the path satisfies all the necessary QoS attributes. If any of the above conditions fail, then it will forward the request to its QoS broker and the same procedure follows. This hierarchical flow reduces topological exchange information.

In [5], the authors discuss how we can achieve QoS even when the underlying network does not support QoS awareness. This can be done by establishing multiple paths thereby ensuring that the availability and delay of the system is enhanced. Source routing was used to reduce processing delay at intermediate nodes which is vital for Smart Grid.

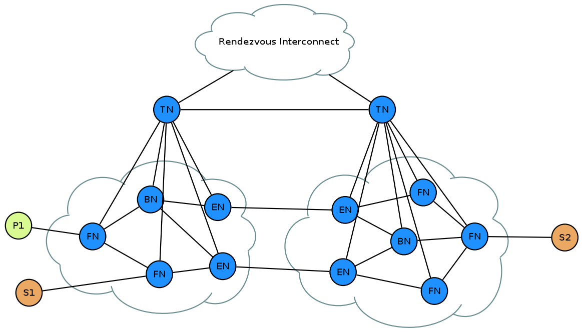


Fig. 3. PSIRP architecture

B. PSIRP

The Publish Subscribe Internet Routing Paradigm (PSIRP) [7] is a clean state Internet architecture based on a native (i.e. not overlay) realization of the publish subscribe paradigm. The PSIRP architecture is shown in Fig. 3. Here, the network is divided into many domains. Each domain has Forwarding Nodes (FNs), Branching Nodes (BNs), Edge Nodes (ENs) and a Topology Node (TN). There is a common rendezvous network for matching the publisher with the subscriber.

PSIRP uses source routing and hence does not overhead of state information at the forwarding node. Each data packet header contains a Bloom Filter which hashes the identifiers of all the links in the selected path. The FN queries the Bloom filter in the packet header with the Ids of all its links and sends the data on the links for which query returns true.

Branching nodes are used to cache publisher data. When it receives a request for the cached content, it will send the cached content without forwarding the request to the publisher. The subscribe request always passes through one of the branching nodes of the domain. The load balancing of the traffic to the branching node is done by the topology node.

Edge nodes are present at edge of the domain. The branching nodes forward the subscription requests through these edge nodes to the neighbouring domain(s). The Topology nodes gather the neighbour information from the forwarding nodes and build the network topology for the domain. They also exchange topology information with the neighbour domains' TNs using a BGP-like protocol. These nodes are responsible for selecting all the inter-domain and intra-domain paths.

Each domain is connected to a Home Rendezvous Network (HRN) which is connected to a central Rendezvous Interconnect (RI). The RI interconnects all the HRNs to form the Rendezvous Network. Rendezvous network performs the publisher to subscriber matching.

Routing in PSIRP: A publisher first advertises its publication Id in the Rendezvous Network that stores the publication Id along with publisher's node Id. A subscriber finds the node Id of a specific publisher by querying the rendezvous network. The branching node forwards the subscribe request to the edge node of the next intermediate domain in the path towards the publisher. The edge node forwards the request to the publisher, if the publisher is in the same domain. Otherwise, it forwards the request to one of the branching nodes in its domain. While forwarding the subscribe request, each node adds the Id of the link on which it is received, to the subscribe request. Using this information, the data is sent in the reverse path taken by the subscribe packet. Other routing details are available in [7].

The PSIRP architecture was not specifically designed to be used as a communication network for Smart Grid. But its efficient forwarding mechanism and publish/subscribe capability makes it a potential network architecture for the Smart Grid.

C. Source Specific Multicast

Multicasting refers to communication involving multiple senders and multiple receivers simultaneously. Ideally, every node can join and leave a multicast group dynamically. In the context of Smart Grid, single-source and multiple-receiver communication is the typical mode. Hence, for comparison purposes, we have implemented a simple source based multicast protocol called Source-Specific Multicast (SSM). Receivers who want to join a group G specify the channel (S,G), which is a combination of source address S and group address G. The shortest path multicast tree is constructed for a group G rooted at S. Only the specified source S can send a data to a group G. The details are not presented due to lack of space.

IV. Proposed Architecture

The proposed network architecture is based on the publish-subscribe paradigm, and is referred to "PSSG" in this paper. Some of the main problems with the current protocols are: they generate high advertisement data, take large time to process data at intermediate nodes and assume that the underlying network efficiently supports Quality of Service (QoS). To overcome the problem of generating high advertising data, our algorithm uses the distributed hash table mechanism [9] as described later. To support QoS, a simple but efficient multicasting mechanism is developed for delay optimization and data is sent on multiple paths to provide better reliability. To reduce intermediate processing logic, a form of source routing approach is used. Providing multiple paths with source routing might lead to loops while forwarding data. Our algorithm provides loop-free forwarding of data and also provides multiple source-destination paths.

The entities in the proposed architecture are presented in Fig. 4. This architecture is different from that of GridStat which uses an overlay middleware approach on top of the underlying network. It depends upon the unicast routing protocols of the same and for multicast, as described in [14]. In our proposal, the routing and data forwarding mechanism from publisher to subscriber requires modification at the router level. The objective is to understand how such a routing mechanism will help improve performance compared to an overlay based approach.

The **Rendezvous Node** stores all the publishers' address and corresponding publishing data ID. Subscribers can obtain information about publishers by contacting a rendezvous node. The **Broker** is a neighbor to either a publisher or a subscriber.

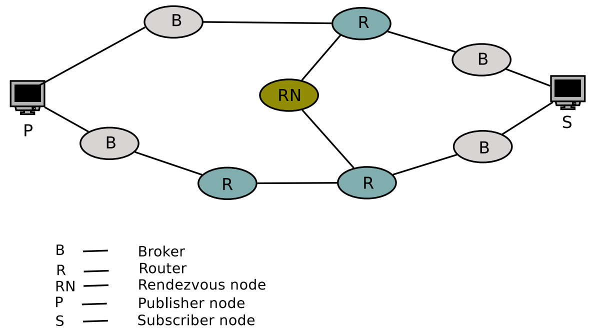


Fig. 4. The proposed publish-subscribe network architecture and entities.

A publisher does not know with which rendezvous node it must register; a subscriber does not know from whom to get data. A broker runs distributed hash algorithm to find a rendezvous node for the publisher or a subscriber. The **Router** router has the same meaning as in traditional Internet except that it forwards data to support the publish-subscribe model.

The network nodes discover the topology using the conventional link-state routing protocol's discovery mechanisms. A node's link state packet consists of information about its neighbor nodes and the delay to those nodes. Rendezvous nodes also sends their rendezvous node identifiers in the link state packet. Each node floods (with pruning) its link state packet to the network. Each node will receive link state packets from all the nodes in the network and by storing this data, topology information can be learned. Each node also knows about its rendezvous nodes and their respective ids.

In the *Smart Grid*, the PGs are the publishers. As PDCs can store historical data, they can provide all the types of data classes included in NASPInet to their subscribers. In our algorithm, publishers use the UDP (User Datagram Protocol) as the transport protocol. It is up to the application of the subscribers to check whether the data is reliable and accurate. Since data should be transmitted with minimal latency, UDP is more suitable than Transmission Control Protocol (TCP) for the NASPInet requirements. TCP uses acknowledgments for data delivery and congestion control mechanisms. Congestion should not reduce the flow rate of critical data. NASPInet requires critical data to be transferred at 60, 90 or 120 samples/sec. Thus, if we use TCP, the sampling rate would be reduced depending on the network performance. In the network layer, publishers use IPv4 (Internet Protocol version 4). In the proposed architecture, reliability is achieved by establishing multiple paths from the publisher to the subscriber. This is done at the expense of increasing the network traffic; hence, the number of redundant paths has to be minimized. Also, acknowledgments are sent only for control packets and not for data packets.

Each publisher registers with the network by sending a register packet to its default gateway (its broker). The broker then finds the corresponding rendezvous Id by computing a hash value of the publish data id. The same hash function is used by all the broker nodes for finding rendezvous ids. In order to effectively distribute the work between rendezvous nodes, an efficient hash function has to be used. After finding the rendezvous Id from the hash function, the broker finds the rendezvous Id's node address by using topological information provided by the link-state routing protocol.

A subscriber sends a subscribe packet which contains the Publisher data Id. This subscribe request will be sent to the broker node. The broker node finds the rendezvous node using the hash function and sends the subscribe packet to the rendezvous node, via other intermediate nodes.

Once the rendezvous node receives this packet, it checks whether any path establishment process is going on for the corresponding publisher data Id. If it is going on, it waits for route establishment acknowledgment. Otherwise, it finds the nearest k nodes that are registered to the publisher data id from its database. Once the rendezvous node finds k nodes, it sends a subscription request packet to all the k nodes. The subscription request packet contains publisher address, publisher data Id, the nearest router address as the subscription request packet's destination and all the intermediate routers' addresses from nearest router to the subscriber. The rendezvous node stores all k nodes for a given subscriber in its database.

Route Establishment: In our algorithm, we use a static routing mechanism instead of dynamic routing to make sure that traffic delays are more predictable. This is similar to establishing a virtual circuit. In case of link or other failures along a path, multiple redundant paths will help improve reliability, as mentioned earlier.

Once the publisher or any intermediate router receives a subscription request packet, it creates a route establishment packet. Upon receiving a route establishment packet, a router checks whether this router has already subscribed to the same publisher data Id. If not, it sends a publisher register packet to the rendezvous node. After receiving this packet, the rendezvous node stores this router's address into its publisher data Id subscriber table. After storing the publisher data Id and its corresponding next hop address into its database, the router sends the route establishment packet to its next hop. In this way, the route is established from the nearest router to subscriber. On receiving a route establishment packet, the subscriber sends a route-establish acknowledgment packet to the rendezvous node.

After receiving publisher register acknowledgment packet and at least one subscription request from the subscriber, the publisher starts sending data. If an intermediate router receives a data packet, it sends the data packet to all the next hops that are stored in its forwarding table corresponding to the publisher data Id. In this way, data packets are delivered to all the subscribers in an efficient manner. A detailed example of the architecture's operation is presented in [6] and is omitted here due to lack of space.

V. Performance Study

The proposed *Smart Grid* architecture has been implemented in OMNeT++ simulator's INET framework, using the

PSGen topology generator [15]. Since data forwarding in intermediate nodes is different from the Internet, we implemented a publish-subscribe layer which lies between the data link layer and the network layer. Path establishment is performed in the network layer.

The metrics studied are: path propagation delay and end-to-end packet delay. Propagation delay of a path is computed by adding the propagation delay of all the links in the path. Propagation delay is chosen as a metric because the proposed and the existing architectures differ in the way the paths are selected. End-to-end delay is time taken by the packet generated by publisher to reach the subscriber.

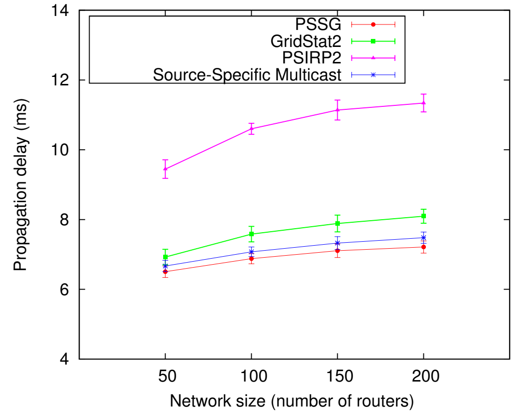
The parameters are: generation rate of 60 samples per second; packet length of 8,192 bytes to present a large aggregated packet; propagation delay per link is 1 ms to 2 ms (uniformly distributed) – to represent medium-length backhaul links. The number of subscriptions per subscriber is 10. Four sets of 30 topologies, were used for the comparison. All topologies in a set contains equal number of subscribers, publishers and routers. Sets 1, 2, 3 and 4 contain 50, 100, 150 and 200 routers/publishers/subscribers respectively. Of the 30 topologies, ten each were created using the Waxman model [16], the Barabasi-Albert model [17] and the random model. The results are presented with 95% confidence intervals. Each simulation runs for 120 seconds. During the first 70 seconds, the initialization and connection establishment takes place and in the last 50 seconds, the data transfer takes place.

A two-level GridStat architecture called Gridstat2 is studied. Here, the Leaf QoS brokers are grouped into five-member groups and connected to a level-1 QoS broker. The level-1 QoS brokers are then connected to a single level-2 QoS. Performance results showed that GridStat2 performed consistently better than GridStat1 [18] and has been used in the subsequent analysis. For PSIRP, a system with two branching nodes per domain (termed PSIRP2) has been studied. Here, the traffic is load balanced between two branching nodes for better performance. Additional details are available in [18].

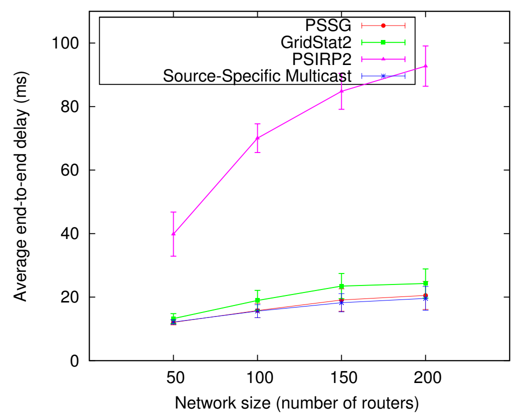
A. Comparison of Network Architectures

The performance of PSSG, GridStat2 and PSIRP2 in terms of mean propagation delay and end-to-end delay is shown is Figure 5, for link capacity of 100 Mbps. It is seen that PSSG has the least propagation delay. This is because it always finds the shortest path, whereas GridStat2 and PSIRP2 do not always finds the shortest path because they do not consider the complete topology while finding the end-to-end path. Also, in PSIRP, all the traffic has to pass through the branching node, which leads to higher delays than GridStat. Hence the propagation delay of the paths computed by PSSG and SSM is less compared to that of GridStat and PSIRP. The performance of PSSG and SSM is similar since both utilize similar multicast routing mechanisms.

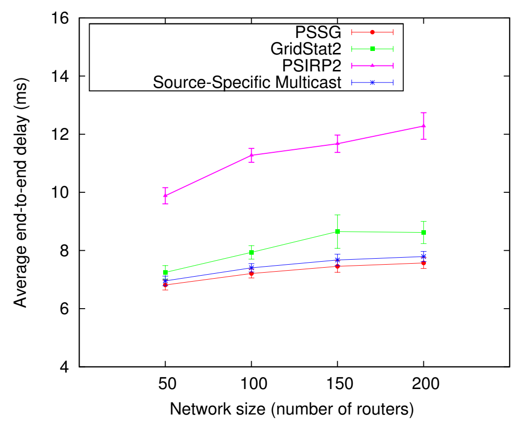
In terms of end-to-end delay, PSIRP2 has the highest delay, while the other systems provide comparable delay. The relative difference is not statistically significant. Figure 5(c) presents end-to-end delay results with link capacity of 1 Gbps. Due to the higher data rate, transmission time and hence queuing delay will be lower. The results are similar to that seen with the 100 Mbps scenario.



(a) Mean Path Propagation Delay



(b) Mean End-to-end Delay, 100 Mbps links

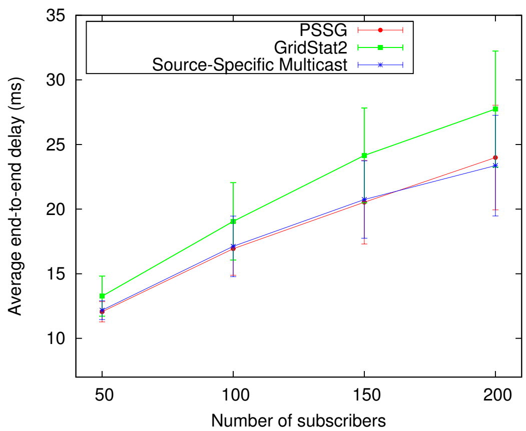


(c) Mean End-to-end Delay, 1 Gbps links

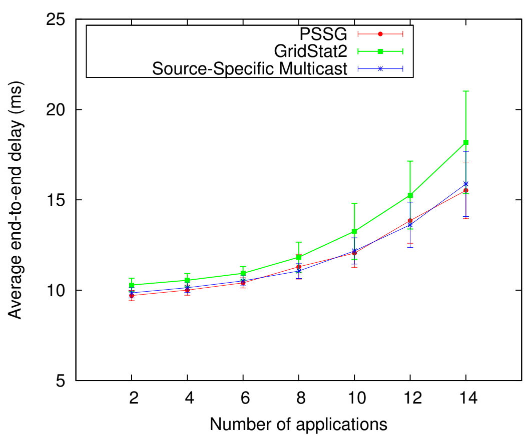
Fig. 5. Comparison of GridStat2, PSIRP2 and PSSG architectures.

B. Varying number of subscribers

Since PSIRP has been shown to have higher delays, only PSSG and GridStat are compared further. The number of subscribers was varied from 50 to 200 keeping the number of publishers and routers as 50. It was seen that PSSG had lower propagation delay compared to GridStat2 (results not shown here). The mean end-to-end delay results are shown in results are shown in Figure 6(a). There was only marginal improvement in terms of end-to-end delay as seen in the figure.



(a) Varying number of subscribers



(b) Varying number of applications

Fig. 6. Comparison of GridStat2 and PSSG, varying number of subscribers and number of applications per subscriber

The number of applications per subscriber was varied from 2 to 14 keeping the number of publishers, subscribers and routers as 50. The mean end-to-end delay results are shown in Figure 6(b). It is seen tat PSSG does have lower mean delay compared to GridStat2; however, since the confidence intervals overlap, the reduction in mean delay is not statistically significant.

VI. Conclusions

In this paper, we have presented a comparison of different publish-subscribe based network architectures for the *Smart Grid* with the goals of low delay and high reliability. The existing mechanisms compared were based on GridStat, PSIRP and Source Specific Multicast. A new publish-subscribe network architecture was also presented. The proposed architecture was compared to GridStat, PSIRP and SSM architectures. The results show that the proposed PSSG architecture provides lower path propagation delays due to more efficient path computation. However, the end-to-end delays of SSM, PSSG and GridStat are comparable. Since PSSG is a native realization of publish-subscribe paradigm and GridStat is a middleware based approach, it is necessary to implement the two approaches in real systems to understand the overhead reduction with a native publish-subscribe architecture. Future studies can also consider filtering of PMU data when the publishing rate is higher than the subscriber's requirement.

Acknowledgments: This work was supported in part by an Open Collaboration Research (OCR) grant from IBM. The first author is presently with Alcatel-Lucent Bell Labs, India; this work was done while he was pursuing his M. Tech. degree at IIT Madras.

References

[1] A. Bose, "Smart Transmission Grid Applications and Their Supporting Infrastructure," *IEEE Transactions on Smart Grid*, vol. 1, pp. 11–19, June 2010.

[2] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart Transmission Grid: Vision and Framework," *IEEE Transactions on Smart Grid*, vol. 1, pp. 168–177, Sept. 2010.

[3] M. Adamiak, D. W. Premerlani, and D. B. Kasztenny, "Synchrophasors: Definition, Measurement, and Application," in *Second Carnegie Mellon Conference in Electric Power Systems*, pp. 11–19, 2006.

[4] Y. Hu, M. Donnelly, T. Helmer, H. Tram, K. Martin, M. Govindarasu, R. Uluski, and M. Cioni, "NASPInet Specification - An Important Step toward Its Implementation," in *Proc. of Hawaii International Conference on System Sciences (HICSS-2010)*, pp. 1–9, Jan. 2010.

[5] D. Bakken, C. Hauser, and H. Gjermundrod, "Delivery Requirements and Implementation Guidelines for the NASPInet Data Bus," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 37–42, Oct. 2010.

[6] K. Kumar, "Performance study of distributed and publish/subscribe architecture for smartgrid networks," Master's thesis, Dept. of CSE, Indian Institute of Technology Madras, June 2011.

[7] V. Dimitrov and V. Koptchev, "PSIRP project: Publish-Subscribe Internet Routing Paradigm: New Ideas for future Internet," in *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, pp. 167–171, 2010.

[8] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 195–206, August 2009.

[9] P. R. Pietzuch and J. Bacon, "Hermes: A Distributed Event-Based Middleware Architecture," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, ICDCSW '02, (Washington, DC, USA), pp. 611–618, IEEE Computer Society, 2002.

[10] P. Haase, "Intelligrid: A smart network of power," *EPRI Journal*, pp. 26–32, 2005.

[11] "NIST Smart Grid." http://www.nist.gov/smartgrid, Sept. 2011.

[12] "Smart Grids European Technology Platform." http://www.smartgrids.eu/, Sept. 2011.

[13] K. H. Gjermundrod, I. Dionysiou, D. Bakken, C. Hauser, and A. Bose, "Flexible and Robust Status Dissemination Middleware for the Electric Power Grid," tech. rep., Washington State University (EECS-GS-003), 2003.

[14] H. Gjermundrod, D. E. Bakken, and C. H. Hauser, "Scalable wide-area multicast with temporal rate filtering distribution framework," in *Proc. of IEEE International Conference on Scalable Computing and Communications (SCALCOM)*, (Paphos, Cyprus), pp. 1–8, 2011.

[15] "PSGen Topology Generator." http://www1.inf.tu-dresden.de/~rf913578/psgen.html, Aug. 2012.

[16] "Waxman random network topology generator." http://www2.math.uu.se/research/telecom/software/stgraphs.html, Aug. 2012.

[17] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Nature*, vol. 286, pp. 509–512, Oct. 1999.

[18] R. Muthukumar, "Performance study of fault tolerance mechanisms for Smart Grid networks," Master's thesis, Dept. of CSE, Indian Institute of Technology Madras, May 2012.