SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications

Young-Jin Kim, Jaehwan Lee, Gary Atkinson, Hongseok Kim, and Marina Thottan,

Abstract—Smart Grid applications are imposing challenging requirements of security and reliability on the N-way communication infrastructure being designed to support multiple grid applications. These challenges stem from the increasing incorporation of distributed renewable energy sources on to the grid, the rising deployment of electric vehicles, and active consumer participation into power grid operations, all of which communicate with the utility control center with varying degrees of priority and security. To address these challenging requirements, we propose SeDAX, a SECure Data-centric Application eXtensible platform for Smart Grid applications. SeDAX implements scalable, resilient and secure data delivery and data sharing in a wide area network. The platform can scalability handle high volumes of data generated by both applications and sensors. The SeDAX architecture has as its basis a Delaunay Triangulation (DT) network. The properties of the DT graph are leveraged to scalably support secure data-centric (or information-centric) group communication. The primary goals of this platform are to support communication resilience and data availability. The key functional blocks of the SeDAX platform are: (1) a geographic hash forwarding algorithm that operates over the DT graph (DT-GHF), and (2) a DT-based data replication scheme. The forwarding and replication schemes are scalable and cost effective in terms of communication overhead and memory. We describe the design details of the SeDAX platform and present empirical results on the performance of SeDAX as compared with other geometric-based alternatives such as Geographic Hash Table (GHT) forwarding and Content Addressable Networking (CAN). The operation of SeDAX is illustrated in the context of implementing demand response, a known Smart Grid application.

Index Terms—Information-centric, end-to-end security, geometric hashing and forwarding, Delaunay triangle.

I. INTRODUCTION

"Smart Grid" refers to the next-generation electric grid designed to: enhance the resilience of the grid to power flow disruptions, improve energy efficiency, and reduce carbon emissions. To accomplish these goals, the modern grid will incorporate a wide variety of Smart Grid applications such as: distributed renewable energy sources, electric vehicles, and intelligent interactive consumer applications. However, one of the main impediments to the deployment of Smart Grid applications is the limited capability of today’s utility communication infrastructure in terms of scalability, reliability and security. The next generation Smart Grid communications will necessitate the following fundamental changes to the infrastructure: (1) two-way communication (today primarily one way), and (2) distributed control (today centralized control).

Consider two important Smart Grid applications, namely, the increased incorporation of renewable energy sources and large scale deployment of electric vehicles. As compared to the power grid today, these two applications alone will introduce significant variability in the generation and consumption of power [1], [2]. Therefore, the important problem of balancing power supply and demand would become more challenging in the absence of an advanced communications platform. These new grid applications could also create fragmentation in the market and complex inter-dependencies on different parts of the grid thus causing serious concerns about the viability of a centralized control. To address these fundamental requirements, Smart Grid must support distributed data sharing and distributed control across wide area networks and across multiple administrative organizations. Also the choice of network technology must adequately address concerns about scalability, availability and security of the grid.

The recent 2005 Houston blackout [3], is a good example to illustrate the importance of a secure data sharing network. In this incident, due to lack of sufficient support for secure sharing of phasor measurement data, the ability of the grid to isolate the blackout and recover from it was severely compromised. The creation of NASPInet (North American SynchroPhasor Initiative network) [4] was indeed prompted by such incidents in the past. NASPInet is a framework for providing a robust and secure synchronized data sharing infrastructure for the interconnected North American electric power system. NASPInet enables secure sharing of synchrophasor data across multiple administrative organizations. However, today NASPInet is a specialized, closed, and non-IP based infrastructure for sharing synchrophasor data. As addressed by Shaw [5], the implementation of a modernized, Smart Grid-enabled utility requires more than just secure data sharing. It is necessary to support active control of power consumption and
generation all the way to the consumer level. This increased customer participation will broadly impact the reliability and efficiency of power distribution. Hence, the reliability and security features of the Smart Grid communication infrastructure must penetrate into the consumer side. This extension of the infrastructure is necessary to enable applications such as: automated metering, building energy management systems, electric vehicles, distributed solar panels, residential energy storage, and advanced demand response programs. NIST (National Institute for Standards and Technology) working group on Smart Grid [6] has identified the requirements of reliability, latency, security, and scale for the different communication flows associated with each Smart Grid application. The reliability requirement of the communication flows range from 98% to 99.5%. The one way latency of the communication flows range from 10 minutes for firmware updates to less than 10 seconds for electric vehicle charging applications. Security levels vary depending on the sensitivity of the data with the highest levels of security for meter communications in demand response applications, e.g., end-to-end (E2E) security is required to protect meter data privacy. To the best of our knowledge, as of today there does not exist a scalable, reliable, and secure communication platform that spans across both the consumer side and the provider side of the power grid.

To address these challenges, we propose SeDAX, a Secure Data-centric Application eXtension platform for Smart Grid applications. SeDAX can flexibly satisfy the requirements of Smart Grid communications while achieving the scalability, availability and security requirements. The use of a data-centric platform enables secure data sharing and supports both transaction and query-based communications. Thus it provides an $N$-way communication infrastructure that spans across multiple grid applications and organizational domains.

Figure 1 depicts the SeDAX architecture. SeDAX provides two kinds of data-centric communication methods: streaming-based data dissemination [7] and query-based data retrieval [8], [9]. Both of these communication methods are enabled by a secure overlay network on top of the existing TCP/IP network. Within this overlay, SeDAX uses a novel scalable and localized data forwarding method which provides good routing performance with minimal routing overhead. This is a key feature of SeDAX and is based on the properties of the overlay network, which is built on a Delaunay Triangulation (DT) graph.\footnote{For a given set $V$ of vertices, a DT($V$) is a triangulation such that no vertex in $V$ is inside the circum-circle of any triangle in DT($V$).}

The DT graph structure also provides another key feature of SeDAX, namely, self-configurable group communication. The security framework of SeDAX covers both information and protocol level security [10] which is briefly discussed in Section V-B. The focus of this paper is to discuss the scalability and resilience of the SeDAX platform.

Our main contributions are summarized as follows. (1) We design a scalable, resilient and inherently-secure data-centric communication platform. Our scaling is in terms of the size of the forwarding tables, and support for a large number of data topics and application end points. (2) We deduce and prove several theorems of DT graphs that are relevant for our SeDAX architecture in terms of self-healing and self-configurability. (3) We propose DT-GHF (Geometric Hash Function), a novel resilient and efficient message forwarding algorithm and prove its path convergence. The hash functions employed can be designed to provide increased data availability. (4) Using extensive simulations, comparative studies with DHTs (Distributed Hash Tables) as well as real implementation we show that the proposed SeDAX and DT-GHF are cost effective and outperform other alternatives in terms of smart grid relevant metrics: scalability, resiliency and data availability.

The paper is organized as follows: In Section II we place our architectural and algorithmic contributions in the context of prior work. Sections III and IV outline the architecture of SeDAX and provide a detailed analysis of the Delaunay triangulated overlay network. The implementation details of the prototype SeDAX platform is provided in V. The application of the SeDAX platform to implement demand response function is discussed in Section VI. Our extensive evaluation studies and additional features of SeDAX are discussed in Sections VII and VIII respectively.

II. RELATED WORK

Our work is placed into context with prior work in four different topics areas: (1) data management architecture, (2) publisher-subscriber systems, (3) DHT-based overlay networks and DT graphs, and (4) geometric forwarding. **Data management architectures:** We first overview recent research that has examined the Smart Grid communication infrastructure and information-centric networks. One proposal, NASPInet, aims to build a data sharing bus among a set of trusted fully meshed nodes [4], but there is no explicit consideration of secure communications over an IP network. OSISoft Process Information (PI) system [11] is a commercial data management solution used to manage and coordinate data collected from sensors. However, in the PI system, data servers are centralized repositories and placed in secure locations. This approach is not scalable to accommodate the distributed data sources in Smart Grid. In contrast, the proposed SeDAX platform is shown to scale effectively for distributed data accessibility, storage and control. In [12], Koponen, et al. propose an information-centric network architecture which involves a clean-slate redesign of network naming and name resolution. SeDAX does not require re-architecting the network. In terms of security, [12] simply assumes protected channels. Van Jacobson et al. [13] consider content-centric networking, and address content security. In their setting, the main issue is data authentication (not confidentiality), and they use public key signatures. The SeDAX platform addresses and enables both user authenticity and data confidentiality by managing security through topic groups. **Publisher-subscriber systems:** Publisher-subscriber communication model (pub-sub) [7] has been identified as a necessary application programming interface to realize the information-centric networking concept. Compared with conventional
client-server communication model, the pub-sub model is: (1) inherently distributed, peer-to-peer, scalable, and enables multicasting; and (2) it has highly resilient operations since it does not require knowledge of end-node addresses, which in turn also prevents denial of service attacks and avoids single points of failure and bottlenecks. Many commercial pub-sub implementations exist: TIBCO Rendezvous, RTI Data Dissemination Service, IBM WebSphere Message Queue (MQ), etc. However, these implementations are assumed to run on highly secured and stable environments such as enterprise service networks or military networks where cyber-security is ensured through external security mechanisms and also requires extensive computing resources in terms of processor, memory, and bandwidth requirements. By design these systems work on small scale (not Internet-scale) systems and thus do not leverage the scaling benefits of a pub-sub communication model. The value of pub-sub is more apparent in large scale machine-to-machine systems that spans across unsecured environments and has end devices with limited computing resources, e.g., million-scale sensor networks for advanced metering or distributed energy resources (DERs) management. Content-based pub-sub systems along with in-network processing for filter matching are also inappropriate for Smart Grid applications requiring E2E security due to the message decryption that is necessary for in-network processing.

Recently, a research prototype, GridStat [14], [15], has been introduced as one candidate proposal for NASPInet to enable distributed synchrophasor applications. The GridStat project proposes a pub-sub network of message routers (data plane) controlled by a hierarchical management plane (a QoS broker tree and a centralized configurable security framework) for meeting the NASPInet QoS requirements [16]. For a given pub-sub group, the rendezvous between publishers and subscribers is statically managed by the hierarchy. The GridStat approach may meet the QoS requirements for synchrophasor data sharing. However, it is not applicable for Internet-scale applications such as advanced metering, EV charging, or DER management due to the following reasons. Firstly, it suffers from scaling problems with regard to managing the E2E security and QoS requirements. In GridStat per-flow (source-destination pair) state for both the data and management plane is managed at a network level. Secondly, the static management through one QoS broker tree poses resilience issues: when a message router or a QoS broker in the tree fails, some regions in a GridStat network may experience service disruption for a non-negligible time period due to lack of self-restoration. Thirdly, GridStat inherits some security weaknesses of conventional group communications: it does not scalably support secure group communications under the condition that using public key operations may not be suitable due to the E2E latency concerns or limited computation ability of end nodes. Lastly, GridStat’s E2E latency grows linearly with the number of message routers in a network.

**DHT-based overlay networks and DT graphs:** Many large-scale pub-sub systems such as SCRIBE [18] and Hermes [19] have been built over a self-configurable overlay network that employs a DHT algorithm such as Pastry [20], Tapestry [21], and CAN [22]. The DHT-based approaches are highly scalable but offers weak worst-case performance guarantees and have a relatively high routing overhead due to large identifier space. They also require O(log N) route memory per node where N is the total number of nodes in an overlay network. The SeDAX design is inspired by the geographic hash table (GHT) system proposed by Ratnasamy, et al. [8]. However, the GHT system often requires complicated recovery-mode forwarding schemes [23], [24] to reach the destination location and furthermore uses an inaccurate and expensive replication scheme. This is not the case in SeDAX (as shown in Theorems 1 – 3) since we leverage DT graph properties and implement geometry-based forwarding algorithms on DT graphs.

DT graph is provably known to be a t-spanner graph, where t ≈ 2 [25], [26], which implies that there is at least one forwarding path not exceeding t times the Euclidean distance between source and destination. Note, however, that it is a theoretical bound requiring global knowledge, and may not be directly applicable to distributed and localized forwarding schemes. Parallel Voronoi Routing [27], which is a multi-path forwarding scheme, is known to be the only localized algorithm achieving constant worst-case path-length bound (5√(1+√5)π ≈ 45.748). However, it is significantly outperformed by greedy forwarding algorithms in terms of average forwarding performance [27], [28]. In the first ever greedy forwarding algorithm proposed by Finn [29], from any node v, data is sent to the node that is geographically closest to the destination t by using only neighborhood information. However, a prerequisite of greedy forwarding algorithm is that all nodes in a network have to know their own location in a given coordinate space.

**Geometric forwarding:** Geometry-based forwarding (including greedy forwarding), one of the localized forwarding schemes that does not rely on a global network view, has received a lot of attention due to its light-weight overhead, i.e., no routing table and no pre-computation to keep up-to-date state. There has been a considerable amount of research activity in this area: beginning with initial sketches of forwarding based on position information [30], [29], [31], to the first proposals for practical schemes GFG [32], GPSR [23], followed by refinements of these proposals for efficiency [24], [33]. However, most of this research relies on the unit-disk graph (UDG) assumption. Although this assumption makes sense in wireless multi-hop networks, it is not necessarily the case for wired or overlay networks where connectivity between two nodes is not affected by Euclidean distance.

---

2Conventional symmetric-key based group communications have limitations on key refreshment, non-repudiation, and confidentiality. Whenever a new member joins a group, the group key shared by members has to be refreshed [17]. Any subscriber can disguise itself as a publisher since a group credential is shared by all members in the group. Messages from any publisher is readable by all members in the group.

3For any two vertices u and v, those two vertices must be connected by an edge if they are less or equal to some distance d apart, but must not be connected by an edge if they are greater than d apart.
Furthermore, in the above geometric forwarding schemes, complicated recovery forwarding [31, 23, 24] is mandated to always reach the destination due to divergence of greedy forwarding (the default engine of geometric-based forwarding schemes) even in the UDGs. There have been two recent papers using DT for efficient greedy forwarding on multi-dimensional Euclidean space; Ghaffari, et al. [34] have proven that a graph with DT is a necessary and sufficient condition for the convergence of greedy forwarding in Euclidean space; Lam and Qian [35] showed experimental results for this convergence based on their distributed DT construction and greedy forwarding on wireless ad-hoc networks.

III. SeDAX OVERVIEW

We describe the system architecture for the SeDAX platform with specific emphasis on the data-centric communication paradigm. This system architecture gives a broad view of SeDAX operation.

A. Data-centric Communications

We illustrate the need for data-centric communication using an example Smart Grid application. The Smart Grid application being considered is for efficient implementation of the demand response (DR) function. The DR function uses automated meter data collection and automated updates of real time pricing of electricity. This application is depicted in Figure 2 and comprises of three main services: (1) Collection of metered data: Aggregating nodes periodically (minute-to-day scale) send electricity consumption data on a per consumer basis to utility-side meter-data collectors or electricity load-analysis servers. Metered data must be sent to the utilities for real-time demand monitoring and estimation with a latency requirement in the order of seconds [6]. (2) Publishing of real-time electricity price information: A set of multicast nodes periodically (minute scale) send pricing information to the consumer. (3) DR and Load Management Server negotiations for demand reduction with participating meter nodes for the implementation of demand response function. Since these negotiations are critical to meeting the power deficit of the utility, the corresponding latency requirements are in the order of seconds [6].

These 3 communication flows also differ in terms of privacy and security requirements. Pricing data is public knowledge and therefore has no privacy requirements, while the meter data is private customer specific data and hence has greater E2E security and privacy requirements. The server negotiation functions are required to have very high levels of data integrity. Furthermore since pricing data is public it is inefficient to send this data over secure connections to individual customers. It is more efficient in terms of resource consumption to periodically publish the pricing data to a few network locations and respond to end-customer pricing queries from this location. This approach avoids traffic bursts on the network and improves network latency and efficiency for other grid applications. To enable this efficient communication paradigm it requires that data be stored and queried using the notion of topics and topic groups - also called data-centric communications. In data-centric communications, the operations of data generation and data consumption are decoupled.

Considering our example application, in the data-centric context, a meter periodically generates data, but it is agnostic how (data consumption rate) the data is consumed or by whom. However, as the data is self-named (topically defined), the data consumers can determine the source of the data and location of the data on the network. This decoupling of data generators and data consumers has a significant impact on making the network scalable and has inherent security features such as address invisibility of the data generators and consumers.

Data-centric communication is well suited for Smart Grid applications because: (1) applications are interested in the data specifics rather than the specific network address of the data source, (2) a large volume of data is generated and delivered, and (3) data heterogeneity originating from a broad spectrum of data sources, e.g., sensors/meters. The SeDAX platform is data-centric and differs from existing data-centric approaches as it is built on the existing IP network, and the notions of data security and reliability are inherently built into the design.

B. SeDAX Platform Elements

A shown in Figure 3, the SeDAX network is an overlay network which consists of a Delaunay Triangulated (DT) graph having SeDAX nodes as vertices and transport layer connections between any two neighboring nodes as edges. A SeDAX node is a computing entity that enables topic-group communications. The SeDAX middleware is a software library that provides Smart Grid applications with interfaces to the SeDAX network. A Smart Grid application is executable in any computing entity (even on SeDAX nodes as necessary). Hardware platforms for SeDAX middleware can have a broad range of computation capability, from TI MSP430 8MHz microcontrollers [36], ARM-based platforms, and up to blade servers. In contrast, hardware platforms for SeDAX nodes require a computation capability in the order of 500MHz.
computing power. A network of security servers is deployed within the security perimeters of utilities, and is responsible for the secure control of SeDAX platform elements.

C. Topic-group Communications with Geographic Hashing

Central to SeDAX is topic-group communications, where each topic is defined by fields such as data type, location, and time. For a given topic, a uniform hash function shared by SeDAX middleware instances produces a location in a given planar coordinate space. Given a topic’s search key, a hash function is used for quickly locating a data record from a large data set. More specifically, in SeDAX the hash function maps the topic to the hash (index) which in turn gives the location of the corresponding data for either retrieval or storage purposes.

For a given topic-group \( g \) (e.g., AMI data at 12:00:00 EST on March 1 2012, New York area), all messages associated with \( g \) are sent to a destination location \( t \) in a given geographic coordinate system. Figure 4 shows that SeDAX node \( v \) is the closest such node to location \( t \) determined using a form of a geographic hash function [8]. Any instance of a SeDAX middleware has to be authenticated for accessing a topic-group \( g \). Therefore, the joining entity sends a group-join message (destined to location \( t \)) to a SeDAX node associated with it (at system initialization time) and this message is forwarded over a DT-based SeDAX network using our multi-hop forwarding algorithm, DT-GHF, discussed in Section IV-B. When the message reaches the SeDAX node \( v \) closest to topic location \( t \), node \( v \) establishes a secure session with the message source so that the message source can exchange authentication messages with a security server connected to node \( v \). A SeDAX node can neither see nor forge the authentication messages as message source’s credentials are known only by security servers. This hash function based approach to access data enables fine-grained access control and replaces traditional naming services that have resilience concerns.

D. Security Considerations

To the best of our knowledge, most existing pub-sub implementations do not support a scalable E2E security framework for distributed data sources across multiple organizations. Thus in Smart Grid applications that span from the consumer side to the utility control center, it is possible that unauthorized subscribers can access the data, and also unauthorized publishers can publish malicious messages into the pub-sub systems. To solve these security problems, existing pub-sub implementations necessarily rely on external security mechanisms such as TLS [38] or IKE/IPSec [39], [40]. However, using these E2E solutions (direct secure sessions for each pub-sub pair) is inappropriate for large-scale pub-sub systems since it requires maintaining one secure session between any two end points. Furthermore, as it breaks the decoupling between publishers and subscribers, the benefits of pub-sub communications are lost. On the other hand, for a pub-sub system implemented using message routers, segmented secure sessions (between publishers and routers, among routers, and between routers and subscribers) rather than direct secure sessions are used. Therefore messages can be listened to, modified, or forged in the event that the message routers are compromised.

By contrast, the SeDAX platform provides a scalable and inherently secure pub-sub implementation for large scaled Smart Grid applications through the use of hash-based topic group management that enables secure group-based communication. Notably, we use symmetric-key based group communications, that can scalably address security issues such as key refreshment and data confidentiality (See Section V-B4). Non-repudiation is indeed a concern and can be addressed within the SeDAX framework, however the description of this approach is beyond the scope of this paper.

IV. FOUNDATIONS OF SEDAX

The theoretical foundation for SeDAX is a Delaunay Triangulation based forwarding algorithm. This algorithm combines greedy forwarding with a geographic hash function [8], and a DT-based replication scheme. The proposed novel forwarding algorithm provides resilience in terms of forwarding and data availability. Before describing the details of the algorithm we first identify some key properties of the DT graph.

A. Properties of Delaunay Triangulation (DT)

A DT graph overlay is chosen as the foundation for the SeDAX platform due to the following reasons:

1) Basic Properties: DT for route convergence: A DT graph is a necessary and sufficient condition for the convergence of greedy forwarding algorithms [34].

DT is extensible for routing in multidimensional space:
As shown in the most-recent paper [35], DT can be easily extended to multi-dimensional space, together with any associated greedy forwarding algorithms.

DT is a sparse degree network: Sparse-degree network graphs like DT have efficient forwarding schemes with a small-size look-up memory, paths with short total distance, and a small number of hops [41], [26].

Observation 1: In any DT graph, the average vertex degree (the average number of neighbors) is less than six, 5

Observation 2: Since a DT is a maximal plane graph, all subgraphs of the DT are also plane graphs.

5Since a DT \( T = (V, E) \) is a plane graph, then the number of edges \( |E| \) is at most \( 3|V| - 6 \) where \(|V| \geq 3 \) is the number of vertices. Informatively, a plane graph has no edge crossing another edge in the graph Since each edge has two ends, the average vertex degree is \( 2|E|/|V| \leq (6)|V| - 12/|V| < 6. \)

5Some pub-sub systems have inherent and configurable security frameworks [15], [37]. However, they do not scalably address E2E confidentiality and authentication under environments where using public-key operations is inappropriate, e.g., latency-sensitive applications or low-cost devices.
DT graph and its non-DT subgraphs that can occur during node failures. We make use of the Voronoi cell⁹, concept for ensuring simple and efficient message delivery. Each SeDAX node has its own Voronoi cell e.g., in Figure 5 (a), node v owns a Voronoi cell corresponding to the blue convex polytope and for all points within that Voronoi cell, node v is closer than all other nodes in the DT graph. SeDAX nodes do not update their Voronoi cell information during transient state caused by SeDAX node failures.

Using this framework we presenting the DT-GHF algorithm, for which the pseudo-code is shown in Figure 7. We first define terminologies to be used in the pseudo-code: \(v_i\) is the SeDAX node visited at the \(i\)-th hop during forwarding; \(N(v_i)\) is a set of \(v_i\)'s neighbors; \(l(v)\) is the identifier of SeDAX node \(v\); \(l(v_i)\) is the location of SeDAX node \(v_i\) in Euclidean space \(S\); \(u\) is a local minimum node visited during greedy forwarding; \(|pq|\) is Euclidean distance between location \(p\) and location \(q\); \(N_g(v_i)\) is a set of greedy forwarding candidates among \(v_i\)'s neighbors; \(C(v_i)\) is \(v_i\)'s Voronoi cell in Euclidean space \(S\). Also, the following tie-breaker rules are defined for handling the degenerate cases that arise during node-placement.

**Definition 1:** If a circum-circle \(C_b\) of \(T\) has more than three vertices on it, one vertex with the lowest identifier on \(C_b\) has edges incident on all other vertices on \(C_b\) (see Figure 6).¹⁰

**Definition 2:** For a vertex set \(V_b\) \(\subseteq V\), when location \(t\) is equidistant from all vertices in \(V_b\), a vertex with the lowest identifier in \(V_b\) is declared to be the closest vertex to \(t\) among all vertices in \(V_b\).

Unlike conventional forwarding algorithms that always send messages to a *given* destination node in \(V\), DT-GHF sends messages to a destination location \(t\) in \(S\) that is produced by the message source’s hash function (see Step 2 of DT-GHF in Figure 7). Because the probability that any node in \(V\) is exactly on location \(t\) is close to zero, messages are delivered to an *eventual* destination node that is closest to \(t\) in \(V\) at a forwarding time. This *late-binding* of destination location-to-node inherently supports reliability of message delivery under system failures, e.g., when node \(v\) currently closest to \(t\) is failed, messages destined to \(t\) can be delivered to node \(w\).

¹⁰For a point \(c \in S\), the set of all points closer to point \(c\) than to any other point of \(S\) is the interior of a (in some cases unbounded) convex polytope called Voronoi cell for \(c\). The set of such Voronoi cells is the Voronoi tessellation corresponding to \(S\). A DT graph for a given set \(V\) is the dual graph of Voronoi tessellation for \(V\) as shown in Figure 5 (a).
newly closest to \( t \) without any extra treatment (self-healing).

Each message is first sent towards \( t \) using a refined version of the greedy forwarding scheme (see COMPUTE_GF in Figure 7): Under normal operation from a current node \( v_i \), data is sent to one of \( v_i \)'s neighbors closer to \( t \) than \( v_i \), rather than the closest to \( t \) among \( v_i \)'s neighbors. This refined scheme improves communication efficiency through exploiting multiple forwarding candidate neighbors. However, as a result of SeDAX node failures, this forwarding scheme could reach a local minimum node \( u \) in a non-DT graph. At this time, the DT-GHF must determine whether or not \( u \) is a destination node for forwarding messages. Namely, the DT-GHF checks whether \( u \) is the closest node to \( t \) in \( V \). If either of the following conditions hold: (1) \( u \)'s Voronoi cell contains \( t \) or (2) recovery forwarding from \( u \) returns to \( u \), \( u \) is declared to be the closest node to \( t \) in \( V \) (see CLOSEST_NODE in Figure 7). This Voronoi cell-based resolution can minimize the impact of recovery forwarding as explained below.

DT-GHF’s recovery forwarding is a right-hand walk on the face that is incident to \( u \) and intersects line segment \( \overline{ut} \) (see COMPUTE_RHW in Figure 7). We borrow this idea of the right-hand walk from GPSR [23], but the use of Voronoi cells makes our right-hand walk scheme have lower communication overhead and adds resilience under network dynamics. If a SeDAX node closer to \( t \) than \( u \) is found during the right-hand walk, the walk falls back to the greedy forwarding routine. DT-GHF leverages Theorem 3 and Observation 2 for the use of combined greedy forwarding with the right-hand scheme.

1) Resilience under normal operation: Under normal operation we assume that (1) each SeDAX node’s location does not change and (2) no new edge is added to the face which is being currently toured by the DT-GHF’s right-hand walk. The DT-GHF’s forwarding resilience stems from the following theorems (whose detailed proofs are provided in Appendix): Let \( T = (V, E) \) be a given DT graph, \( v_0 \in V \) be a source SeDAX node, and \( s \in S \) be a destination location.

**Theorem 4:** In \( T \), DT-GHF can always establish a greedy forwarding path from \( v_0 \) to the SeDAX node closest to \( t \).

**Theorem 5:** In a connected subgraph \( T^- \subseteq (V^- \subseteq V, E^- \subseteq E) \) of \( T \), DT-GHF always converges to the SeDAX node closest to \( t \) in \( V^- \).

2) Resilience under network dynamics: We now consider message delivery in transient-state network graphs that are introduced by SeDAX nodes join, failure, and restoration operations. Under these network dynamics, forwarding loops could arise only due to the right-hand walk since greedy forwarding never creates any forwarding loops. However, forwarding loops can be easily broken using our loop-escaping scheme. For example, adding a new edge during a right-hand walk may make the right-hand walk create a forwarding loop. The loop eventually walks onto a triangle on our DT construction. Then, DT-GHF can escape the forwarding loop by using the line between current forwarding node and destination location \( t \). Note that the details of the loop-escaping scheme (see DETECT_LOOP in Figure 7) is omitted here for simplicity.

### Algorithm DT-GHF(\( v_i, t, l(u) \))

1. if \( i = 0 \)
2. then \( t \leftarrow \text{geographic_hash_output} \)
3. \( l(u) \leftarrow \text{NULL} \)
4. if \( \text{CLOSEST_NODE}(v_i, t, l(u), l(v_i-1)) \)
5. then stop forwarding at \( v_i \)
6. \( v_{i+1} \leftarrow \text{COMPUTE_GF}(v_i, t, l(u)) \)
7. if \( v_{i+1} \neq \text{NULL} \)
8. then \( l(u) \leftarrow l(u) \)
9. else if \( l(u) = \text{NULL} \)
10. then \( l(u) \leftarrow l(u) \)
11. \( v_{i+1} \leftarrow \text{COMPUTE_RHW}(v_i, t) \)
12. else \( v_{i+1} \leftarrow \text{COMPUTE_RHW}(v_i, l(v_i-1)) \)
13. \( v_{i+1} \leftarrow \text{DETECT_LOOP}(v_i, v_i-1, v_{i+1}, l(u)) \)
14. continue forwarding to \( v_{i+1} \)

**PROC COMPUTE_GF(\( v_i, t, l(u) \))**

1. if \( l(u) = \text{NULL} \)
2. then \( l_i \leftarrow l(v_i) \)
3. \( i_i \leftarrow i(v_i) \)
4. \( l_i \leftarrow l(v_i) \)
5. \( i_i \leftarrow i(v_i) \)
6. for all \( x \in N(v_i) \)
7. if \( N_g(v_i) \neq \emptyset \)
8. then return a neighboring node in \( N_g(v_i) \)
9. else return \( \text{NULL} \)

**PROC COMPUTE_RHW(\( v_i, q \))**

1. \( \theta = 2\pi / r \)
2. for all \( x \in N(v_i) \)
3. do \( \theta_x \leftarrow \text{angle}(v_i, l(x)) \)
4. if \( \theta_x < \theta \)
5. then \( \theta \leftarrow \theta_x \)
6. return \( r \)

**PROC CLOSEST_NODE(\( v_i, t, l(u), l(v_i-1) \))**

1. if \( \text{CLOSEST}(v_i) \) encloses \( t \) and \( l(u) = l(u) \)
2. \( \left( \text{COMPUTE_RHW}(v_i, l(v_i-1)) = \text{COMPUTE_RHW}(v_i, t) \right) \)
3. then return true; # verify the nearestness to \( t \)
4. return false;

**PROC DETECT_LOOP(\( v_i, v_i-1, v_{i+1}, l(u) \))**

1. if \( v_i, v_i-1, \) and \( v_{i+1} \) constitute one DT triangle
2. then \( \text{edge } v_i v_{i+1} \) crosses line \( v_i v_i-1 \)
3. then \( v_{i+1} \leftarrow \text{COMPUTE_RHW}(v_i, l(v_i-1)) \)
4. if \( \text{edge } v_i v_{i+1} \) crosses line \( v_i v_i-1 \)
5. then \( v_{i+1} \leftarrow v_i-1 \)
6. \( l(u) \leftarrow l(u) \)
7. return \( v_{i+1} \)

Fig. 7. DT-GHF Pseudo-code.

**C. Cost-effective DT-based Replication**

Our SeDAX platform leverages Theorem 2 for its replication scheme. For presenting our replication scheme, we assume that a DT graph consists of at least three SeDAX nodes \( v, w, \) and \( y \) (see Figure 5 (a)). For a given topic-group \( g \), node \( v \) becomes a primary SeDAX node as it is the closest to location \( t \) as determined by the hash function. Using local coordinate space information, \( v \) can determine that \( w \) is the SeDAX node closest to location \( t \) among all of \( v \)'s neighbors. \( w \) is then designated as a secondary SeDAX node for topic-group \( g \) and so all data (either information of group participants in streaming-based access or application data in query-based access) for topic-group \( g \) is replicated to node \( w \).

In the event of failure of a SeDAX node \( v \) as shown in Figure 5 (b), secondary SeDAX node \( w \) for topic-group \( g \) is declared to be the new primary SeDAX node and it then
designates node \( y \) which is closest to \( t \) among all of \( w \)'s neighbors as the new secondary SeDAX node for topic-group \( g \). Later on when SeDAX node \( v \) appears again in the SeDAX network, it is restored to be the primary SeDAX node for topic-group \( g \). \( w \) reverts back to be a secondary SeDAX node for topic-group \( g \) and it then requests the existing secondary node \( y \) to remove \( g \)'s data.

V. SeDAX DESIGN AND PROTOTYPE IMPLEMENTATION

We have built a prototype implementation of the SeDAX platform (node and middleware) using the \( C++ \) language and the CentOS operating system which is a Linux distribution. Due to portability issues, we have implemented SeDAX systems in the user space instead of kernel space and so POSIX TCP socket and multi-threaded programming interfaces have been used for inter-communication either between SeDAX node and middleware or among SeDAX nodes. Figure 8 and Figure 9 show the schematic for the prototype SeDAX platform and the simple Smart Grid AMI application implemented on the SeDAX platform. The application was built using C# and the Windows operating system, together with other \( C++ \) modules and CentOS capabilities.

A. Publisher-Subscriber Communication Interfaces

The SeDAX middleware supports the following three communication primitives for applications (shown in Figure 8):

- \( \text{open}(\text{topic}, \text{role}) \)
  - For a topic-group \( g \), when an application calls either \( \text{open}(g, \text{PUBLISHER}) \) or \( \text{open}(g, \text{SUBSCRIBER}) \), either a publisher object or a subscriber object is created within the SeDAX middleware. If either object successfully joins a topic-group in a SeDAX network, it returns a pointer to the object to the calling application. Otherwise, it returns a NULL.

- \( \text{write}(\text{data}) \)
  - Using \( \text{write}() \), an application can pass its own data to a publisher object in the SeDAX middleware which will then encrypt the data and pushes the encrypted data to the SeDAX network. However, if the \( \text{write}() \) primitive is called on a subscriber object or on a nonexistent object, it is immediately returned with an error.

- \( \text{read}(\text{data}) \)
  - Using \( \text{read}() \) operation, an application can read data from a subscriber object in the SeDAX middleware which then receives data from the SeDAX network. For streaming data access, whenever data from a SeDAX network arrives at the subscriber object, the object calls the \( \text{read}() \) primitive to notify its associated application of the arrival of new data. For query-based data retrieval, an application calls the \( \text{read}() \) primitive to request the subscriber object to pull data from a SeDAX network. Then, it waits for a while until the subscriber object obtains the requested data.

B. Internals of SeDAX Operations

1) Topic-group, Hashing, and Multi-hop Forwarding: Recall that a topic is defined by data type, location, and time. As shown in Figure 8, when a topic specified by an application is passed to either a publisher object or a subscriber object, it is input to a geographic hash function which outputs coordinates for the location \( t \) on a given Euclidean space. A group-join message destined to location \( t \) is then sent to a SeDAX node associated with it at system initialization time. The SeDAX node forwards the message over a DT-based SeDAX network using our multi-hop forwarding algorithm, DT-GHF. When the message reaches the SeDAX node \( v \) currently closest to location \( t \), a topic-group manager in \( v \) receives the message and then establishes a secure session to the message source’s security client. The security client exchanges authentication messages with a security server using the authentication procedures described in Section V-B4. The communications with the security server occurs over a secure session. Authentication messages exchanged via the topic-group manager can neither be modified nor forged as the security client’s credentials are safely managed by security servers.

After the topic-group manager in the SeDAX node \( v \) is notified of successful authentication from the security server, the secure session to the security client is terminated and the topic-group manager reserves memory resources for either streaming-based or query-based data dissemination/storage. The security server is the entity that determines whether a streaming service or a storage/query service is appropriate for a given topic-group. The above process is initiated by a Smart Grid application on a per topic of interest.

2) Node Bootstrap, Coordinates, and Discovery: We assume that all SeDAX nodes have their own public-private key pairs and are associated with a list of bootstrap servers in a SeDAX network. If there are existing SeDAX nodes in the network, they can function as bootstrap servers for new SeDAX nodes. Whenever a new SeDAX node is booted-
up, it has to be authenticated by a security server via the bootstrap server. The bootstrapping task is considered a special topic-group and can be handled by the authentication process described in Section V-B4. Newly authenticated SeDAX nodes can establish edges (TCP connections) to their neighboring SeDAX nodes using a DT construction process. Each node needs its own coordinates for SeDAX operations. A network coordinate system [43] is required to embed network latency information. Thus, whenever a SeDAX node joins a SeDAX network, its coordinates are assigned by a security server. This assignment is made based on the measurement of latency to existing nodes in the coordinate system. This measurement task is not cumbersome in low-churn Smart Grid communications where node join/leave events occur infrequently.

During system loading time, a SeDAX middleware must discover SeDAX nodes either on the subnet where it is situated or on other subnets. After the discovery process is completed, the SeDAX middleware maintains the associations with these SeDAX nodes. For scalability, SeDAX nodes themselves do not keep the state of the SeDAX middleware.

3) Distributed DT Construction: A DT graph can be built in a distributed and incremental manner by the flipping algorithm described in [44] or using the candidate-set approach in [45]. In the flipping algorithm, once a joining node is led to a closest node in an existing Delaunay triangle, the triangle enclosing the joining node is divided into new triangles. The new triangles are adjusted by flipping edges if the triangles do not meet the DT property: e.g., for two triangles △ABD and △CBD with common edge BD, if the sum of two angles ∠BAD and ∠BCD is more than 180° (namely, the triangles do not meet the Delaunay properties), switching common edge BD for common edge AC produces two new triangles △ABC and △ADC that meet the Delaunay property. In the candidate-set approach the joining node computes its local DT graph based on its own candidate set, and updates its candidate set by making contacts with any new neighbors. The beauty of distributed DT construction is that it is perfectly scalable to network size. The operations for node join/leave/failure-recovery can be done within O(1) operations. For example, if the flipping algorithm is used for DT construction on a 2-dimensional space, only \( \frac{2}{3} \) operations are needed for node join in the worst case, and \( O(k \log k) \) operations are needed for node-leaving/failure-recovery, where \( k \) is the number of neighbors of the node on DT [44]. By Observation 1, the average number of neighbors on DT is bounded by 6, so the complexity of join/leave operation is a constant, whereas most other DHT algorithms such as Pastry [20] require \( O(\log N) \) operations on average where \( N \) is the number of nodes in the system. For robust but simple implementation, our join and leave processes need \( O(k) \) and \( O(k^2) \) operations respectively.

4) Authentication and Confidentiality: The cluster of security servers constituting a trusted network provides authentications for both new SeDAX host nodes that are joining a DT network of SeDAX nodes and the middleware of new security clients that are accessing a topic-group.

SeDAX node authentication is based on public key certifications (X.509 and certificate authority) and is necessary for preventing man-in-the-middle attacks. So, all SeDAX nodes have their own public-private key pairs as well as a preconfigured public key certificate of the trusted network. A SeDAX node which is about to join a DT network has to be authenticated by one of the security servers protecting the DT network. Mutual-authentication messages are exchanged between the SeDAX node and an arbitrary security server. The procedure for authentication is similar to that of client-authenticated TLS handshake [38]. If the node passes the authentication procedure, it can obtain its own public key certificate signed by the security server. As the node establishes an edge with an existing node in the DT network, the two nodes authenticate each other through exchanging their own public key certificates since they can verify each other’s public key certificate using the trusted network’s public key certificate.

On the other hand, for topic-group authentication, authentication messages between a security client of the SeDAX middleware and an arbitrary security server are encrypted using a symmetric key rather than a public key in order to support security clients running over low-power hardware platforms. The detailed scheme of the symmetric key based authentication is similar to that of SSTP\(^{11}\) described in recent literature [10]. After a security client is authenticated, a security server determines the client’s access permission to a topic data through checking the client’s preprovisioned privileges. For e.g., no sensor/meter is allowed to participate in a “data-collection” group as a subscriber member. Given a topic-group \( g \), a security server manages one group master key and one session key generator derived from the master key. For all subscriber clients associated with the topic-group \( g \), the session key generator is assigned. In contrast, for each publisher client associated with the topic-group \( g \), one session key is assigned which is created by the session key generator. Note that each group master key is created not on a per client basis but on a per topic-group. This allows scalable access control over the topics where the number of topics is less than the number of clients. Data encrypted by the publishers with a session key and AES [46] can be decrypted only by subscribers who can extract the session key (E2E confidentiality). For a group, even when a publisher’s session key is exposed to the adversary, the approach described above can preserve the confidentiality of the communications of other publishers. Moreover, it does not need session key updates each time a new member joins the group.

We emphasize that all credentials for topic-groups are managed only by the security servers in the trusted network. This requirement is necessary due to the significant security concerns in the mission critical environment of the power grid.

\(^{11}\)Our security server can efficiently store a large number of security client’s long-term keys. Our scheme is to have these keys not be truly random, but, rather, pseudo-randomly generated from the server’s master key \( k \). That is, given a server’s master key \( k \), for a client with identity \( id \), we set its long-term key \( k_{id} = AES_{k}(id) \). Each client with identity \( id \) is then provisioned, e.g., at the time of manufacture, with \( k_{id} \). The server need not store this key, as it can readily generate it, given it’s master key and the client’s identity.
VI. CLOUD-BASED DEMAND RESPONSE IMPLEMENTED USING THE SeDAX PLATFORM

We now illustrate how a novel Smart Grid application such as Demand Response (DR) can be deployed on the SeDAX platform. The DR we consider here is a price-based demand control mechanism to be used at times when there is a power deficit. Specifically, we propose the concept of cloud-based demand response (CDR), a cloud service provided by SeDAX on the consumer side (see [47] for details). Demand response we consider is an optimization process for computing the appropriate amount of power reduction per customer and the associated incentive (i.e., pay-back) price. Two main requirements on demand response are security and scalability. To meet these requirements, CDR exploits the topic-based, secure and scalable group communication aspects of SeDAX.

Fig. 10 shows the topic-based group communication for demand response. Initially, customers who want to participate in the demand response application are authenticated by the secure control server. A meter-data collection group is responsible for collecting power consumption data. Based on these measurements the current demand is calculated. Once the demand is projected to exceed power supply capacity, the demand response application is automatically invoked. This application provides an incentive price for power reduction and this information is multicast throughout a topic-group, called the updating group. The participating customers respond with their power reduction bid through the bidding group. Price updating and bidding processes are done iteratively until the optimal equilibrium is found. From the utility’s perspective, CDR appears as a black box information system that takes an input (power deficit) from the utility and gives an output (the optimal incentive price and the power reduction on a per customer basis). All computations are done within the SeDAX network, and the utility can deploy demand response as a cloud service. The operational aspects of CDR are similar to the open automated demand response (OpenADR) standard, and thus SeDAX can be easily used to provide the OpenADR service (http://openadr.lbl.gov).

VII. SeDAX: PERFORMANCE EVALUATION

The strength of data-centric and publisher-subscriber communication concepts have been clearly illustrated in prior literature [12], [13], [7]. In this section, we focus on the evaluation of a key functional block of SeDAX, namely, DT-GHF combined with DT-based replication scheme. This functional block provides scalability and inherent resilience for data-centric communication. We compare the proposed DT-GHF scheme with two other geometric-based alternatives GHT [8] and CAN [22]. As shown in Table I, the basis for all three algorithms discussed in this section is that the greedy forwarding (GF) algorithm [29] is used as the default message delivery algorithm. In all three approaches, the closest node to a destination location is determined by a geographic hash function. The implementation of these algorithms differs from the basic greedy forwarding in terms of methods for recovery from local minima. The major differences of DT-GHF, GHT and CAN are as follows. GHT uses a complicated face-routing scheme [23] combined with right-hand walks. It assumes a plane graph such as a Gabriel graph [48]. CAN uses TTL (Time-To-Live) based broadcast to recover from a local minimum node and the basic connectivity among CAN nodes is irrelevant to physical proximity. For node association, each CAN node is responsible for its own portion (called zone) of the entire CAN space. Similarly, recall that each SeDAX node has its own Voronoi cell in the Voronoi tessellation. In GHT, the node association is for a given point in space.

Our SeDAX platform is expected to support a wide range of Smart Grid devices including some resource constrained sensors. Furthermore, reliability and latency of forwarding are important criteria for Smart Grid applications. For this reason, we do not explicitly compare the performance of DT-GHF against a different class of DHT algorithms such as Pastry which require relatively-large route tables O(log N) and incur relatively-high latency, despite having fewer routing hops [49].

A. Comparison on Deployment Considerations

Table II summarizes the differences between the algorithms in terms of resource overhead, reliability of data delivery and latency which are crucial to smart grid platform requirements.

1) Route Table Size: The route table size of an algorithm depends on a network’s average node degree. It is an important metric to understand a given forwarding engine’s complexity and memory cost in terms of scalability. As shown in Table II, for all compared algorithms, the average number of neighbors is quite small and independent of the total number of nodes in a network. This is because the average number of neighbors in a DT graph is less than six (Observation 1), the number of edges in a plane graph with vertex set \( V \) is less than or equal to that in a DT graph with vertex set \( V \) [24], and CAN’s average node degree is known to be \( O(2d) \) where \( d \) is the dimension of the given space [22]. Hence, without complex data-structures for fast look-up, all compared algorithms can quickly and simply find the next-hop from the route table of each node visited during the forwarding operation.

2) Message Overhead: The extra information (the message header) required for forwarding is referred to as message
overhead since it consumes a portion of the available communication bandwidth. In the Smart Grid domain, meters and sensors have been known to generate small-sized data, e.g., 50Byte - 500Byte. Thus, forwarding algorithms with large message overhead are obviously inappropriate for Smart Grid in terms of their bandwidth consumption. Note that the measure of message overhead excludes portions of message authentication codes such as SHA-2 mostly employed for message integrity. In GF the message overhead is just one field to carry destination location information. In DT-GHF, in addition to the field for greedy forwarding, there is one extra field containing the location of the current local minimum. This additional overhead is needed for right-hand forwarding. In contrast, GHT has five times the overhead of GF due to its complicated recovery scheme, as shown in [50]. In CAN, by default only the destination location field is used. However, most DHTs like CAN in general use either 128-bit or 160-bit key space which is the output range of a hash function. The messaging overhead for CAN in the recovery mode (extended ring search) is unknown.

3) Message Delivery Performance: Recall that on a DT graph and its connected sub-graphs, DT-GHF shows perfect message delivery (Theorem 4). Greedy forwarding is sufficient for message delivery in a steady-state network (DT graph) and right-hand forwarding is used only in the face of node failures.

In a steady-state network, CAN shows perfect message delivery [22]. In the presence of node failures, CAN forwarding may visit a local minimum node to a destination location and then uses TTL-based broadcast to escape from the local minimum. However, we cannot ensure that the TTL-based broadcast finds a zone enclosing the destination location with limited cost (communication and memory resources).

On a steady-state and connected plane graph, GHT can provably deliver messages to a node on a destination location using greedy forwarding with recovery [24]. However, it is possible that a node may not exist at a location produced by the geographic hash function. This means that recovery forwarding is performed at least once for correct message delivery since GHT has no way to determine whether or not the current local minimum to a location t is the global minimum to t. Thus, even on a DT graph, GHT shows longer paths than DT-GHF for the same source-destination pair.

4) Replication Cost: We now assume for simplicity that all compared algorithms use a single-output hash function, even though they could all employ multiple-output hash function for improving data availability (as described in Section VIII-C). We also limit our discussion to a two dimensional space. For this comparison, t is g’s topic and t is g’s hashed location in a given coordinate space.

CAN replicates g’s data to all neighbors of the CAN node closest to t. As a result, it costs at least 4 (=2 × 2) additional communication and storage costs for data replication since each CAN node has more than 4 (=2 × 2) neighboring CAN nodes. GHT has to replicate g’s data to all nodes on the face enclosing t. The number of nodes on the face of a plane graph is variable and may sometimes be extremely high. In an extreme case that a plane graph with N nodes has just one face, N nodes become replicas for any topic and GHT has N - 1 hops of communication cost per replication. Unlike CAN and GHT, DT-GHF replicates g’s data or state to just one more node in a SeDAX network which is a neighbor of the primary SeDAX node and is also the second closest node to t, as described in Section IV-C. As a result, there is only one extra communication and storage expense required for SeDAX data replication. Thus DT-GHF has inherent scalability benefit when the amount of data to be replicated increases.

B. Simulation-based Comparison

The simulation studies presented here provide a performance comparison of the algorithms in terms of route table size, delivery ratio, data access latency, and costs. The route table size is an important metric to determine the computational requirements of Smart Grid nodes. The delivery ratio and access latency measures indicate the reliability and availability metrics of the Smart Grid network. The following simulation results show that DT-GHF significantly outperforms both GHT and CAN in terms of most of these performance metrics over different topology settings.

1) Experimental Settings: We incorporate a network coordinate system which is a virtual positioning scheme in a Euclidean space to embed network-latency information. In the chosen network coordinates, the Euclidean distance between any pair of nodes approximates actual network latency i.e. data access time. For realistic scenarios, we used real-world data made available by Harvard Network Coordinates Research Group [51]. They have measured delivery latencies among 226 nodes in PlanetLab for four hours. To transform measured latencies to the two-dimensional network coordinates space, we use the Vivaldi [43] simulator from the same group since that algorithm is fully distributed and simple to implement. In addition to the real-world data-set, we generated synthetic data-sets for large scale experiments. In this data-set, nodes are
uniformly distributed in a two-dimensional coordinate space. We tested different network sizes by varying the number of nodes as follows: 500, 1000, 2000, 5000, 10000. Each of the results described is averaged over five experimental instances.

Given the two-dimensional coordinates data-set, we first create three kinds of graphs for comparison. For each node set \( V \), a DT graph \((V, E)\) and a CAN graph are generated respectively. We can also extract a Gabriel graph (GG) \((V, E^-)\) for GHT from the DT graph since a GG is a sub-graph \((V, E^- \subseteq E)\) of DT. For the generated graphs, we then compare the performance of DT-GHF against the alternatives by measuring route table size, delivery ratio, data access latency, and communication/storage cost. Based on the previous discussion the basic CAN is expected to show extremely-low performance and fairly-high cost compared with other alternatives. This is because its edge relationship is irrelevant to physical proximity among nodes. Thus, to make a conservative and fair comparison the topology-sensitive CAN is used. The other optimization techniques to improve CAN’s performance are not implemented for simplicity.\(^{12}\)

2) Route Table Size: We first show that using a DT graph, a GG, or a CAN graph that it is possible to build a scalable routing scheme in terms of routing state with respect to the number of nodes in the network. Figure 11 shows the number of states per node as a function of network size. As expected, all compared graphs keep constant state (less than 6) regardless of the number of nodes in the system. It is also confirmed in Table III which shows the results on the realistic data-set.

We measured the node degree distribution of the three kinds of graphs constructed from 10,000 randomly placed nodes for each iteration. Figure 10 shows the cumulative distribution function for the number of neighbors through 10 iterative computations. We see that for DT, the probability that a certain node has less than 12 neighbors is more than 0.99, and approaches 1 for 14 neighbors. This implies that the memory size of route table would be quite small.

3) Delivery Ratio: Table IV shows the delivery ratio for the compared schemes as a function of the network size over steady-state networks. Delivery ratio measures the fraction of successful message delivery operations between all source node \((s_i)\) and destination location \((l_j)\) pairs and is defined as:

\[
\frac{1}{N_L} \sum_{l_j=1}^{L} \sum_{i=1}^{N} f(s_i, l_j),
\]

where \( N \) is the number of nodes in a graph, \( L \) is the number of locations produced by a uniform hash function, and \( f(s_i, l_j) \) is either 0 or 1. If messages from node \( s_i \) reach node \( l_j \) which is closest to location \( l_j \), \( f(s_i, l_j) \) is 1, and otherwise 0. For all measured data, we see that CAN, DT-GHF, and GF over the DT have perfect delivery ratio. In contrast, GHT and GF over the Gabriel graph show delivery failures. Between these two schemes, GHT’s delivery failure is small and occurs only at large network sizes.\(^{13}\) The delivery ratio of GF over the Gabriel graph is found to deteriorate logarithmically with network size. Therefore we can clearly see that the perfect delivery of GF is not simply dependent on the network size but depends on the specific properties of the underlying graphs (Theorem 4). Besides, GF’s results imply that GHT over the Gabriel graph has to frequently employ its expensive recovery scheme to escape local minima visited during greedy forwarding and so will naturally show lower performance and higher communication cost.

4) Resilience to node failure: We now measure delivery ratio in the case of node failures. GF and GHT are excluded in this measurement since they have incurred delivery failures even under normal operations. We simulate node failure as follows: For a given network graph \( T = (V, E) \), a certain number of nodes are randomly removed from \( V \) and also their incident edges are removed from \( E \). If the resulting subgraph of \( T \) is disconnected, it discarded and the above process repeated until a connected subgraph is obtained. Figures 13 shows delivery ratio for DT-GHF and CAN as a function of proportion of failed nodes in a 1000-node network (generated from the synthetic data-sets). The measured proportions are 1%, 2%, 4%, 8%, and 16%. CAN’s delivery failures sharply increase with node failures since CAN does not have an explicit recovery node operation from a local minimum. By comparison, DT-GHF works perfectly, independent of node failures, which confirms Theorem 5.

5) Data Access Latency: Figures 14 and 15 show normalized data access latency for the compared algorithms as a function of the network size. The metric primarily captures network latency between any given source-destination pair. It does not account for message processing time such

\(^{12}\)The topology-sensitive CAN and its optimized schemes are described in Section 3 of [22]. The fully-optimized CAN may show similar performance to DT-GHF, but has significantly higher implementation cost and complexity.

\(^{13}\)GHT exhibited an interesting and pathological behavior, unlike the literature description [22]. For a topic and its hashed location \( t \), messages from different sources did not always reach a node closest to \( t \) but instead nodes on the face enclosing \( t \).
as encryption, queuing, and decryption. These delays are quite small relative to network latency: e.g., about a few of microseconds per message for encryption/decryption using our AES implementation. The queuing delays are negligible given a high performance platform for SeDAX nodes. The read operation is in general faster than write operation for all compared algorithms due to the multiple data replicas deployed to improve availability.

**Data-write:** Each write operation to a topic is completed when the operation’s data is delivered to the node closest to the topic’s hashed location, as illustrated in Figure 4. We use both the average and maximum access latency measures: \( \frac{1}{|P|} \sum_{(i,j) \in P} d(s_i, t_j) = d(s_i, t_j) \) and \( \max_{(s_i, t_j) \in P} p(s_i, t_j) = d(s_i, t_j) \) where \( P \) is a set of source destination pairs with a path in a graph, \( p(s_i, t_j) \) is the length of path successfully established from node \( s_i \) to node \( t_j \) closest to location \( l_j \) and \( d(s_i, t_j) \) is the Euclidean distance between two nodes \( s_i \) and \( t_j \). For both average and maximum latency, DT-GHF clearly outperforms all compared alternatives. Interestingly, it incurs 2–3 orders of magnitude less maximum write latency than CAN and GHT for all measured topology settings.

**Data-read:** For topic-groups using query-based data access, there is a clear performance difference among compared algorithms. This is because queries from subscribers to the SeDAX node can be delivered along significantly different paths. On the contrary, data delivery time from a SeDAX node providing data to subscribers is not differentiated among compared algorithms since the data is delivered over direct communication channels between the SeDAX node and query sources, as illustrated in Figure 4. Also, we note that the read performance for streaming-based topic-groups is similar across the compared algorithms since query processing is not used in streaming-based data access. Thus, the difference in read-latency only corresponds to differences in latency measure of the query-delivery process. For a topic \( g \) and its hashed location \( l \), all queries to topic \( g \) do not necessarily reach the node closest to location \( l \). As illustrated in Figure 4, data can be retrieved when a query reaches topic \( g \)’s replica.

So, read-latency is measured by counting the path length from query source to one of \( g \)’s replicas (including the node closest to \( l \)). Over all topology settings considered (Table III and Figures 14 and 15), DT-GHF clearly outperforms CAN, whereas GHT shows similar average read latency to DT-GHF. Furthermore, DT-GHF exhibits write latency similar to read latency as compared with the other alternatives for which the read operation is clearly faster than the write operation.

**6) Topic Scale:** We have so far compared the three algorithms as a function of network-size with a fixed number of topics (1024 topics). To study topic scalability, we vary the number of topics (4 to 1024 topics) and compare DT-GHF with alternatives over networks with a fixed number of nodes.

Figures 16 and 17 plot the average normalized access latency and hop counts of compared algorithms over the 2000-node network. DT-GHF clearly shows constant results for latency and hop counts independent of the number of topics. Interestingly, the performance measures of GHT’s write operation linearly increases with the decrease in the number of topics. On the other hand, GHT’s read operations show performance close to DT-GHF. This is due to the fact that GHT topology is controlled by the number of topics. From this result we can say that, GHT is not well suited for Smart Grid applications since it exhibits poor data-access speed and high communication cost over a large-scale network where the number of topics is smaller than the number of nodes in the network. In Smart Grid applications typically the number of topics is much smaller than the number of nodes.

**7) Communication Cost for Data Access:** Communication cost for both data-write and data-read is measured as the hop counts in the path established between data/query source and the node where data is to be stored or retrieved. The higher the hop counts, the greater the bandwidth consumption. We measure the average hop counts of all compared algorithms over all measured topology settings (Table III and Figure 18). As expected, DT-GHF has smaller communication cost than the other alternatives for both data-write and data-read, even though analytically the average hop counts of all compared algorithms is \( O(\sqrt{N}) \) on a two-dimensional space.

**8) Communication and Storage Cost for Replication:** Data replication clearly improves data availability and also enables faster read operations. However, there is a trade-off between the benefits and resource consumption for communication and storage. As shown in Table III, in general, the more replicas

---

**Table IV**

<table>
<thead>
<tr>
<th>Network Size</th>
<th>DT-GHF</th>
<th>GHT</th>
<th>CAN</th>
<th>GF/DG</th>
<th>GF/GG</th>
</tr>
</thead>
<tbody>
<tr>
<td>500-nodes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.8775</td>
</tr>
<tr>
<td>1000-nodes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.8723</td>
</tr>
<tr>
<td>2000-nodes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.8207</td>
</tr>
<tr>
<td>5000-nodes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.7704</td>
</tr>
<tr>
<td>10000-nodes</td>
<td>0.9999</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.7215</td>
</tr>
</tbody>
</table>

---

Fig. 14. Average Normalized Access Latency.  
Fig. 15. Maximum Normalized Access Latency.  
Fig. 16. Access Latency for Topic-scale.
we have, the faster the data access but greater the consumption of communication and storage resources. Figure 19 shows relative replication cost of CAN and GHT against DT-GHF as a function of the network size. The cost is represented as a factor of the communication and storage resources consumed by DT-GHF replication. DT-GHF shows a constant replication cost as it by design has just two replicas: the primary SeDAX node and the secondary SeDAX node. As observed in Figures 11 and 19, CAN’s replication cost is proportional to the average number of neighboring nodes. Interestingly, over all measured settings, we find that GHT cannot outperform DT-GHF in terms of read performance even though it pays at least ten times the replication cost of DT-GHF (Figures 14 and 19).

VIII. DISCUSSION

In the previous sections we have shown the cost effectiveness, scalability and resilience features of SeDAX. We now discuss some extensions to the SeDAX platform that illustrates support for scalable security and improved data availability.

A. The Data-plane of SeDAX Platform

Our discussions have thus far focused on the control-plane components of the SeDAX platform. We now briefly describe the data-plane operation of the SeDAX platform. As shown in Figure 8, each SeDAX node has data aggregator/multicaster or storage that constitute the data-plane components of the SeDAX platform. These components can be generally called rendezvous points. A topic-group manager (a control-plane component of the SeDAX platform) creates and manages one rendezvous point per topic-group which represents one instance of a message router or storage. For a streaming topic-group \( g_s \), its corresponding rendezvous point \( R_s \) aggregates data from the publishers of \( g_s \) and immediately multicasts the data to the subscribers of \( g_s \). For resilient streaming, information regarding the subscribers of \( g_s \) is stored in \( R_s \) and replicated to a neighboring node of \( v \) which hosts \( R_v \).

For a query topic-group \( g_q \), its rendezvous point \( R_q \) stores data from the publishers of \( g_q \) and responds to queries from the subscribers of \( g_q \). For improved data availability, the data is replicated to a neighboring node of \( v \) which hosts \( R_v \). For any topic-group \( g \), its rendezvous point drops data or queries from any unauthorized entities trying to access topic-group \( g \).

B. Per-node Security vs Per-topic Security

Similar to other secure systems, SeDAX uses the well-known RSA [52] and AES [46] cryptography for authentication and message encryption as shown in Section V-B4. The distinguishing feature of SeDAX is the use of the topic-group concept that enables scalable key management and the preservation of end-to-end security.

Consider the AMI infrastructure consisting of million meters \((N)\) and a small number of meter-data collectors. In the per-node security model employed in most security frameworks, the two communicating entities (client and server) first mutually authenticate each other and then directly communicate with each other using one session key established during authentication. Thus, under per-node security, a meter-data collector (or its associated security server) has to maintain \( N \) credentials for per-meter authentication and also maintain \( N \) session keys for enabling secure data-readings from all meters.

However, under the secure topic-group idea used by SeDAX, a meter-data collector does not directly communicate with each meter. The AMI infrastructure can be implemented using group master keys per-topic rather than per-node. As the number of topics (location, time, and data type) is independent of the number of meters and in general quite small, topic-based security schemes have the benefit of scalable key management.

C. Improved Data Availability through Load Balancing and Multiple-output Hash Function

Data availability is a feature that captures the requirement of guaranteed data access. Improving the data access reliability can be accomplished both by local load-balancing of the topic groups and by geo-diverse redundancies.

Local load balancing of topic-groups eliminates hotspots and performance bottlenecks. Note that in Figure 4, all subscribers in a topic-group get data through a single SeDAX node, thus making it a single point of bottleneck. This performance problem can be avoided in SeDAX by load-balancing the node association of topic-groups using a multiple-output hash function tuned for load distribution. Using the same mechanism of multiple-output hash functions, the SeDAX platform can be extended to support geographical resilience. The resulting hash locations can be made to be geo-diverse and thus avoid data availability issues even in the face of natural disasters such as the recent hurricanes and earth quakes.

IX. CONCLUSION

Our proposed SeDAX platform equipped with DT-GHF algorithm and DT-based replication scheme provides the necessary scalable, resilient, and secure communications infrastruc-
ture to support various Smart Grid applications. Our prototype implementation of this platform validates the performance results that we have shown in our evaluation studies. Through the illustrative example of cloud-based demand response, we have shown the relevance and flexibility of data-centric platforms for Smart Grid communications. The topic group concept employed by the SeDAX enables scalable security implementation, especially for group communications. In addition to Smart Grid systems spanning across electricity consumers and utilities, the SeDAX platform can be seamlessly employed for wide-area measurement and control in continental-scale transmission grids as well, as was recently shown in [53].

REFERENCES

and edge set $E$, $v \in V$ be a vertex with neighbors $N(v) \subseteq V$, and $t \in S$ be a (target) point or location. Closeness of distance between two points in $S$ is measured by the Euclidean metric. We will assume that in any DT graph, no four points are co-circular. This is not a limitation for our application since (1) the probability of this occurring in practice is zero almost surely, and (2) if this situation were to occur, location coordinates could simply be perturbed infinitesimally to avoid unnecessary technical complications.

A. Proof of Theorem 1

**Proof:** The Voronoi diagram for $V$ is the geometric dual of the Delaunay graph associated with $T$. Consider the Voronoi cell $c_v$ of $v$. Every Voronoi edge of $c_v$ corresponds to a neighbor of $v$ and, conversely, every neighbor of $v$ corresponds to an edge in $c_v$ since no four vertices of $V$ are co-circular. Each edge of $c_v$ is a line segment contained in the perpendicular bisector of the line segment $\overline{tv}$ between $v$ and some $u \in N(v)$ which subdivides the plane into open half planes $h_u \equiv \{ z : |\overline{tv}| < |\overline{zu}| \}$ and $h_u \equiv \{ z : |\overline{tv}| > |\overline{zu}| \}$ (see Figure 20).

Since $v$ is closer to $t$ than any $u \in N(v)$, then $t$ must belong to the intersection of all $v$-containing open half planes defined by perpendicular bisectors of line segments $\overline{tv}$ for all $u \in N(v)$. That is, $t \in c_v$ and so the theorem holds.

B. Proof of Theorem 2

**Proof:** Let $v \in V$ be the closest vertex to location $t$. Then location $t \in c_v$, the Voronoi cell for $v$. Since the Voronoi diagram Vor($V$) of vertex set $V$ is the geometric dual of the associated Delaunay graph of $T$, and since by assumption no four vertices in $V$ are co-circular, the Voronoi cells of vertices in $N(v)$ form a doubly connected region $D$ that surrounds $c_v$ with disjoint inner and outer boundaries, $\partial D_i$ and $\partial D_o$, respectively. Suppose $w \in V \setminus N(v) \cup \{ v \}$ is the second closest vertex to $t$. Since $w \not\in c_v \cup D$, then line segment $\overline{tw}$ must intersect $\partial D_o$ (and $\partial D_i$) at a Voronoi edge (either at a non-endpoint or at a Voronoi vertex). Suppose the intersected edge belongs to $c_w \in$ Vor($V$) for $w \in N(v)$. Then the perpendicular bisector of line segment $\overline{tv}$ partitions the plane into half planes $h_u \equiv \{ z : |\overline{tw}| \leq |\overline{wu}| \}$ and $h_u \equiv \{ z : |\overline{tw}| > |\overline{wu}| \}$.

Since $c_w$ is the intersection of all closed half planes containing $u$ and defined by the perpendicular bisectors of line segments between $u$ and all other nodes in $V$, then $c_w \subseteq h_u$. Also, $t \in h_u$ since $\overline{tw}$ intersects $c_w$. Therefore, $u$ is closer to $t$ than $w$ is, contrary to assumption. Hence, the theorem holds.

C. Proof of Theorem 3

Before showing the proof of Theorem 3, we first address a lemma required for Theorem 3.

**Lemma 1:** For a given circle $C$ centered with point $t$, if a triangle $\alpha$ meets the following conditions: (1) only $\alpha$'s one point $p_1$ is within $C$ and (2) line segment $\overline{p_2p_3}$ between $\alpha$’s other two points $p_2$ and $p_3$ intersects line segment $\overline{tp_1}$, $\alpha$’s circum-circle encloses the region which is the subregion of $C$ partitioned by $\overline{p_2p_3}$ and does not contain $p_1$.

**Proof:** Let $L$ be the line segment between two end points where $\overline{p_2p_3}$ intersects $C$. Then, we can draw a triangle $\beta$ made up of $p_1$ and $L$’s two end points, as illustrated in Figure 21. In $\beta$, the angle $\theta$ between two line segments incident to $p_1$ is more than $90^\circ$ since $p_1$ is within $C$. It follows that $\beta$’s circum-circle encloses the region which is either subregions of $C$ partitioned by $L$ and does not contain $p_1$. On the other hand, the angle between line segment $\overline{p_1p_2}$ and line segment $\overline{p_1p_3}$ is not less than $\theta$ since both $p_1$ and $p_2$ are not within $C$ and also on the extension of $L$. It follows that $\alpha$’s circum-circle encloses $\beta$’s. Thus, the lemma holds.

**Theorem 3:** Proof: Let $u \in V^-$ be a vertex in $T^-$ and $C$ be a circle centered at location $t$ with radius $|\overline{ut}|$. $T^-$ has a face $f$ which is incident to $u$ and intersects line segment $\overline{ut}$. Let $E_f$ be $f$’s edge set, $V_f$ be $f$’s vertex set, and $k$ be the number of vertices in $V_f$.

If circle $C$ encloses either no vertex in $V^-$ or at least one vertex ($\not\in V_f$) in $V_f$, this theorem trivially holds. Otherwise, suppose that circle $C$ encloses at least one vertex $w$ in $V_\Omega \equiv V^- \setminus V_f$. Namely, $w \not\in V_f$ is closer to $t$ than $u$ in $T^-$. It is not enclosed by face $f$ since otherwise at least one edge in $E_f$ has to be crossed by edges in $E_\Omega \equiv E^- \setminus E_f$ and the result contradicts that $T^-$ is a connected and plane graph. At the same time, $E_f$ has an edge $\overline{v_iv_{i+1}}$ which is clockwise with respect to line segment $\overline{v_{i+1}t}$ and also cuts circle $C$, since otherwise $w$ is not enclosed by circle $C$.

Consider the line segment $\overline{w\overline{tv}}$ between $w$ and $t$. It is either intersected by edge $\overline{v_iv_{i+1}}$ at a point $p$ (see Figure 22) or not (see Figure 23). For the former, the DT graph $T$, which $T^-$ is extracted from, has a series of triangles $T_\Delta$ successively cut by line segment $\overline{v_ip}$; one end triangle of $T_\Delta$ is incident to edge $\overline{v_iv_{i+1}}$ and the other end triangle of $T_\Delta$ is incident to $w$. The circums-circle of any triangle of $T_\Delta$ can not enclose $w$ because of DT’s circums-circle property and Lemma 1. It means that no vertex in $T_\Delta$ has to be within circle $C$. However, $w$ is within $C$. It is a contradiction. For the latter, edge $\overline{v_iv_{i+1}}$ intersects line segment $\overline{w\overline{tv}}$ at a point $q$. Then, $T$ has a series of triangles $T_\Delta$ successively cut by line segment $\overline{v_ip}$; one end triangle of $T_\Delta$ is incident to edge $\overline{v_iv_{i+1}}$ and the other end triangle of $T_\Delta$ is incident to $u$. The circums-circle of any triangle of $T_\Delta$ can not enclose $w$ because of DT’s circums-circle property and
Lemma 1. Namely, no vertex in $T_A$ is within $C$. It implies that no vertex in $N(u)$ is within $C$. From the result and Theorem 1, it follows that no vertex in $V$ is within $C$. However, it is a contradiction since we assumed that $w$ is within $C$.

From the above results, it follows that no vertex in $V_0$ is within circle $C$. Thus, this theorem holds.

D. Proof of Theorem 4
Proof: Let $u \in V$ be a current forwarding SeDAX node in a DT. The equivalent restatement to Theorem 1 is that in the DT, unless $u$ is the closest to $t$, it can always have at least one neighboring SeDAX node that is closer to a destination location. It means that in $T$, DT-GHF can always find the next hop in each SeDAX node visited during greedy forwarding. As a result, it can establish a greedy forwarding path from $v_0$ to the closest SeDAX node to $t$.

E. Proof of Theorem 5
Proof: A connected subgraph $T^-$ is extracted from the DT $T$ in the face of SeDAX node failures. In $T^-$, DT-GHF may visit a local minimum SeDAX node $u_i \in V^-$ to $t$ during greedy forwarding in $T^-$ and then starts a right-hand walk from $u_i$ on the face $f$ which is incident to $u_i$ and intersects line segment $u_f$.

If the right-hand walk from $u_i$ does not find other SeDAX node closer to $t$ than $u_i$ in face $f$, it eventually returns to $u_i$ in the plane graph $T^-$. In this case, we notice from Theorem 3 that DT-GHF reaches the closest SeDAX node $u_i$ to $t$ in $T^-$. On the other hand, unless the right-hand walk from $u_i$ returns to $u_i$ in $T^-$, we notice from Theorem 3 that DT-GHF finds a SeDAX node $w$ closer to $t$ than $u_i$ in the face $f$. Namely, the right-hand walk from $u_i$ falls back to greedy forwarding at SeDAX node $w$. During new greedy forwarding from $w$, DT-GHF reaches either the closest SeDAX node to $t$ in $T^-$ (Theorem 4) or other local minimum SeDAX node $u_{i+1}$ to $t$. For the latter, we notice by recursive induction that DT-GHF reaches the closest SeDAX node to $t$ in $T^-$. Hence, we conclude that DT-GHF always converges to the closest SeDAX node to $t$ in $T^-$. 

Young-Jin Kim is a Member of Technical Staff of the Mission-Critical Communications Department at Bell Labs, Alcatel-Lucent, Murray Hill, NJ. Dr. Kim received his B.S. and M.S. degree in Computer Science from Yonsei University, Korea and a Doctorate in Computer Science from University of Southern California, and worked at Samsung Electronics Telecommunication Labs, Korea. Since joining Bell Labs in 2010, he has been contributing to machine-to-machine communications for Smart Grids. His work typically involves the design of platforms, algorithms, and protocols on aspects of end-to-end security, self-organization, and information-centricism. His research interests include distributed algorithms and protocols for large-scale networked systems such as information-centric networks, cloud network infrastructures, and sensor networks. His research has been published in IEEE/ACM conference proceedings and journals, and has been distributed as publicly-available software.

Jaehwan Lee is a Ph.D. Candidate in the Computer Science Department at the University of Maryland, College Park. His current research interests include peer-to-peer cloud/grid computing for multi-core and heterogeneous environments, and distributed/networked system. Before his study in Maryland, he was a researcher at Korea Telecom (KT) Labs for five years. His research focused on public wireless network protocol. He received his B.S. and M.S. in Electrical Engineering from Seoul National University, Korea. He was a recipient of the General Electric (GE) Scholarship and the Korean Government Scholarship for Electric Power Industry.

Gary Atkinson is a Member of Technical Staff of the Mission-Critical Communications Department at Bell Labs, Alcatel-Lucent, Murray Hill, NJ. Dr. Atkinson received a B.S. in Math and a B.S. in Physics from the University of Texas at Austin, a M.S. in Physics from Northeastern University in Boston, and a Doctorate in Operations Research from The George Washington University in Washington, D.C., where he produced an award-winning doctoral dissertation in applied mathematical chemistry. Since joining Bell Labs in 1998, and has worked in the areas of wireless, optical, and data network design and analysis. His work is typically cross-disciplinary and involves mathematical modeling and algorithm development for network plan, design, and tradeoff analysis, including research into and application of algorithmic methods for challenging network combinatorial optimization problems. He has authored numerous technical papers in several fields and served on review committees for several conferences and journals. He is a full member of the American Physical Society, the Sigma Xi Scientific Research Society, and INFORMS.

Hongseok Kim is an Assistant Professor in the Department of Electronic Engineering at Sogang University, Korea. He received his B.S. and M.S. degree in Electrical Engineering from Seoul National University in 1998 and 2000, respectively, and Ph.D. degree in Electrical and Computer Engineering at the University of Texas at Austin in 2009. He was a member of technical staff at Korea Telecom (KT) Labs from 2000 to 2005. He worked as a Post-Doctoral Research Associate in the Department of Electrical Engineering at Princeton University from 2009 to 2010 and worked as a Member of Technical Staff at Bell Labs, USA from 2010 to 2011. His research interests are network resource allocation and optimization including cross layer design of wireless communication systems, green IT/wireless, MIMO and OFDMA, network economics, and Smart Grid. Dr. Kim is the recipient of Korea Government Overseas Scholarship in 2005-2008.

Marina Thottan is Director of the Mission-Critical Communications Department at Bell Labs, Alcatel-Lucent, Murray Hill, NJ. She joined Bell Labs in 1999, and has contributed to a wide variety of projects in diverse subject areas, including Online Gaming, Content Distribution, VoIP Routing protocols, Data over Optical Networks, High Speed Router Design, Network Management and Anomaly Detection. Most recently she has been leading work on Smart Grid Communication Networks. Marina holds an M.S. in Biomedical Engineering and a Ph.D. in Electrical and Computer Engineering from Rensselaer. Marina has published over 50 papers in scientific journals, book chapters and refereed conferences. She holds several patents in the area of network management and interactive network applications, routing algorithms and new network architectures. Marina is a member of the IEEE, and ACM.