Enabling Resilient Smart Grid Communication over the Information-Centric C-DAX Middleware

Michael Hoefling*, Florian Heimgaertner*, Michael Menth*, Konstantinos V. Katsaros†,
Paolo Romano‡, Lorenzo Zanni‡, George Kamel§
*University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany, Email: hoefling@uni-tuebingen.de
†University College London, Department of Electronic and Electrical Engineering, London, United Kingdom
‡Swiss Federal Institute of Lausanne (EPFL), Distributed Electrical System Laboratory, Lausanne, Switzerland
§University of Surrey, Centre for Communication Systems Research, Guildford, Surrey, United Kingdom

***Abstract*—Limited scalability, reliability, and security of todays utility communication infrastructures are main obstacles to the deployment of smart grid applications. The C-DAX project aims at providing and investigating a communication middleware for smart grids to address these problems, applying the information-centric networking and publish/subscribe paradigm. We briefly describe the C-DAX architecture, and extend it with a flexible resilience concept, based on resilient data forwarding and data redundancy. Different levels of resilience support are defined, and their underlying mechanisms are described. Experiments show fast and reliable performance of the resilience mechanism.**

I. Introduction

Power distribution networks are undergoing major changes in operational procedures and monitoring, thereby evolving from passive to active networks [1], [2]. Advanced smart monitoring tools result in faster and more reliable real-time state estimation (RTSE) [3], [4]. Especially extensive synchrophasor measurements can achieve a more complete view and improve control of power networks [4], [5], [6]. Main obstacles to the deployment of smart grid (SG) applications are limited scalability, reliability, and security of todays utility communication infrastructures. The *National Institute for Standards and Technology* (NIST) working group on SGs [7] identified reliability requirements for SG communication flows.

The *Cyber-secure Data and Control Cloud for power grids* (C-DAX) project [8] aims to provide such a communication middleware by applying the emerging information-centric networking (ICN) [9] and publish/subscribe (pub/sub) [10] paradigm to the electric utility network of sensors and controls. The major advantages of C-DAX architecture are resiliency, inter-domain communication, cyber security, flexibility, and support for real-time applications.

The main contribution of this paper is a brief description of the overall C-DAX architecture, and a detailed presentation of its flexible resilience concept. Instead of a fixed resilience concept for all SG applications, C-DAX' resilience concept provides four different levels of resilience support, which can be selected per information channel by application developers. Parts of the resilience concept are already implemented in the C-DAX prototype, and will be deployed in a real-world power grid as part of a field trial.

This work is structured as follows. We review the use case of synchrophasor-based RTSE of active distribution networks in Section II, and briefly present the C-DAX architecture in Section III. In Section IV, we explain the resilience concept of the C-DAX architecture in detail, and Section V shows experimental performance evaluation results based on the current prototype implementation. We discuss related work in the context of resilience in pub/sub and ICN architectures in Section VI, and draw conclusions in Section VII.

II. Synchrophasor-Based Real-Time State Estimation of Active Distribution Networks

As known, power networks can be divided in two main systems: transmission and distribution networks. The role of transmission networks is mainly electrical power transport, whereas distribution networks transport and deliver power to the consumers. The latters are experiencing large changes in view of the vast deployment of distributed generation essentially associated to dispersed renewable energy resources. In this respect, the concept of active distribution network (ADNs) is applied to distribution networks characterized by the presence of distributed energy resources (DERs) together with a smart energy management system capable to exploit various control functionalities, e.g., voltage control, losses minimization, optimal dispatch of DERs, and automatic adaptation of protections. Currently, the lack of available distributed measurement infrastructures at the distribution grid level represents one of the main obstacles for distribution network operators to develop adequate controls capable to enable the seamless integration of DERs. Within this context, one of the most promising technologies for the ADNs monitoring is associated to the concept of the synchrophasor-based RTSE [2], [3], [4], [11], [12], [13]. The technical base components of this technology are phasor measurement units (PMUs) and phasor data concentrators (PDCs). PMU devices measure the equivalent phasor representation of the power-system waveforms (i.e., voltages and currents) in different points of the power grid. The measurement data are accurately time-stamped using a reliable time source, such as the UTC-GPS, and sent to the PDC with a refresh rate up to 50 times per second [14], [15]. PDCs receive, time-align and aggregate measurement data from different

978-1-4799-5804-7/15/$31.00 © 2015 IEEE
ITG/GI International Conference on Networked Systems (NetSys), Cottbus, Germany, March 2015

PMUs based on the time-stamp, and provide the aggregated data to the RTSE application. In turn, this feeds the time-aligned and aggregated measurement data into a mathematical model of the distribution grid to estimate the current state of the grid. The outcome of the estimation may be used by several power-system applications, e.g., grid monitoring and control, and fault identification and location. Describing the RTSE mechanism and its possibilities for distribution network operations in detail is out of scope of this work. However, compared to traditional *supervisory control and data acquisition* (SCADA) systems, synchrophasor-based RTSE allows estimating the system's state with increased accuracy, high refresh rate and reduced time latencies, providing distribution network operators a complete and real-time view and control of their ADNs.

Today, PMU measurement technology is already deployed on the transmission grid level in several countries around the world, e.g., the NASPI (North American SynchroPhasor Initiative) operates a large-scale measurement infrastructure called NASPInet [16], [17], or the Synchrophasor Initiative in India [18]. Still, PMU measurement technology has not been widely deployed on the distribution grid level yet. The C-DAX project [8] implements synchrophasor-based RTSE as one of its use cases in which the C-DAX middleware is used as communication technology between PMUs and PDCs, to demonstrate the advantages and feasibility of RTSE on the distribution grid level to potential stakeholders. Data transmission between PMUs in the field and the PDC at the grid control center needs to be robust against potential intermediary network component failures to perform reliable RTSE; accidental data loss may be interpreted as power grid failures by the RTSE which may distort the estimation result. Therefore, we use RTSE as use case in this paper to demonstrate the reliable and fast resiliency feature of the C-DAX architecture.

III. C-DAX: A Cyber-Secure Data and Control Cloud for Power Grids

C-DAX is an FP7 project funded by the European Commission which adapts the ICN and pub/sub paradigm to the needs of power grids. It aims at developing a cyber-secure and scalable communication middleware for SGs to facilitate the flexible integration of emerging SG applications. It proves the benefits of by suitable use cases, a prototype, and a field trial. We give a broad overview on the C-DAX architecture, its design rationales, components, basic interactions, and briefly introduce its more advanced features.

A. Design Rationale

Traditional power grid communication solutions are based on the client-server communication model. This requires both communication end-points to be aware of each other. Clients need to be configured with detailed communication parameters, e.g., IP addresses and port numbers of servers, and probably more communication protocol-specific parameters. Servers need to be configured properly to allow only access

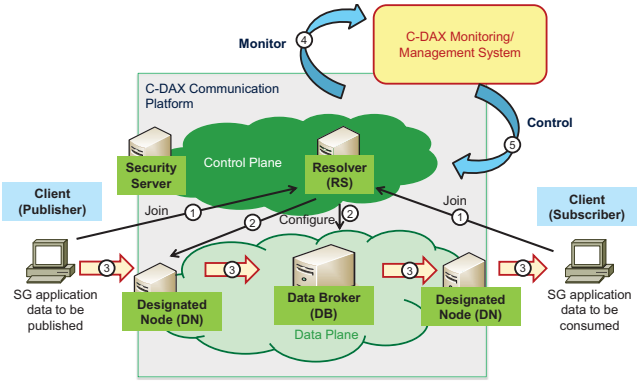


Fig. 1. The C-DAX architecture. Basic signaling steps include client join (step 1), data plane configuration (step 2), and topic data transmission (step 3). Further signaling includes monitoring (step 4) and general control of the C-DAX cloud (step 5).

from trustworthy clients. When servers undergo a service cycle, all clients need to be re-configured to communicate with backup servers. When new clients are added to the system, the access control of the servers needs to be re-configured.

C-DAX uses the information-centric communication model instead of the client-server communication model. Information is organized in so-called *topics*. A topic is an abstract representation of a unidirectional information channel with a certain storage capacity; the storage capacity is the validity period of the stored information. A topic is addressed using its unique name and probably attributes, e.g., data type, location, and time. An example for a topic is phasor measurement data for a specific geographic region inside the distribution grid. Topics and topic names are key elements for the pub/sub and ICN paradigm.

The basic idea of the pub/sub paradigm is the decoupling of communication partners in space, time, and synchronization [10], [19]. Publishers and subscribers register at a *broker* for a certain topic. Publishers send messages for that topic to the broker, which eventually forwards them to the subscribers. In the RTSE use case, PMUs are publishers, and PDCs are subscribers. The ICN paradigm is a global-scale version of the pub/sub paradigm. It provides finer grained interface semantics for accessing information in the network compared to pure pub/sub, universal in-network caching, and content-oriented security [9]. The goal of applying pub/sub and ICN in C-DAX is to improve scalability compared to traditional client-server communication, and to facilitate development of new communication-based applications by providing a standardized transparent interface [10], [20], [21].

B. Components

Fig. 1 illustrates the basic structure and interactions of the C-DAX architecture. It is composed of C-DAX clients and the C-DAX cloud. SG applications use *C-DAX clients* as interface to the C-DAX cloud, which handle all C-DAX signaling transparent to the respective application. *Publishers* are C-DAX clients generating data for a specific topic. *Subscribers* are C-DAX clients interested in certain topic data.

C-DAX nodes form the *C-DAX cloud*, and provide a specific set of functions to the cloud and clients. Possible functions are storage of topic data, resolving topic-to-node mappings, providing security functionalities, providing monitoring facilities, and providing management interfaces for operators. We briefly describe the functions from bottom to top, and assign them to their respective plane, e.g., data, control, or management plane.

1) Data Plane: Designated nodes (DNs) provide access for clients to the C-DAX cloud. They act as first point of contact and are responsible for forwarding topic data to and from the cloud, i.e., clients are pre-configured with DNs. *Data brokers (DBs)* store and forward topic data to DNs. Each topic is assigned to a DB, where its publishers send topic data to. DBs store topic data for a certain time, and forward it to the topic's subscribers. The exact assignment of topics to DBs is subject to management decisions, and may be changed during runtime.

2) Control Plane: Topic names need to be mapped to DBs so that join requests can be sent to appropriate DBs that manage registrations. To that end, *resolvers (RSes)* hold topic-to-DB mappings and provide a resolution interface through which they answer mapping requests of other nodes. There may be several RSes in a C-DAX cloud, e.g., for resiliency or extensibility reasons. In that case, a *resolver discovery system (RDS)* is necessary which provides a mechanism to discover RSes when given a topic name. Security-related functionalities are provided by a *security server (SecServ)*, e.g., authentication, authorization, and key distribution.

3) Management Plane: Management and monitoring is provided by the respective *management system (MgmSys)* and *monitoring system (MonSys)*. The MgmSys is responsible for topic and node management, and provides an operator interface for remote management. Topic management includes creation, deletion, migration, and configuration of topics during runtime. Topic migration allows operators to move topics from one set of DBs to another set of DBs, e.g., to perform load balancing. Topic configuration allows operators to change the attributes for a topic, e.g., changes in the access control list of a topic. Node management enables addition and removal of a C-DAX node from the cloud. The MonSys provides mechanisms to gather, and aggregate monitoring information.

C. Basic Interactions

1) Publication of Topic Data: Initial message exchange prior to topic data publication is shown on the left side of Fig. 1. We assume that the publisher is authenticated by the SecServ and authorized to publish data to a topic. When the publisher wants to publish topic data, it first sends a join message to the RS over its DN using the topic identifier (step 1). The RS looks up its database for the topic-to-DB mapping. If such a mapping exists, the RS sends the responsible topic-to-DB mapping to the DN which installs a forwarding entry for that topic in its internal forwarding table (step 2). The publisher starts pushing data to its DN which forwards it to the responsible DB which stores the topic data (step 3).

2) Subscription to Topic Data: Topic data retrieval works similar. Initial message exchange prior to topic data retrieval is shown on the right side of Fig. 1. We again assume that the subscriber is authenticated by the SecServ and authorized to retrieve data of the topic. When the subscriber wants to retrieve topic data, it first sends a join message to the RS over its DN using the topic identifier (step 1). At the same time, the DN installs a topic-to-client entry in its internal forwarding table. The RS looks up its database for the topic-to-DB mapping. If such a mapping exists, the RS forwards the join message to the responsible DB which installs a topic-to-subscriber's-DN entry in its internal forwarding table (step 2), and starts pushing topic data to all registered subscriber's DNs (step 3).

3) Monitoring and Control of the C-DAX Cloud: Any C-DAX node is a publisher to a special *monitoring* topic and publishes its node state information to that topic. This information is gathered and aggregated by the MonSys, which is a subscriber of this topic (step 4 in Fig. 1). The MgmSys issues management commands to individual C-DAX nodes in order to perform topic and node management operations (step 5 in Fig. 1).

D. Communication Modes

C-DAX supports three different communication modes: streaming, query and point-to-point. In *streaming communication mode*, subscribers continuously receive topic data after successfully joining a topic without requiring further explicit requests. In *query communication mode*, subscribers have to send explicit topic data queries to fetch specific topic data, e.g., a snapshot of streamed data. In *point-to-point communication mode*, publishers send data directly to subscribers without DNs and DBs involved in the actual data transmission. The latter mode violates the ICN paradigm but may be necessary for use cases requiring extremely low latency. Modes are set per topic to fit the requirements of the application, e.g., low latency for PMUs or improved scalability for retail energy transactions on the retail energy market [22].

E. Security Concept

C-DAX security rationales are strong authentication of clients and nodes based on asymmetric cryptography, end-to-end security for topic data, minimal trust in the underlying infrastructure, and a flexible match of security parameters to the requirements of use cases. C-DAX nodes do not have to trust each other for secure operation, and clients do not have to trust the C-DAX cloud for guaranteed end-to-end security.

F. Inter-Domain Concept

C-DAX enables utilities to cluster their infrastructure into *C-DAX domains*, i.e., sets of components of the same jurisdiction. Direct communication between clients and nodes of different domains may be restricted, e.g., due to business reasons, laws, operations rules, or security. Each domain operates C-DAX DNs at its domain borders which provide a uniform interface for external subscribers, and hide the domain's network. DNs are responsible for forwarding inter-domain traffic, and for

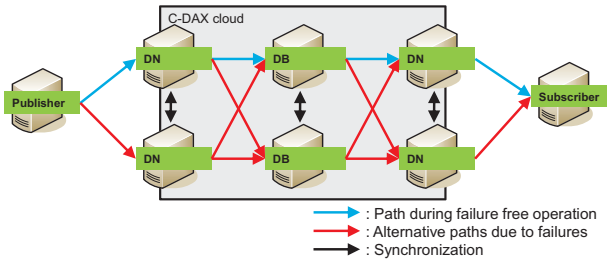


Fig. 2. The C-DAX resilience concept. Topic data is stored on two DBs. Each critical communication path is divided into a path during failure free operation (top) and alternative paths due to failures (bottom).

enforcing inter-domain security policies. A domain operator may operate multiple DNs to balance inter-domain traffic. The SecServ of the each domain manages the respective rights for its internal and external subscribers.

IV. Resilience Concept

We now describe the resilience concept of the C-DAX architecture. We first discuss the design rationale behind the concept, and the envisioned resilience support levels. Then we specify the required signaling, and finally depict actions upon node failure detection.

A. Design Rationale

Topic data should be highly available to SG applications, even in case of C-DAX component failures. In addition, resiliency should be transparent to and configurable by the actual SG application. Fig. 2 shows the basic idea of the resilience concept in the C-DAX architecture. Component and data redundancy yields a simple yet robust resilience concept, enabling the infrastructure to survive in case of any component failure. Robustness here means that C-DAX should be able to cope with single component failures without additional communication with the MgmSys. Each client is configured with at least a primary and backup DN with whom it may communicate, and each topic is stored on at least a primary and backup DB. Each critical communication path is divided into a path during failure free operation (top paths in Fig. 2) and alternative paths due to failures (bottom paths in Fig. 2). Node failure detection is based on a heartbeat mechanism which we will elaborate on in Section IV-C.

B. Resilience Support Levels

A SG application may tolerate data loss, data delay, and failover delay to some extent. *Failover delay* includes the time of the failure detection and the successful failure recovery. It gives the lower bound of service unavailability time in case of a failure which must be dealt with by the SG application. *Data delay* means that time-stamped data may not be delivered with the original data rate. Reasons for data delay may be, e.g., intermediary buffering, network congestion, or retransmissions. *Data loss* means that topic data sent by publishers is not received by subscribers. Reasons for data loss may be, e.g., node failures and network failures.

TABLE I
Overview on C-DAX resilience support levels.

Level	Data loss (during failover)	Data delay (during failover)	Complexity
RSL-0	Y	Y	Low
RSL-1	Y	N	Low
RSL-2	N	Y	Middle
RSL-3	N	N	High

Component and data redundancy allows for several meaningful communication patterns between publishers and subscribers. Depending on the communication pattern, different levels of resilience quality can be realized, which we summarize under the generic term *resilience support levels* (RSLs). We define four different RSLs as listed in Table I, and describe them in detail in the following. RSLs are configured per topic during topic creation time.

a) Resilience Support Level 0: No Resilience: For completeness, we include RSL-0 as the no resilience mode of C-DAX. There are certainly use cases where resiliency may not be necessary because the underlying applications can cope with temporary service degradation. Topics in RSL-0 are only stored on the primary DB, i.e., there are no backup DBs for topics. If the primary DB fails, data forwarding is interrupted until the DB problem is resolved, e.g., by restarting the failed DB, or by moving the topics to a non-failed DB.

b) Resilience Support Level 1: Data Loss Possible: RSL-1 is the simplest resilience mode of C-DAX, and it is the least complex RSL with regard to signaling and provisioning. In contrast to RSL-0, topic data is stored on primary and backup DBs. Topic data is sent unreliably from publishers over the C-DAX cloud to subscribers. Should any intermediary node between publishers and subscribers fail, topic data will be dropped until the upstream node of the failed node switches to a configured backup node. That means, data loss depends on the response time of the node failure detection mechanism. The advantage of this RSL is that neither publishers nor intermediary nodes need retransmission buffers, i.e., it is cheap to implement.

c) Resilience Support Level 2: No Data Loss, But Delays Possible: RSL-2 builds on top of RSL-1 and adds reliable data transmission. Topic data is now sent reliably from publishers over the respective primary DNs and primary DBs to the subscribers. Should any intermediary node between publishers and subscribers fail, topic data will be buffered at the upstream nodes of the failed node. After the upstream nodes successfully switched over to a pre-configured backup node, they re-send the buffered topic data to the backup node. Subscribers will not notice data loss but may experience data delay during the switchover process. That means, the experienced data delay depends on the response time of the node failure detection mechanism. Compared to RSL-1, RSL-2 requires more resources because retransmission buffers are necessary at publishers and intermediary nodes. Still, well-considered placement of topics on primary and backup nodes may allow for efficient backup capacity sharing.

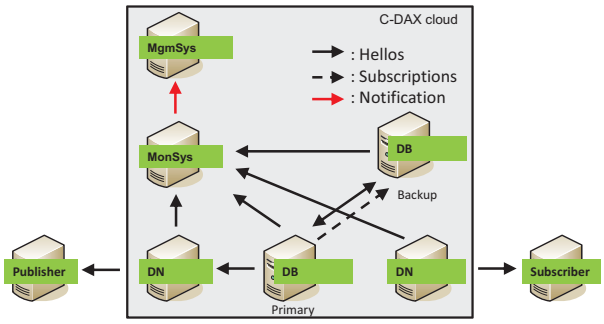


Fig. 3. Resilience signaling in C-DAX. Node failure detection is based on a heartbeat mechanism using periodic hello messages. Missing hello messages indicate node failures. Primary and backup nodes synchronize their subscriptions to guarantee smooth switchovers.

d) Resilience Support Level 3: Near Real-Time Resilience: RSL-2 is an improvement to RSL-1 with regard to data loss. Still, data delay may be a problem for near real-time SG applications. Using RSL-2 for such applications would require a very fast and highly reliable node failure detection mechanism which may itself introduce significant signaling load on the communication substrate. We therefore propose RSL-3 for near real-time resilience.

The key concept behind RSL-3 is simultaneous topic data transmission on disjunct data paths from publishers to subscribers. Within the limits of the system, RSL-3 provides reliable topic data delivery and close-to-zero data delay. Prerequisites for RSL-3 are perfect subscription synchronization of primary and backup nodes, and appropriate provisioning of the communication substrate. During failure-free operation, subscribers receive all topic data twice and perform duplicate data removal before handing the data over to the SG application. Should any intermediate node fail, data is still delivered to the subscriber. RSL-3 is the most expensive and complex solution compared to RSL-1 and RSL-2 because it also requires careful communication substrate planing and provisioning.

C. Node Failure Detection

Node failure detection has direct impact on the performance of RSL-1 and RSL-2. The involved components and the necessary signaling of the node failure detection mechanism of C-DAX are shown in Fig. 3. It is implemented using hello messages and timers. That means, one component is periodically sending hello messages and another component is receiving hello messages. After the reception of a hello message, the receiving component starts an internal timer. When the receiving component receives another hello message from the same sending component before the timer expires, the sending component is considered alive, the timer is restarted and the receiving component awaits the next hello message. When the timer expires before another hello message is received, the sending component is considered failed, and a failure event is raised at the receiving component. The timer value at the receiving component, called vulnerability window, has to be set carefully because network disruptions may cause hello messages to be dropped during regular operation, too. Otherwise, the receiving component may falsely assume a failed sending component.

Hello message signaling is applied in C-DAX as follows. All cloud nodes periodically send hello messages to the MonSys. In case of DNs, this information is only logged for monitoring purposes. In case of DBs, additional steps may take place should a node failure be detected, e.g., determination and selection of a new primary or backup DB for the failed DB, triggering of topic migration operations to make the system ready for the next DB failure, and notification of the MgmSys. Clients receive hello messages from their connected DNs. This allows for a faster switchover to a backup DN should the primary DN fail compared to periodically querying the DN for availability. The MgmSys selects primary and secondary DBs at topic creation time while primary nodes synchronize subscriptions with backup nodes during operation, as will be elaborated in the following. In the latter, nodes refers to both DBs and DNs.

D. Subscription Synchronization

Subscription synchronization among primary and backup nodes yields fast node switchover without service degradation should the respective primary node fail because all necessary forwarding information is already available at the backup node. The subscriptions for a topic are stored on primary and backup DBs, and forwarding state is synchronized between the primary and backup DNs as well. There are several possible implementation options for subscription synchronization. One approach is to include proactive synchronization in the client join and leave process, e.g., clients send their join messages to the primary and backup nodes, which in turn have to know if they are the primary and backup node for the requested topic. When clients leave the cloud, their subscriptions are removed from any respective node. Another approach is to have a reactive synchronization signaling scheme in place, i.e., primary nodes in the cloud update the state of the backup nodes whenever a change in the subscriptions or forwarding occurs. This is also necessary when topics shall be migrated to different DBs inside the cloud. However, describing topic migration is out of scope of this paper. For the prototype implementation, we used the proactive subscription synchronization.

E. Actions Upon Failure Detection

C-DAX provides autonomous operation of the system should primary or backup nodes fail with minor service degradation, and with only limited interaction with the MgmSys. We now describe the actions that take place upon failure detection.

1) Primary DB Fails: When the primary DB fails, the MgmSys promotes the backup DB to the new primary DB. Then, the MgmSys selects a new backup DB and informs the new primary DB. The new primary DB synchronizes its subscriptions with the new backup DB. Publishers' DNs, aware of the primary DB failure, may query the RDS/RS for the new backup node, and now send their data to the new

primary DB. Alternatively, the new primary DB may notify the publishers' DNs about the new backup DB. Eventually, the new primary DB sends the data to the subscriber DNs.

2) Backup DB Fails: When the backup DB fails, the MgmSys selects a new backup DB and informs the primary DB. The primary DB synchronizes its subscriptions with the new backup DB. Publishers' DNs, aware of the backup DB failure, may query the RDS/RS for the new backup node, but continue to send their data to the primary DB. Alternatively, the primary DB may notify the publishers' DNs about the new backup DB.

3) Primary and Backup DB Fail Simultaneously: When both DBs fail simultaneously, the subscriptions are temporarily lost. In that case, the MgmSys selects a new primary and backup DB for the topics, and the publishers' and subscribers' DNs re-register via the RDS/RS and receive information about the new DBs.

4) DN of Publishers Fails: When the primary DN of a publisher fails, the publisher may switch over to its backup DN, and send its data to the backup DN. Should the backup DN of a publisher fail instead, the publisher will notice this event, but not take any further actions. To make publisher more robust against DN failures, it may be configured with more than two DNs.

5) DN of Subscribers Fails: When the primary DN of a subscriber fails, the subscriber will receive topic data from its backup DN instead. When the backup DN of a subscriber fails, the subscriber will continue to receive topic data from its primary DN. No additional signaling is necessary from the subscriber's perspective. Like for publishers, subscribers can be made more robust against DN failures by configuring more than two DNs.

6) Cloud Core Components Fail: Cloud core components are central components of the cloud and can fail as well. In C-DAX, this includes the MgmSys, the MonSys, the RDS/RS, and an initial set of DBs and DNs. In order to avoid a single point of failure, a redundant array of cloud core service nodes is operated which synchronizes its information. Thus, topic-to-RS and topic-to-DB mapping information is highly available.

F. Protected Failures

C-DAX' resilience mechanism primarily addresses the failure of DBs which are needed as forwarding nodes in a classical pub/sub system. Network failures such as link, switch, or router failures, should be rather protected by re-routing mechanisms. However, C-DAX' resilience mechanism can also limit the impact of a network failure when the network breaks into disconnected islands. Then, communication is possible among all C-DAX clients and nodes that still have a working path via reachable primary or backup DB.

V. Performance Evaluation

We now investigate the performance of the presented resilience concept by experimentation with the C-DAX prototype. The setup of experiments is described first, followed by experimental results from traffic experiments during failure-free operation and during a DB failure. Our results show data throughput for C-DAX before, during, and after a DB failure, and further demonstrate that our mechanism performs fast and reliably.

A. Experiment Setup and Methodology

To evaluate the performance of the resilience concept, we created a dumbbell-like topology with one publisher on the left side, the C-DAX cloud in the middle, and one subscriber on the right side. The C-DAX cloud is configured with one DN for publisher and subscriber each, and two DBs; the current prototype implementation supports RSL-2 only. We created one topic for PMU measurement data to which the publisher and the subscriber join. We used recorded IEEE C37.118-compliant [15] PMU measurement data provided by EPFL as realistic workload; the publisher replayed the data set and sent 50 packets per second, and one interleaved configuration frame every 60 seconds.

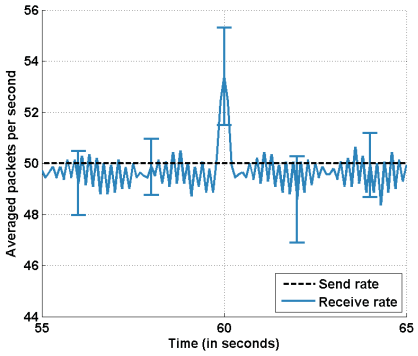
We deployed our setup on a dedicated network testbed with 100 Mbit/s link bandwidth, and measured the data throughput of the C-DAX cloud at the subscriber side. This enabled us to measure the time and quality of service degradation during the actual DB switchover. Our data throughput measurement method is based on packet arrival timestamp sampling. We first log the time of each packet arrival at the subscriber. Then, we sample the recorded timestamps with a higher frequency than the send rate, i.e., we count the number of packet arrivals during one sample period, and retrieve the receive rate. We performed each experiment 50 times with each experiment running for 70 seconds, averaged the throughput measurements, and show the 95% confidence intervalls.

B. Failure-Free Operation

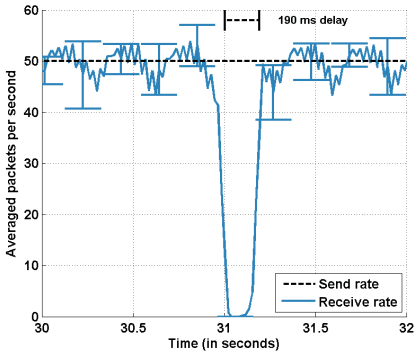
We first assume that no nodes fail. We start the data replay at the publisher and measure the data throughput at the subscriber. The results are shown in Fig. 4(a); the dashed line represents the send rate of the publisher. The subscriber receives topic data with a rate of 50 packets per second with a small peak at 60 seconds as expected. We recognize fluctuations in the data throughput which stem from the network substrate of the network testbed. We use these values as a benchmark for failure-free operation.

C. One DB Failure

We re-use our previous experiment setup and emulate the failure of a DB during regular operation. First, all nodes and clients are started, the publisher replays the data, and we wait until we have a stable receiving rate at the subscriber. After 30 seconds, we disconnect the primary DB of the PMU measurement topic, and measure the time until the receiving rate at the subscriber is stable again, i.e., until the switchover finished successfully. The results are shown in Fig. 4(b); the dashed line represents the send rate of the publisher. Before the DB failure, the subscriber receives topic data with an average rate of 50 packets per second; these results are in-line with



(a) Failure-free operation.



(b) One DB failure after 30 seconds.

Fig. 4. Averaged packet receive rate at the subscriber side including 95% confidence intervalls. The dashed line represents the send rate at the publisher side. The peak at 60 seconds is part of the IEEE C37.118 PMU communication protocol [15] and represents a periodically interleaved configuration frame.

our measurements during failure-free operation. During the switchover, we can see a drop in data throughput down to 0 packets per second. After successful switchover, the data throughput returns to the same level as before the DB failure. The switchover time is less than 190 milliseconds. This shows that our proposed mechanism works fast and reliably.

VI. Related Work

Van Jacobsen et al. [23] propose in-network caching and solicited retransmissions as resilience concept for their content-centric networking (CCN) architecture. In CCN, data is routed over the CCN overlay from the publisher to the subscriber; intermediary CCN nodes may cache the forwarded data. Subscribers actively trigger retransmission of data should a timeout on the receiver's side occur. Intermediary CCN nodes may reply on retransmission requests immediately to speed up retrieval of lost data. Other ICN architectures such as DONA [24], 4WARD [25], PSIRP/PURSUIT [26], SAIL [27], and COMET [28] use a similar approach.

The SeDAX[20], [29] architecture uses geographical routing on a Delaunay-triangulated (DT) overlay network to forward messages to the responsible broker. Geographic hashing assigns static overlay network coordinates to topics. Topic data is stored on adjacent primary and backup brokers which are closest and second-closest to the topic's coordinate to make the system resilient against node failures. After a node failure, the backup broker takes over automatically, and the overlay reconfigures itself to restore the overlay DT properties and heal the forwarding. Storage requirements for resilient SeDAX operation have been investigated in [30], and an extension to support distributed load balancing has been proposed in [31].

The OMG Data-Distribution Service (DDS) [32] is a pub/sub architecture which targets real-time communication. Direct point-to-point connections between publishers and subscribers yield minimal delay and latency, i.e., no brokers are involved in the communication. DDS proposes the concept of *data-stream ownership* [33] to provide fault tolerance and automatic failover. That means, publishers and subscribers communicate over so-called data-streams which have owners with pre-configured ownership-strength assigned. In case of node failures, the next-strongest data-stream owner takes over.

The DataTurbine [34] pub/sub architecture proposes a ring buffer network bus comprised of either a single broker or a federated set of brokers. Brokers can be mirrored to make the system resilient against network failures but failover between brokers is not intended.

The ZeroMQ [35], [36] high-performance asynchronous messaging library provides a message queue for scalable distributed and concurrent applications. It can be operated without a broker, but offers a framework to introduce brokers and failover mechanisms. ZeroMQ can be used to build complex pub/sub architectures, using socket polling and heartbeating for reliable node failure detection, and primary-backup server pairs to provide high-availability.

Java Message Service (JMS) [37] is an API for message oriented middleware specified as part of the Java EE Platform. JMS supports both point-to-point and pub/sub messaging modes. While resilience mechanisms are not defined in the JMS API, common implementations like Apache ActiveMQ [38] or Oracle Glassfish [39] provide high-availability schemes. Clients can be configured with a set of brokers to automatically switch brokers in case of a failure.

The distributed messaging platform NSQ [40] uses a redundant set of brokers which are normally co-located with their publishers, and discovered by subscribers using a redundant set of resolvers; deploying publishers and brokers on different hosts is possible as well. The knowledge of the resolvers is so-called *eventually consistent*, i.e., joining clients have to query all their configured resolvers to find all responsible brokers for a certain topic. Each broker forwards received topic data to all interested subscribers. Subscribers have to handle duplicate message reception themselves. The architecture provides means for reliable data transfer using acknowledgments, and in-network caching.

VII. Conclusions

The C-DAX project aims at providing and investigating a communication middleware for SGs to address the limited scalability, reliability, and security of todays utility communication infrastructures. In this paper, we briefly described the C-DAX architecture and extended it with four resilience

support levels: no resilience (RSL-0), with packet loss during switchover (RSL-1), with packet delay but without packet loss during switchover (RSL-2), and without packet loss and delay during switchover (RSL-3). RSL-0 and RSL-2 are implemented in the C-DAX prototype, and we presented measurement data of the switchover process for RSL-2. The prototype will also be used in the C-DAX field trial. For the future, we envision an integration of all resilience support levels in the prototype.

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 Cynthia Mills, Mario Paolone, Teklemariam Tesfay, Marcel Mampaey, Herman Bontius, Michel Hasz, and Sebastian Veith for valuable input and stimulating discussions.

REFERENCES

[1] CIGRE Working Group C6.11, "Development and Operation of Active Distribution Networks," Apr. 2011.

[2] G. T. Heydt, "The Next Generation of Power Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, 2010.

[3] R. Singh, B. Pal, and R. Jabr, "Choice of Estimator for Distribution System State Estimation," *IET Generation, Transmission & Distribution*, vol. 3, no. 7, 2009.

[4] J. Liu, J. Tang, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Trade-Offs in PMU Deployment for State Estimation in Active Distribution Grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, 2012.

[5] S. Sarri, M. Paolone, R. Cherkaoui, A. Borghetti, F. Napolitano, and C. Nucci, "State Estimation of Active Distribution Networks: Comparison between WLS and Iterated Kalman-Filter Algorithm Integrating PMUs," in *IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*. IEEE, 2012.

[6] A. Borghetti, C. A. Nucci, M. Paolone, G. Ciappi, and A. Solari, "Synchronized Phasors Monitoring During the Islanding Maneuver of an Active Distribution Network," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, 2011.

[7] "D1 Requirements for different Smart Grid Applications, SG Network Systems Requirements Specification V4.0," NIST Smart Grid Interoperability Panel PAP01: Role of IP in the Smart Grid, Nov. 2009.

[8] C-DAX Consortium, "Cyber-secure Data And Control Cloud for Power Grids," 2014. [Online]. Available: http://www.cdax.eu/

[9] 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.

[10] 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, pp. 114 – 131, Jun. 2003.

[11] P. Romano and M. Paolone, "Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype," *IEEE Transactions on Instrumentation and Measurement*, 2014.

[12] L. Zanni, M. Pignati, S. Sarri, R. Cherkaoui, and M. Paolone, "Probabilistic Assessment of the Process-Noise Covariance Matrix of Discrete Kalman Filter State Estimation of Active Distribution Networks," in *International Conference of Probabilistic Methods Applied to Power Systems (PMAPS)*, Durham, UK, 2014.

[13] M. Pignati, L. Zanni, S. Sarri, R. Cherkaoui, J.-Y. L. Boudec, and M. Paolone, "A Pre-Estimation Filtering Process of Bad Data for Linear Power Systems State Estimators Using PMUs," in *Power Systems Computation Conference (PSCC)*, Wroclaw, Poland, 2014.

[14] "IEEE Standard for Synchrophasor Measurements for Power Systems," IEEE C37.118.1-2011, December 2011.

[15] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," IEEE C37.118.2-2011, December 2011.

[16] P. T. Myrda and K. Koellner, "NASPInet - The Internet for Synchrophasors," in *International Conference on Systems Sciences (HICSS)*. IEEE, 2010.

[17] North American Synchro-Phasor Initiative, "Data Bus Technical Specifications for North American Synchro-Phasor Initiative Network," 2009. [Online]. Available: https://www.naspi.org/File.aspx?fileID=587

[18] Power System Operation Corporation Limited, "Synchrophasors Initiative in India," Power System Operation Corporation Limited, Power Grid Corporation of India Limited, New Delhi, India, technical report, Dec. 2013.

[19] P. Bellavista, A. Corradi, and A. Reale, "Quality of Service in Wide Scale Publish-Subscribe Systems," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, 2014.

[20] 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.

[21] K. V. Katsaros, W. K. Chai, N. Wang, G. Pavlou, H. Bontius, and M. Paolone, "Information-Centric Networking for Machine-to-Machine Data Delivery A Case Study in Smart Grid Applications," *IEEE Network, Special Issue on Information-Centric Networking Beyond Baseline Scenarios: Research Advances and Implementation*, vol. 28, no. 3, 2014.

[22] M. Hoefling, F. Heimgaertner, B. Litfinski, and M. Menth, "A Perspective on the Future Retail Energy Market," in *Workshop on Demand Modeling and Quantitative Analysis of Future Generation Energy Networks and Energy Efficient Systems (FGENET)*, Bamberg, Germany, Mar. 2014.

[23] 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.

[24] 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.

[25] P. A. Aranda et al., "Final Architectural Framework," Jun. 2010. [Online]. Available: http://www.4ward-project.eu/

[26] D. Trossen et al., "Conceptual Architecture: Principles, Patterns and Sub-components Descriptions," May 2011. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[27] "Scalable and Adaptive Internet Solutions (SAIL)." [Online]. Available: http://www.sail-project.eu/

[28] "COntent Mediator architecture for content-aware nETworks (COMET)." [Online]. Available: http://www.comet-project.org/

[29] 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 International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2012.

[30] M. Hoefling, C. G. Mills, and M. Menth, "Analyzing Storage Requirements of the Resilient Information-Centric SeDAX Architecture," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Vancouver, Canada, Oct. 2013.

[31] ——, "Distributed Load Balancing for Resilient Information-Centric SeDAX Networks," in *IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014.

[32] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural Overview," in *ICDCS Workshops*, 2003, pp. 200–206.

[33] S. Schneider and B. Farabaugh, "Is DDS for You?" RTI Whitepaper, 2009.

[34] S. Tilak, P. Hubbard, M. Miller, and T. Fountain, "The Ring Buffer Network Bus (RBNB) DataTurbine Streaming Data Middleware for Environmental Observing Systems," in *IEEE Conference on e-Science and Grid Computing*, 2007.

[35] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. O'Reilly Media, Inc., 2013.

[36] ——, "Zeromq - the guide," http://zguide.zeromq.org/page:all, 2013.

[37] Oracle Corporation, "Java Message Service 2.0 Released," 2013. [Online]. Available: http://www.oracle.com/technetwork/java/jms/index.html

[38] The Apache Software Foundation, "Apache ActiveMQ 5.9.0 Released," 2013. [Online]. Available: http://activemq.apache.org/

[39] Oracle Corporation, "GlassFish Server 4.0 Released," 2013. [Online]. Available: https://glassfish.java.net/

[40] M. Reiferson and J. Czebotar, "NSQ - a real-time distributed messaging platform," 2014. [Online]. Available: http://nsq.io/