Distributed Load Balancing for Resilient Information-Centric SeDAX Networks

Michael Hoefling, Cynthia G. Mills, and Michael Menth
University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany
Email: {hoefling,menth}@uni-tuebingen.de, cynthia.mills@student.uni-tuebingen.de

***Abstract*—SeDAX is a publish/subscribe information-centric networking architecture where publishers send messages to the appropriate message broker over a Delaunay-triangulated overlay network. Resilient data forwarding and data redundancy enable a high level of reliability. Overlay nodes and topics are addressed via geo-coordinates. A topic is stored on primary and secondary nodes, those nodes closest and second-closest to the topic's coordinate, respectively. The overlay automatically reroutes a topic's messages to its secondary node should its primary node fail. Currently, SeDAX determines the coordinate of a topic by hashing its name. This kind of topic allocation is static, which can lead to unintended load imbalances.**

In this paper, we propose a topic delegation mechanism to make the assignment of topics to nodes dynamic. Our proposed mechanism is the only existing method to improve the flexibility and resource management of the SeDAX architecture so far. We define the load of SeDAX nodes and coordinates at different levels of resilience. On this basis, we develop distributed algorithms for load balancing. Simulations show that significant load imbalance can occur with static topic assignment and that the proposed algorithms achieve very good load balancing results.

I. Introduction

The SEcure Data-centric Application eXtension (SeDAX) architecture [3] is a scalable, resilient, and secure data delivery and sharing platform for smart grids. SeDAX applies the emerging information-centric networking (ICN) paradigm to the electric utility network of sensors and controls for electricity generators, consumers, and brokers. First, we introduce the SeDAX architecture and point out the intrinsic potential for load imbalance. Then we describe our approaches to better balancing the SeDAX load.

A. The SeDAX Architecture

SeDAX's publish-subscribe communication paradigm decouples information contributors from information consumers by organizing information into *topics*. Publishers send messages to brokers that forward them to subscribers. This requires that publishers and subscribers have registered with the broker for that topic. The broker stores published topic data and keeps it available for some time. This creates load on the server.

SeDAX stores a large set of topics $\mathcal{T}$ on a set $\mathcal{V}$ of multiple brokers which are called SeDAX nodes. It supports the discovery of the appropriate message broker in a decentralized way. An overlay network steers messages addressed to a certain topic to the right SeDAX node. Thus, publishers and subscribers do not need to know the addresses of the corresponding SeDAX node to send registration and data messages, they just need to have access to the overlay network. As a result, SeDAX does not require a mapping system, that may be compromised or fail, to resolve topics to SeDAX nodes.

The overlay network is organized as follows: SeDAX nodes $v \in \mathcal{V}$ are equipped with geo-coordinates $C(v)$. Nodes are connected to selected geographic neighbors via TCP transport connections to form a Delaunay triangulated (DT) overlay network. The DT overlay network enables SeDAX nodes to forward a message addressed to a certain coordinate to the closest SeDAX node. All coordinates for which a node v is closest form its Voronoi cell $Voronoi(v)$. The SeDAX authors [3] have shown that this kind of overlay forwarding creates only little path stretch compared to the shortest path in the overlay. Furthermore, the DT overlay is self-healing: if a node fails, the DT property is restored after some local and self-organized reconfiguration.

A geographic hashing function (GHF) derives a Euclidean coordinate $h(t)$ from the name of a topic t. A topic is stored on the SeDAX node closest to that coordinate, i.e., on the node with the least Euclidean distance $d(C(v), h(t)), v \in \mathcal{V}$. The GHF and the DT overlay enable other SeDAX nodes to forward messages destined to a topic to the SeDAX node responsible for that topic.

SeDAX can be made resilient against node failures. The data and information of a topic t are stored on the SeDAX nodes that are closest (primary) and second-closest (secondary) to the topic's coordinate $h(t)$. This is simple as they are neighboring nodes. The failure of a node is detected via broken TCP connections. This triggers the self-healing of the DT overlay. Messages for topic t are then automatically forwarded to the respective alternate node, which starts delivering messages to subscribers. This resilience concept may be extended to protect against consecutive failures by ensuring that topic data and information are always kept on the closest and second-closest working SeDAX node. Thus, the self-healing property of the DT overlay combined with the backup concept constitutes a simple and effective resilience concept in SeDAX that can survive even multiple consecutive failures.

B. Problem Statement

SeDAX statically assigns topics to coordinates using the hash value $h(t)$. This is problematic if a SeDAX node becomes overloaded, since taking load away from a node is not possible without changes to the SeDAX network.

978-1-4799-0913-1/14/$31.00 © 2014 IEEE
IEEE/IFIP, Network Operations and Management Symposium (NOMS), Krakow, Poland, May 2014

C. Contributions of this Paper

(1) We propose adding *topic delegation* to SeDAX which allows dynamic assignment of topics $t \in \mathcal{T}$ to configurable coordinates $C(t)$ instead of to a fixed hash value $h(t)$.

(2) We define *load metrics* for SeDAX nodes and coordinates for different levels of resilience.

(3) We provide several distributed *load balancing algorithms* that make use of these definitions and the topic delegation mechanism.

(4) We show by *simulations* that static topic assignment can lead to significant load imbalance among SeDAX nodes and analyze the causes for that imbalance.

(5) Finally, we demonstrate that the proposed load balancing schemes can almost equalize the load over the SeDAX nodes for different resilience levels.

The paper is structured as follows. In Sect. II we review related work. Sect. III suggests the topic delegation as an extension to SeDAX. Sect. IV defines the loads of SeDAX nodes and coordinates for different levels of resilience. In Sect. V, various distributed load balancing algorithms are presented. Sect. VI quantifies and analyzes the load imbalance in a SeDAX, and shows that the proposed load balancing algorithms almost equalize the load over all SeDAX nodes. Finally, Sect. VII concludes this work.

II. Related Work

SeDAX [3] builds upon prior work in the area of publish-subscribe [4] and ICN [5]. It specifically addresses the requirements of the smart grid. A security framework [1] covers security considerations for SeDAX as a cyber-physical system. SeDAX uses topic names as input for the GHF instead of publisher names [6]. In recent work [2], we investigated the storage requirements of SeDAX necessary to survive the failure of multiple SeDAX nodes without storage shortages. This led to high requirements that could be reduced by optimized node placement, which is generally difficult to implement.

SeDAX uses the DT overlay and GHF to locate its publish/subscribe-based message brokers. Most existing ICN architectures such as PSIRP/PURSUIT [7], 4WARD/SAIL [8], NDN/CCNx [9], [10], DONA [11], and CAN [12] are based on distributed hash tables (DHTs) and publish-subscribe. They differ in the way topic names are resolved, data is forwarded, and whether the organization of data distribution is hierarchical [13] or flat as in SeDAX. QoS constraints for replication in more complex topologies with hierarchical data stores are discussed in [14], [15]. LIPSIN [16] uses bloom filters to quickly resolve names and find topic stores.

Chord [17] allocates coordinates on a ring to predecessor and successor nodes. Others like CAN [12] allocate rectangular areas to a primary node, further subdividing or combining rectangles as nodes join or exit the network. Greedy routing schemes like SeDAX organize the space into Voronoi cells so that the closest node to a coordinate is the home node for that coordinate, thus avoiding the need to maintain routing tables.

ICN systems can be viewed as structured P2P systems [18]. In a structured system like SeDAX, some nodes may provide more centralized services such as directory services (e.g. maintaining a lookup table of underloaded nodes) or security services (authoritatively authenticating a node, publisher, or subscriber). Most load balancing approaches in P2P systems focus on unstructured P2P systems [18] where nodes of disparate capacities join and exit the network frequently. SeDAX nodes are both more structured and less ephemeral whereas SeDAX publishers and subscribers can readily be mobile without requiring updates to the node routing overlay.

Load balancing schemes differ as well. Most, including those described in this paper, benefit from the "power of 2 choices" described by Mitzenmacher [19] in ball-bin load balancing. As Bridgewater et al. summarize in Balanced Overlay Networks (BON) [20] , "The important result from ball-bin systems is that if one probes the population of more than one bin prior to assigning a ball, the population of the most full bin will be reduced exponentially in N." Even and Medina [21] further discuss lower bounds for ball-bin load balancing.

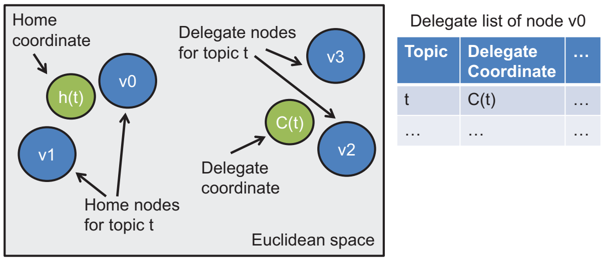
In BON, nodes change the number of immediate incoming neighbors in response to the node's availability. Thus, the overlay network can be viewed as a directed graph that is dynamically reconfigured to reflect the current system load. BON uses random walks through the directed graph to select the least loaded node on the path. BON's target application is job allocation in grid computing. In this environment jobs enter and leave the network frequently, whereas SeDAX's storage requirements tend to be of longer if not permanent duration. However, an implementation of the SeDAX random query approach might use such a random walk to include the least loaded (best) node on the path to the queried location, effectively increasing the scope of queries.

III. Topic Delegation

The existing SeDAX architecture does not support load shifting if a SeDAX node is overloaded because of its static assignment of topic coordinates $C(t)$ to coordinates $h(t)$. Our topic delegation proposal uses $h(t)$ as the default coordinate of a topic, but allows for a reassignment of $C(t)$ to any other coordinate. Topic delegation adds flexibility to SeDAX without sacrificing its benefits, e.g., resilient overlay forwarding, decentralized control, and the ability to cope without a mapping system. In the following, we explain the principle and operation of topic delegation in SeDAX.

A. Topic Delegation Principle

The basic concept of topic delegation is shown in Fig. 1. The node closest to a topic's default coordinate $h(t)$ is the topic's *home node*. By default, the topic coordinate $C(t)$ equals the topic's hash value $h(t)$ and is called *home coordinate* of topic t. When the topic coordinate $C(t)$ is set to a value other than $h(t)$, the coordinate is called *delegate coordinate* of topic t and the node closest to that coordinate is called the *delegate node* for topic t. Home nodes store the active topic coordinates $C(t)$ for topics $t \in \mathcal{T}$ in a *delegate list*. Delegate nodes are responsible for the topic, i.e., they store published topic data and keep client subscriptions.



Topic	Delegate Coordinate	…
t	C(t)	…
…	…	…

Fig. 1. Topic delegation in SeDAX. The home nodes ($v0$ and $v1$) closest to topic t's home coordinate $h(t)$ store information about topic t's delegated coordinate $C(t)$ in a delegate list. The delegate nodes ($v2$ and $v3$) store topic t's actual data.

Each topic data store, whether at the default topic coordinates or the delegate coordinates, has a secondary node which it replicates the topic data and control structures. Should a home or delegate node fail, the secondary seamlessly takes over as traffic is automatically forwarded to the secondary. When a topic moves from one delegate node to another, registrations are transferred to the new delegate node. It is up to the implementer whether the old or new delegate node informs the clients about that event.

B. Topic Delegation Operation

When a SeDAX client (publisher or subscriber) joins a topic, the client first sends a join message over the overlay to the topic's home coordinate $h(t)$ so that the message reaches the home node. The topic's home node checks its delegate list for that topic. If there is no entry, the home node itself is the message broker for that topic; no modification to the existing SeDAX architecture is needed. If the delegate list holds an entry for that topic, the home node forwards the join message to the delegate coordinate $C(t)$; this can be achieved by encapsulation to the delegate coordinate or rewrite of the destination coordinate. Upon receipt of the join message, the delegate node registers the client for the requested topic and informs the client to use the new topic coordinate $C(t)$ instead of $h(t)$ in all subsequent messages. In particular, publishers will address all data messages to $C(t)$ instead of $h(t)$. Should the topic be moved for some reason to another node, all registered clients are informed of the new delegate coordinate.

IV. Load Definitions for SeDAX Nodes and Coordinates

In this section, we propose load metrics for different levels of resilience. We first introduce auxiliary functions that facilitate later definitions. Then we consider three different levels of resilience for the operation of SeDAX. We define load metrics for SeDAX nodes and coordinates, based on which we determine a SeDAX node's best coordinate. These concepts are used by the load balancing algorithms presented in Sect. V.

A. Auxiliary Functions

The following auxiliary functions facilitate the formulation of subsequent definitions and formulae.

- $C(v), v \in \mathcal{V}$: coordinate of node v.
- $C(t), t \in \mathcal{T}$: (delegate) coordinate of topic t.
- $N_j(c)$: node whose coordinate is j-closest to coordinate c among all other SeDAX nodes, i.e, $N_1(c)$ is the closest node, $N_2(c)$ is the second-closest, etc.
- $\mathcal{T}_j(v) = \{t : t \in \mathcal{T}, N_j(C(t)) = v\}$; set of topics for which v is the j-closest node.

Since topic data may expire, SeDAX nodes require only sufficient capacity to store current, i.e., non-expired topic data, and are not intended for archival purposes. Therefore, limited storage is sufficient for the data of a topic $t \in \mathcal{T}$ which is given by the topic load $L_T(t)$.

B. Considered Resilience Levels

We consider three different resilience levels for SeDAX operation.

1) No resilience. Topic data and topic information are stored only on SeDAX node $N_1(C(t))$. If the node fails, the topic information is lost.
2) Resilience against one node failure. Topic data and information are stored redundantly on two SeDAX nodes $N_1(C(t))$ and $N_2(C(t))$. If $N_1(C(t))$ fails, messages are automatically rerouted to $N_2(C(t))$ so that they can be forwarded to the registered subscribers. If both $N_1(C(t))$ and $N_2(C(t))$ fail, the topic information is lost and publishers cannot longer reach a broker.
3) Resilience against two node failures. Topic information is stored redundantly on two SeDAX nodes $N_1(C(t))$ and $N_2(C(t))$ like above. If $N_1(C(t))$ fails, messages are automatically rerouted to $N_2(C(t))$ so that they can be forwarded to the registered subscribers. In addition, if $N_1(C(t))$ or $N_2(C(t))$ fails, topic data and information are copied to SeDAX node $N_3(C(t))$. Should the remaining node $N_1(C(t))$ or $N_2(C(t))$ also fail, then $N_3(C(t))$ takes over.

More than two successive node failures are repetitions of the two node failure scenario.

C. Load Definitions

We provide definitions for a topic's *load on a SeDAX node* and the *load on a coordinate* for different resilience levels. While the node loads serve to quantify load imbalance among nodes, the coordinate loads are used to find appropriate coordinates for load balancing.

1) Topic Load $L_T(t)$: Each topic $t \in \mathcal{T}$ induces a certain load $L_T(t)$ on the node where it is stored. Depending on a SeDAX node's capabilities, load may be measured in terms of required processing power or I/O capacity. To facilitate further considerations and calculations, we assume the topic load to be an additive metric.

2) Node Load $L_N^i(v)$: The node load $L_N^i(v)$ is the maximum load induced by topics on a node $v \in \mathcal{V}$ in any failure scenario considered by resilience level i. It is the minimum capacity for v to guarantee operation on resilience level i.

a) Resilience Level 1: A SeDAX node v is responsible only for topics $t \in \mathcal{T}$ for which it is the closest node. The maximum load induced by topics on this node is

$$L_N^1(v) = \sum_{t \in \mathcal{T}_1(v)} L_T(t). \tag{1}$$

b) Resilience Level 2: A SeDAX node v is responsible for topics for which it is the closest or second-closest node. The maximum load induced by topics on this node is

$$L_N^2(v) = \sum_{t \in (\mathcal{T}_1(v) \cup \mathcal{T}_2(v))} L_T(t). \tag{2}$$

c) Resilience Level 3: As above, a SeDAX node v is responsible for topics for which it is the closest or second-closest node; the resulting base load is $L_N^2(v)$. In addition, node v becomes responsible for topics for which it is the third-closest node should the closest or second-closest node fail. Note that these topics may have different closest and second-closest nodes so the additional node load is the sum of the additional topic loads experienced over all relevant failure cases. Those are the failures of the closest and second-closest nodes of the topics that have v as third-closest node. The maximum additional load induced by topics on this node is

$$L_{aN}^3(v) = \max_{w \in \left\{ \begin{smallmatrix} x: x \in \mathcal{V}, u \in \mathcal{T}_3(v), \\ N_1(C(u))=x \vee \\ N_2(C(u))=x \end{smallmatrix} \right\}} \sum_{t \in \left\{ \begin{smallmatrix} s: s \in \mathcal{T}_3(v), \\ N_1(C(s))=w \vee \\ N_2(C(s))=w \end{smallmatrix} \right\}} L_T(t) \tag{3}$$

and the node load for resilience level 3 is

$$L_N^3(v) = L_N^2(v) + L_{aN}^3(v). \tag{4}$$

3) Minimum and Maximum Coordinate Load ($L_{min}^i(c)$ and $L_{max}^i(c)$): We define the minimum (maximum) load of a coordinate c as the minimum (maximum) of all node loads that are affected by topics assigned to coordinate c.

a) Resilience Level 1: A topic assigned to coordinate c is stored only on the closest node $N_1(c)$ so that $N_1(c)$ stores only information of topics for which it is closest node. The (minimum and maximum) coordinate load is

$$L_{min}^1(c) = L_{max}^1(c) = L_N^1(N_1(c)). \tag{5}$$

b) Resilience Level 2: A topic assigned to coordinate c is stored on the closest node $N_1(c)$ and on the second-closest node $N_2(c)$. These nodes store the information of topics for which they are closest or second-closest. The coordinate loads are

$$\begin{aligned} L_{min}^2(c) &= \min\left(L_N^2(N_1(c)), L_N^2(N_2(c))\right) \text{ and} \quad (6) \\ L_{max}^2(c) &= \max\left(L_N^2(N_1(c)), L_N^2(N_2(c))\right). \quad (7) \end{aligned}$$

c) Resilience Level 3: Like above, a topic assigned to coordinate c is stored on the closest node $N_1(c)$ and on the second-closest node $N_2(c)$. The maximum load of those nodes is $L_N^3(N_1(c))$ and $L_N^3(N_2(c))$. Moreover, the topic may be stored on the third-closest node $N_3(c)$ if $N_1(c)$ or $N_2(c)$ fails. That node $v = N_3(c)$ carries the load $L_N^2(v)$ from topics for which it is closest or second-closest node. If $N_1(c)$ or $N_2(c)$ fails, node $v = N_3(c)$ carries in addition the load from all topics that have $N_1(c)$ or $N_2(c)$ as closest or second-closest node, and v as third-closest node. Thus, the failure-set-specific additional node load $L_{faN}^3(v,c)$ of v for coordinate c is

$$L_{faN}^3(v,c) = \max_{w \in \{N_1(c), N_2(c)\}} \sum_{t \in \left\{ \begin{smallmatrix} s: s \in \mathcal{T}_3(v), \\ N_1(C(s))=w \vee \\ N2(C(s))=w \end{smallmatrix} \right\}} L_T(t). \tag{8}$$

Hence, the coordinate loads are

$$\begin{aligned} L_{min}^3(c) &= \min(L_N^3(N_1(c)), L_N^3(N_2(c)), \\ &\quad L_N^2(N_3(c)) + L_{faN}^3(N_3(c),c)) \text{ and} \quad (9) \\ L_{max}^3(c) &= \max(L_N^3(N_1(c)), L_N^3(N_2(c)), \\ &\quad L_N^2(N_3(c)) + L_{faN}^3(N_3(c),c)). \quad (10) \end{aligned}$$

D. Definition of Best Coordinates

We define $\mathcal{C}^*$ as a set of coordinates. If a new topic should be assigned to a coordinate from that set, the coordinate should be carefully selected such that it minimizes the maximum load of all nodes, maximizes the minimum load of all nodes, and minimizes the required backup capacity. This translates to the following three criteria based on the metrics L_{max}^i, L_{min}^i, and coordinate-specific spare capacity:

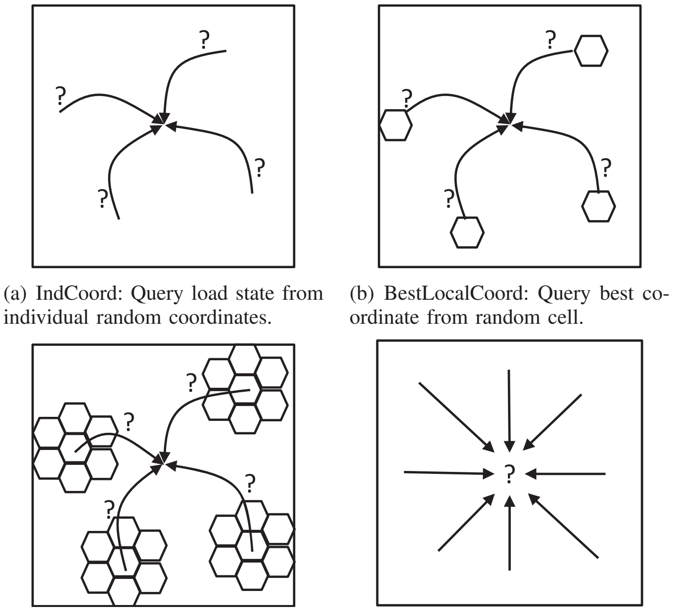
1) Select a small maximum coordinate load L_{max}^i.
2) Select a small minimum coordinate load L_{min}^i.
3) Only for resilience level 3: select a large coordinate-specific spare capacity on the coordinate's third-closest node $N_3(c)$. It is the spare capacity on $N_3(c)$ if either the closest node $N_1(c)$ or the second-closest node $N_2(c)$ fails. That capacity is calculated as $L_N^3(N_3(c)) - (L_N^2(N_3(c)) + L_{faN}^3(N_3(c),c))$.

We define that a coordinate c_0 is better than a coordinate c_1 if it is better in the first criterion (small $L_{max}^i(c)$). Or if it is equal in the first criterion but better in the second one (small $L_{min}^i(c)$). Or if it is equal in the first two criteria and better in the third one (coordinate-specific spare capacity). A coordinate of a coordinate set $\mathcal{C}^*$ is best if there is no better coordinate in that set. Thus, several best coordinates may exist. These criteria combine the best heuristics in our experiments.

For resilience level 1, all coordinates of a Voronoi cell $Voronoi(v)$ of a node $v \in \mathcal{V}$ are equally good. This is different for resilience level 2 and 3. Here, a mathematical analysis yields the area of best coordinates. Alternatively a best coordinate may be found empirically by selecting the best coordinate of a set of random coordinates within a node's Voronoi cell. This is much simpler, but may not find the absolute best coordinate.

V. Distributed Load Balancing Algorithms

We present four different algorithms for distributed load balancing in SeDAX that support resilience levels 1, 2, and 3. If a node v wants to delegate a topic with a coordinate $C(t) \in Voronoi(v)$ within its own Voronoi cell to another coordinate, we call it a delegating node. This delegating node needs to find a better coordinate according to the definitions in Sect. IV-D. Load metrics in this section should be computed excluding the topic to be delegated.



(a) IndCoord: Query load state from individual random coordinates.

(b) BestLocalCoord: Query best coordinate from random cell.

(c) BestRegionalCoord: Query best coordinate from random region.

(d) BestGlobalCoord: Choose globally best coordinate based on flooding.

Fig. 2. Algorithms for finding a delegation coordinate.

A. *Querying for Individual Coordinates (IndCoord)*

A delegating node may send a query to a random coordinate c that is forwarded to its closest node $N_1(c)$ over the DT overlay. This node locally computes the metrics L^i_{max}, L^i_{min}, and coordinate-specific spare capacity as proposed in Sect. IV-D and returns them to the delegating node. The delegating node may issue $n_{queries}$ such queries so that it eventually knows the relevant metrics of $n_{queries}$ other coordinates and its own $C(t)$. On this basis the delegating node can choose the best coordinate and assign the topic. This method is illustrated in Fig. 2(a).

B. *Querying Locally Best Coordinates (BestLocalCoord)*

This differs from IndCoord in that the node $N_1(c)$ determines a locally best coordinate within its Voronoi cell $Voronoi(v)$ according to Sect. IV-D. It returns that coordinate including the relevant metrics to the delegating node. Thus, the delegating node receives $n_{queries}$ locally best coordinates and also computes its own locally best coordinate. The topic is assigned to the best coordinate among them. This method is illustrated in Fig. 2(b). It causes more computational overhead than IndCoord, but it is likely to find better coordinates.

C. *Querying Regionally Best Coordinates (BestRegionalCoord)*

Here, the node receiving the query returns a regionally best coordinate selected from the coordinates of its own cell and of those cells within n_{hops} hops. Thus, the delegating node receives $n_{queries}$ regionally best coordinates and also computes its own regionally best coordinate. The topic is assigned to the best coordinate among those. This method is illustrated in Fig. 2(c). It causes more computational overhead and involves more communication than the methods presented above, but it is more likely to find better coordinates.

D. *Determining Globally Best Coordinates Based on Flooding (BestGlobalCoord)*

The delegating node floods a request to all other nodes (or at least one node in each region) for their best coordinates. The responses allow the delegating node to determine a globally best coordinate to which the topic is assigned. This method is illustrated in Fig. 2(d). It may require more computation and communication than the methods presented above, but it is able to find a network-wide best delegation coordinate given the current network state.

VI. Performance Evaluation

This section investigates potential load imbalance in SeDAX overlays by simulation experiments. First, the simulation setup is described. The complementary cumulative distribution functions (CCDFs) illustrate that the existing SeDAX can lead to significant load imbalance for which we analyze the causes. We show that load balancing based on global information can equalize the load among all nodes and highlight the importance of the appropriate resilience level for load balancing. As global information may be difficult to obtain, we show that simpler approaches can also lead to good load balancing results.

A. *Experiment Setup and Methodology*

We use a square plane as coordinate space on which n_{nodes} nodes are positioned randomly. Each node is assigned n^{topics}_{node} topics on average. We generate $n_{topics} = n^{topics}_{node} \cdot n_{nodes}$ topics, and each t of these topics comes with a random coordinate $C(t) = h(t)$. These topics are iteratively added to SeDAX. When topic delegation is enabled, a load balancer may reassign each topic to a different coordinate $C(t)$ based on the current load situation in the overlay; otherwise the original random topic coordinates remain. We study two choices for topic loads.

- Homogeneous topic load: each topic has the same load $L_T = 1$.
- Heterogeneous topic load: 80% of the topics have load $L_T = \frac{1}{4}$, 20% of the topics have load $L_T = 4$. The average load is also $E[L_T] = 1$ and the coefficient of variation of that distribution is 1.5.

After the successive generation of topics, assignment to coordinates, and load balancing, node loads are calculated for all nodes $v \in \mathcal{V}$ and the CCDF of their loads is determined. We perform each experiment 100 times, average the CCDFs from single simulation runs, and show the 95% confidence intervals where appropriate.

B. *Load Distribution in SeDAX without Topic Delegation*

We simulate $n_{nodes} = 100$ nodes in the plane with $n^{topics}_{node} \in \{1000, 100, 10\}$ homogeneous-load topics per node and $n^{topics}_{node} \in \{100, 10\}$ heterogeneous-load topics per node. Fig. 3 shows the CCDF of the node loads $L^1_N(v)$ for resilience level 1 for $n^{topics}_{node} \in \{100, 10\}$. The curve for $n^{topics}_{node} = 1000$ is omitted in the figure as it visually coincides with the curve

TABLE I
MEAN VALUE $\bar{x}$, 1% AND 99% QUANTILES OF NODE LOAD $L_N^i(v)$ FOR $n_{nodes} = 100$ WITHOUT TOPIC DELEGATION.

	$n_{node}^{topics} = \mathbf{1000}$			$n_{node}^{topics} = \mathbf{100}$						$n_{node}^{topics} = \mathbf{10}$					
	(homogeneous topic loads)			(homogeneous topic loads)			(heterogeneous topic loads)			(homogeneous topic loads)			(heterogeneous topic loads)		
	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$
L_N^1	100.0%	11.8%	247.8%	100.0%	12.4%	254.6%	100.0%	12.3%	252.8%	100.0%	0.0%	264.0%	100.0%	0.0%	310.2%
L_N^2	200.0%	49.6%	410.9%	200.0%	51.0%	415.5%	200.0%	47.1%	416.4%	200.0%	30.4%	433.2%	200.0%	9.3%	494.2%
L_N^3	257.4%	78.8%	499.1%	258.0%	83.8%	511.2%	259.2%	78.6%	506.5%	262.9%	68.0%	544.0%	271.3%	29.3%	586.5%

TABLE II
CORRELATION COEFFICIENTS BETWEEN VORONOI CELL SIZE $A(v)$ AND NODE LOAD $L_N^i(v)$ FOR $n_{nodes} = 100$.

$corr$	$n_{node}^{topics} = \mathbf{1000}$	$n_{node}^{topics} = \mathbf{100}$		$n_{node}^{topics} = \mathbf{10}$	
	(homogeneous)	(homo-geneous)	(hetero-geneous)	(homo-geneous)	(hetero-geneous)
L_N^1	0.9984	0.9844	0.9522	0.8680	0.6996
L_N^2	0.6830	0.6740	0.6528	0.6026	0.4914
L_N^3	0.6028	0.5954	0.5770	0.5336	0.4365

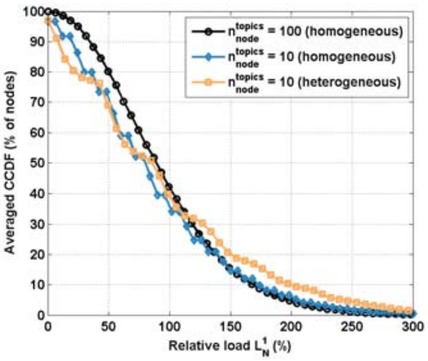


Fig. 3. CCDFs of the node loads L_N^1 for resilience level 1 demonstrate a significant load imbalance. The experiments were conducted with $n_{nodes} = 100$ using homogeneous and heterogeneous topic loads. The CCDFs are averaged over 100 simulation runs.

for $n_{node}^{topics} = 100$. Node loads are relative, i.e., 100% relative load corresponds to a node load of n_{node}^{topics}. The lines are interpreted as follows: for a node load x on the x-axis, the y-axis gives the percentage of nodes whose node load X is greater than x. Thus, equal load on any node would result in a vertical line at 100% node load. The figure rather shows a continuous decrease over a load range between 0% and 250% for $n_{node}^{topics} = 100$. The curves for $n_{node}^{topics} = 10$ homogeneous-load topics have a slightly greater load imbalance which increases for heterogeneous-load topics.

Fig. 4(a) shows in addition to the distribution of node load L_N^1 the distribution of node loads L_N^2 and L_N^3, i.e., the loads for resilience levels 2 and 3. The loads are significantly greater than the load of resilience level 1. While the L_N^1 loads have a mean of 100%, the L_N^2 loads have a mean of 200% because each topic has to be stored twice, and they range between 0% and 450% per node. The L_N^3 loads have a mean of about 260% and range between 0% and 550%. The mean of the L_N^3 load is less than 300% because topics can share the normally unused backup capacity of SeDAX nodes if they have different primary and secondary nodes. Load imbalance increases both with fewer topics per node and with increasing variance of the topic loads. As exact values for load imbalance are hard to determine from the figures, Table I shows the 1% and 99% quantiles of the loads. These values increase with increasing resilience level. The 99% quantiles may be useful for capacity provisioning. They can easily amount to 200% – 250% of the respective mean values. This is highly inefficient but necessary in the absence of load balancing capabilities.

A good part of the strong load imbalance is caused by the strong imbalance of the Voronoi cell size. The average Voronoi cell size is $\frac{A_{square}}{n_{nodes}}$, where A_{square} is the area of the square on which the simulation is based. If we take this as 100%, the 1% and 99% quantile of the cell sizes is 11.6% and 249.4%. This is very close to the quantiles of the load distribution with $n_{node}^{topics} = 1000$ homogeneous-load topics. Table II shows the correlation coefficients between the Voronoi cell size and the load of SeDAX nodes for different topic loads and resilience levels. We observe high correlations for all cases. The correlation is largest for resilience level 1 and 1000 homogeneous-load topics per node, and decreases for fewer topics per node, heterogeneous topic loads, and higher resilience levels. Thus, the observed load imbalance is largely due to different cell sizes. [*]

C. Load Distribution in SeDAX for Load Balancing Using Global Knowledge

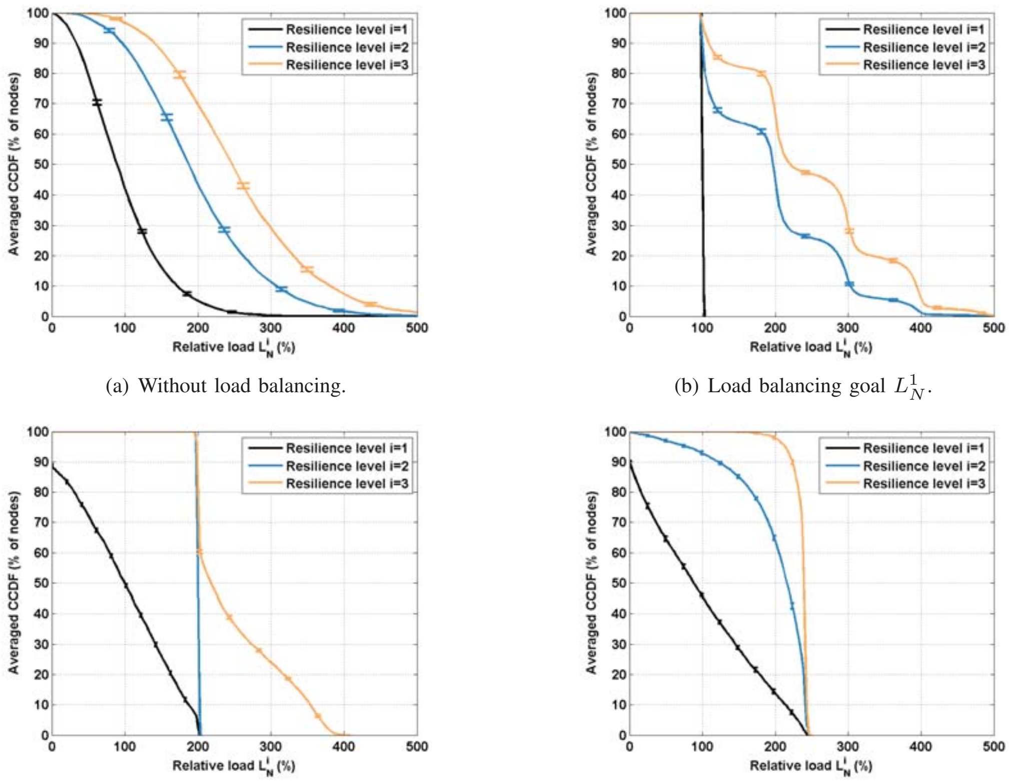
In this section we investigate the effect of topic delegation and the associated load balancing using global knowledge as proposed in Sect. V-D. We add topics one after another to SeDAX and perform a load balancing decision for each new topic, i.e., whether it should be assigned to its default coordinate $C(t) = h(t)$ or to a coordinate $C(t)$ recommended by some other node. In such an experiment, the following inequality must be met after each topic assignment:

$$\min_{c \in \mathcal{C}}(L^3_{max,old}(c)) \leq \max(\max_{v \in \mathcal{V}}(L^3_{N,new}(v)) - L_T^{assigned}, L^3_{N,new}(N_3(c_{assigned}))) \qquad (11)$$

whereby the subscript "old" and "new" refer to the respective metric before and after topic assignment, and $L_T^{assigned}$ and $c_{assigned}$ refer to the load and the coordinate of the last assigned topic. We used this equation to validate the correctness of the load balancing results.

In the following, we perform load balancing with various objectives, namely to equalize the L_N^1, L_N^2, or L_N^3 load.

[*] We also conducted experiments with more and fewer nodes, but the results are so similar that we do not show them here.



(a) Without load balancing.

(b) Load balancing goal L_N^1.

(c) Load balancing goal L_N^2.

(d) Load balancing goal L_N^3.

Fig. 4. CCDF of node loads L_N^1, L_N^2, and L_N^3 without load balancing and for load balancing goals L_N^1, L_N^2, and L_N^3. The experiments were conducted with $n_{nodes} = 100$, $n_{node}^{topics} = 100$ heterogeneous-load topics per node. The CCDFs are averaged over 100 simulation runs with 95% confidence intervals.

1) Equalizing L_N^1 Node Load: Fig. 4(b) illustrates the CCDF of the L_N^1, L_N^2, and L_N^3 load when topics are load balanced for L_N^1. The L_N^1 load is well balanced over all nodes and the maximum L_N^1 load is near 100%. However, the L_N^2 load ranges between 100% and 500%. Thus, this simple load balancing approach does not lead to equalized data volumes on SeDAX nodes when SeDAX is operated under failure-free conditions in a resilient mode. For resilience level 3, the maximum load on SeDAX nodes ranges even between 100% and 500%.

2) Equalizing L_N^2 Node Load: Fig. 4(c) shows the respective results when L_N^2 is used as load balancing goal. The L_N^1 load is almost equally distributed between 0% and 200% which is not a balanced result. However, the L_N^2 load is well equalized among all nodes, which is the balancing goal. That means, the data volumes on SeDAX nodes are about the same on all nodes when SeDAX is operated under failure-free conditions in a resilient mode. The CCDF of the L_N^3 load shows the distribution of the maximum node load during single node failures. In spite of an excellent load distribution under failure-free conditions, heavy load spikes[†] can occur on nodes during single node failures with values ranging from 200% to 400%.

3) Equalizing L_N^3 Node Load: Fig. 4(d) presents the load distribution for load balancing objective L_N^3. The L_N^1 and L_N^2 loads are each approximately uniformly distributed between 0% and 240%. However, the maximum load node L_N^3 is about 240%; this means that no SeDAX node carries much more than 240% even during single node failures. This is a desirable feature even though the distribution of the actual load under failure-free operation is far from being equalized.

These investigations demonstrate that the load balancing objective for SeDAX needs to be carefully chosen. The simple L_N^1 load balancing goal cannot equalize the load of resilient SeDAX under failure-free conditions. The more complex L_N^2 load balancing goal achieves that objective, but cannot avoid load spikes during single node failures. Only the more complex L_N^3 load balancing goal is able to minimize load spikes during single node failures.

D. Load Distribution in SeDAX for Load Balancing Using Limited Knowledge

Load balancing with global knowledge requires the calculation of the best coordinates of all SeDAX nodes and their communication to the load balancing node. That can be expensive in networks with many nodes and topics, so it is interesting to consider load balancing approaches that require less effort.

In the following, we examine the various load balancing algorithms presented in Sect. V. We focus on balancing of

[†] Load spikes refer to the additional storage capacity to be provided by tertiary nodes, not the signaling overhead.

TABLE III

IMPACT OF DIFFERENT LOAD BALANCING ALGORITHMS AND $n_{queries}$ ON MEAN VALUE $\bar{x}$, 1% AND 99% QUANTILES OF NODE LOAD L_N^3.

	IndCoord			**BestLocalCoord**			**BestRegionalCoord**			**BestGlobalCoord**		
$n_{queries}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$	$\bar{x}$	$q_{1\%}$	$q_{99\%}$
1	254.0%	127.2%	291.6%	245.3%	130.3%	371.4%	235.8%	216.7%	244.4%	236.2%	186.2%	243.3%
10	245.8%	181.9%	253.7%	235.6%	218.5%	237.9%	235.1%	211.6%	240.8%	236.2%	186.2%	243.3%
100	237.7%	222.0%	241.6%	235.4%	211.6%	240.8%	236.1%	187.1%	244.5%	236.2%	186.2%	243.3%

the L_N^3 load with $n_{nodes} = 100$ nodes and $n_{node}^{topics} = 100$ heterogeneous-load topics. All investigated approximation algorithms are based on the principle of random queries. In all experiments, we use $n_{queries} = \{1, 10, 100\}$ queries per topic delegation decision.

Table III shows the mean L_N^3 load, the 1% and the 99% quantiles of the averaged CCDFs of the experiments. These values are all significantly lower than without load balancing (259.2%, 78.6%, and 506.5% in Table I). This means that *all* the proposed algorithms solve the practical problem for SeDAX, as they reduce the 99% quantile of the load by as much as $\frac{506.5\%-237.9\%}{506.5\%} \approx 53\%$. Nevertheless, since the general load balancing problem maps to the NP-hard 0/1 knapsack problem and the investigated algorithms all try a greedy solution at each topic arrival, none of the results is likely to be fully optimal.

For IndCoord the load balancing results improve with increasing $n_{queries}$: the mean load decreases, the 1% quantile increases, and the 99% quantile decreases. BestLocalCoord behaves similarly, but the 99%-quantile slightly degrades for large $n_{queries}$. For BestRegionalCoord with $n_{hops} = 1$ and $n_{queries} = 10$ or more, the 99% quantile of the L_N^3 load is worse than for BestLocalCoord. This is surprising because that algorithm has load information about more coordinates than BestLocalCoord. BestGlobalCoord does not depend on $n_{queries}$, therefore, we have the same values for all three rows. BestGlobalCoord leads to very good load balancing results compared with no load balancing at all, but its 99% quantile of the L_N^3 load is outperformed by any approximation algorithm for at least one setting of $n_{queries}$.

The fact that load balancing algorithms with limited knowledge can outperform the load balancing algorithm with global knowledge seems surprising. By incrementally equalizing existing load before adding large topics, BestGlobalCoord can cause load spikes on a few nodes. In contrast, load balancing algorithms with limited knowledge equalize the load for only a limited set of coordinates, leading to a globally imperfect balance with larger load differences between coordinates. This leaves room for larger topics to be more evenly distributed, since when a large topic is assigned, the probability of a coordinate having significantly less load is larger than for BestGlobalCoord. Although this helps explain the observed phenomena, it also hints that future research can further improve load balancing algorithms, particularly for the investigation of load balancing in larger networks.

VII. CONCLUSIONS

The existing SeDAX information-centric architecture statically assigns topics to coordinates. In this work we showed that this can lead to severe load imbalance on SeDAX nodes. Therefore, we proposed a modification allowing dynamic reassignment of topics to coordinates while retaining the benefits of SeDAX, i.e., resilient overlay forwarding, decentralized control, and the ability to cope without a mapping system.

We characterized the load on SeDAX nodes for three different levels of resilience. On this basis, we developed various load balancing algorithms. We showed that the observed load imbalance in existing SeDAX is due to topological structures, i.e., varying Voronoi cell sizes, and does not vanish with scaling to larger number of topics or nodes.

We demonstrated that the proposed load balancing works well for all three considered resilience levels, i.e., it significantly reduces the 99% quantile of the load on all nodes. For resilient SeDAX that survives at least two node failures, the relative reduction is 53% and also the amount of shared backup capacity is clearly reduced. As load balancing using global knowledge requires many information updates, which may raise scalability concerns, we also proposed simpler load balancers that work with only limited knowledge. We showed that they can lead to equally good or even better results than load balancing with global knowledge. This suggests that such simpler variants are feasible and that there is room for improvement of the load balancer operating on global knowledge.

The performance evaluation in our work was simplified in the sense that node capacities were not limited, topic sizes remained stable, and topics were not removed so that load balancing decisions were required only at the creation of new topics. A full-fledged resource management scheme needs to cope with such events; in particular it requires appropriate triggers to re-assign topics to other coordinates if needed. Re-assignment during operation is challenging and causes significant communication overhead, therefore, the re-assignment rate should be kept small while the load over all nodes should still be balanced. Thus, the proposed load definitions and load balancing algorithms provide a useful base for further research.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7-ICT-2011-8 under grant agreement n° 318708 (C-DAX). The authors alone are responsible for the content of this paper.

The authors thank Marina Thottan, Marcel Mampaey, Wolfgang Braun, and Florian Heimgaertner for valuable input and stimulating discussions.

REFERENCES

[1] Y.-J. Kim, V. Kolesnikov, H. Kim, and M. Thottan, "Resilient End-to-End Message Protection for Large-scale Cyber-Physical System Communications," in *IEEE SmartGridComm*, Nov. 2012.

[2] M. Hoefling, C. G. Mills, and M. Menth, "Analyzing Storage Requirements of the Resilient Information-Centric SeDAX Architecture," in *IEEE SmartGridComm*, Oct. 2013.

[3] Y.-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications," *IEEE JSAC*, vol. 30, no. 6, 2012.

[4] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys*, vol. 35, no. 2, 2003.

[5] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Information-Centric Networking: Seeing the Forest for the Trees," in *ACM HotNets*, Nov. 2011.

[6] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table," *Mob. Netw. Appl.*, vol. 8, no. 4, 2003.

[7] D. Trossen and G. Parisis, "Designing and Realizing an Information-Centric Internet," *IEEE Mobile Communications*, vol. 50, no. 7, 2012.

[8] R. Alimi, L. Chen, D. Kutscher, H. H. Liu, A. Rahman, H. Song, Y. R. Yang, D. Zhang, and N. Zong, "An Open Content Delivery Infrastructure using Data Lockers," in *ACM SIGCOMM Workshop on Information-Centric Networking (ICN)*, 2012.

[9] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *ACM SIGCOMM CCR*, vol. 42, no. 3, 2012.

[10] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *ACM CoNEXT*, 2009.

[11] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinsky, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," in *ACM SIGCOMM*, Aug. 2007.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *ACM SIGCOMM CCR*, vol. 31, no. 4, 2001.

[13] A. Ghose, J. Grossklags, and J. Chuang, "Resilient Data-Centric Storage in Wireless Ad-Hoc Sensor Networks," in *Proc. of the International Conference on Mobile Data Management (MDM)*, 2003.

[14] M. Shorfuzzaman, P. Graham, and R. Eskicioglu, "Distributed Placement of Replicas in Hierarchical Data Grids with User and System QoS Constraints," in *IEEE 3PGCIC*, 2011.

[15] X. Tang and J. Xu, "QoS-Aware Replica Placement for Content Distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, 2005.

[16] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," *ACM SIGCOMM CCR*, vol. 39, no. 4, 2009.

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *ACM SIGCOMM CCR*, vol. 31, no. 4, 2001.

[18] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, 2005.

[19] M. D. Mitzenmacher, "The Power of Two Choices in Randomized Load Balancing," Ph.D. dissertation, University of California at Berkeley, 1996.

[20] J. S. A. Bridgewater, P. O. Boykin, and V. P. Roychowdhury, "Balanced Overlay Networks (BON): Decentralized Load Balancing via Self-Organized Random Networks," *CoRR*, vol. cs.DC/0411046, 2004.

[21] G. Even and M. Medina, "Parallel Randomized Load Balancing: A Lower Bound for a More General Model," in *SOFSEM 2010: Theory and Practice of Computer Science*. Springer Berlin Heidelberg, 2010, vol. 5901.