



Workshops der  
Wissenschaftlichen Konferenz  
Kommunikation in Verteilten Systemen 2009  
(WowKiVS 2009)

Data Centric Peer-to-Peer Communication in Power Grids

Christoph Gerdes, Kolja Eger and Jörg Müller

12 pages

# Data Centric Peer-to-Peer Communication in Power Grids

Christoph Gerdes<sup>1</sup>, Kolja Eger<sup>1</sup> and Jörg Müller<sup>2</sup>

<sup>1</sup> [c.gerdes](mailto:c.gerdes), [kolja.eger@siemens.com](mailto:kolja.eger@siemens.com)

Siemens AG, Corporate Technology, Information and Communications

<sup>2</sup> [joerg.mueller@tu-clausthal.de](mailto:joerg.mueller@tu-clausthal.de)

Clausthal University of Technology

**Abstract:** We study the use of peer-to-peer based declarative data management to enable efficient monitoring and control of power transmission and distribution networks. We propose methods and an architecture for data centric communication in power networks; a proof-of-concept decentralized communication infrastructure is presented that uses and advances state of the art peer-to-peer and distributed data management protocols to provide real time access to network state information. We propose methods for adaptive network reconfiguration and self-repair mechanisms to handle fault situations. To efficiently handle complex queries, we present a centralized metadata index, and propose a query language and execution method that allows us to handle high volume data streams in-network.

**Keywords:** Peer-to-Peer, Complex Query Processing, Anomaly Detection, Wide Area Monitoring

## 1 Introduction

Electricity networks belong to the class of the largest machines ever built. Consisting of thousands of sensors and actuators distributed over large distances, these networks exhibit complex real-time dynamics that has to be controlled to achieve acceptable levels of power quality. Traditionally, electricity networks were strictly hierarchical with a top-down flow of electric power from a few large power plants down to a large number of consumers. This situation is changing dramatically as power generation shifts from a central structure towards distributed generation.

The new distributed power generation scenario raises the need for advanced communication paradigms in order to maintain a stable balance of generation and consumption. Of particular importance is the real-time availability of data collected at various distributed nodes. The raw data needs to be processed and aggregated to valuable information such that the current system state can be assessed and stabilizing control decisions can be made. To meet these new challenges, new methods and architectures for information and communication infrastructures are required.

In this paper, we present the architecture of a data centric communication infrastructure for power networks and corresponding decentralized indexing, routing, and stabilizing methods that use and advance state of the art peer-to-peer (P2P) [SMK<sup>+</sup>01] and distributed data management protocols. By positioning our work on the application layer, we directly address the specifics of power networks that are highly heterogenous and contain a broad variety of technologies and standards, some of which have been in operation for several decades.

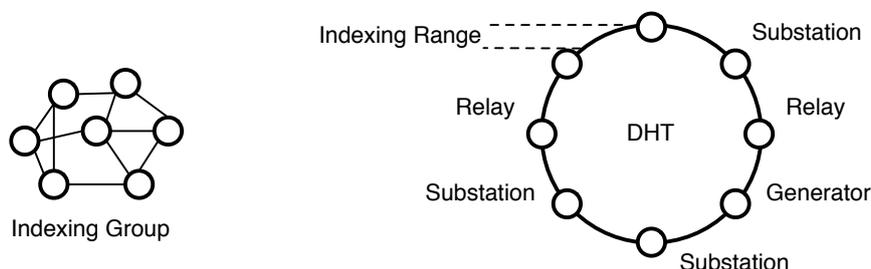


Figure 1: Platform macro architecture. Transmission and distribution networks units build a DHT. Dedicated indexing nodes provide a system wide metadata index.

The contribution of this paper is threefold: firstly we adapt and extend P2P protocols, originally designed for the distribution of digital content, for the industrial domain with its special requirements. Secondly we establish a data centric view on wide area transmission and distribution systems that dramatically reduces complexity of control tasks. Finally, aiming at real-world applicability, we present a pragmatic and constructive approach for current and emerging energy control systems.

The structure of this paper is as follows: Core concepts of our architecture are described in Section 2. Subsequently we elaborate the underlying indexing and query execution model, evaluate the benefits of our approach, and present simulation results. In Section 4 we relate to past and ongoing research in global sensor networks (GSN), P2P systems and distributed databases. We conclude by giving the status of our work and an outlook.

## 2 Core Concepts

The architecture includes the entire transmission and distribution network with components hosted by various entities such as energy automation equipment and supervisory control and data acquisition (SCADA) systems. Instead of trying to pull all data into a central database, the we provide methods to enable in-network distributed processing of data.

Our approach is to provide declarative data management as common in most standard database products. However, by using internet-proven peer-to-peer protocols the architecture is capable to scale to a very high number of nodes potentially distributed over large distances. Since direct end-to-end addressability is an unrealistic assumption in global networks, all nodes build a distributed hashtable (DHT). The DHT spans a global address space and enables each peer to directly address any other peer within the global network. Meta data describing node capabilities and services are stored in centralized metadata indices at dedicated indexing nodes. The set of indexing nodes is highly connected to ensure fast data exchange within the indexing group (Figure 1).

Every parameter of a device, sensor or actuator can be accessed like a table in a relational database. By enabling this data centric perspective where data can be retrieved either through declarative description or keywords, the system becomes independent of physical node addresses

```
Event isOutOfBand
  sensor1 = SELECT * FROM sensors
            WHERE location='substation A'

  sensor2 = SELECT * FROM sensor
            WHERE location='substation B'

  if (sensor1.voltage > 110kV OR sensor1.voltage < 90kV)
    return 1
  if (sensor2.voltage > 110kV OR sensor2.voltage < 90kV)
    return 2

  return nil
END

CREATE STREAM isOutOfBand@node5621 WINDOW(now, forever, 1s)
```

#### Program 1: Event definition

and whereabouts. For example: Program 1 demonstrates how measurements can be aggregated to events using the provided query language: first an event is defined which collects measurements of two sensors hosted by two separate substations. If the measurements of either one of the sensors are either above or below given thresholds, the executing node will emit this information to the receiver specified in the following query:

```
SELECT * FROM isOutOfBand RECEIVER(rcv_node)
```

Otherwise, if the value is in limits, nothing is sent. The event is named and assigned by the *create* statement. Thereby *@node5621* means that the event logic is evaluated at a node identified by the string *@node5621*. Additionally, the *WINDOW* identifier specifies the window of activity and the interval updates are being expected. The data centric approach is highly beneficial in large networks such as power networks where a small fraction of nodes is constantly being added or replaced or do fail. Addresses may change or functions may migrate from one physical node to another without the need to reconfigure referencing nodes because the underlying DHT and the metadata index will update automatically.

Industrial installations in general and power networks in particular have strong security requirements. As the architecture is realized on the application layer, we assume a secure environment through private networks and lower level encryption.

## 2.1 Meta Data Index and Global Query Catalog

As stated by many others, e.g., [HHH<sup>+</sup>02], complex queries (i.e., queries that go beyond keyword search), are difficult to support with most peer-to-peer algorithms. Our solution of the problem is centralization of important meta data at dedicated nodes called indexing nodes. Indexing

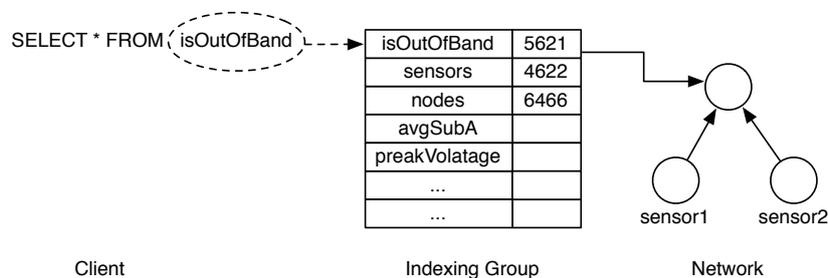


Figure 2: Queries are resolved through the query catalog of the centralized index. The catalog maps to nodes that are responsible for the data requested.

nodes are assigned a dedicated key range of the DHT. The indexing group key range is relatively large; thus the group can be scaled by adding new nodes according to current demand. Metadata updates are sent by picking a node from the key range at random and transferring the update. The receiving nodes disseminate the update internally in the indexing group. In addition to metadata, the indexing group maintains a query catalog which includes all searchable streams and events such as those defined in program 1. The query catalog constitutes the mapping from the query name to the node extracting the data from the network either by aggregating several sources or reading a single value. While conceptually being suited for a DHT, the catalog is centralized to avoid overloading of nodes in the occurrence of popular queries potentially yielding unstable network operations. Figure 2 illustrates the role of the query catalog using the example provided in the previous section.

Data dissemination inside the indexing group is done using an asynchronous gossiping algorithm; data is transmitted to a fixed set of nodes which will in turn forward the data until it has been received by all nodes within the indexing group. Indexing nodes checkpoint their data periodically to a relational database back-end. The back-end serves also as synchronization point for indexing nodes that have been separated from the network due to failure or when they join for the first time. The indexing group does not support strict ACID transactions and follows a weak consistency model.

As data model for the metadata, the object model defined by the IEC 61850 [IEC04] standard for energy automation equipment was adapted. We assume the environment rather static with few replacements, failures and additions of equipment. Consequently the system will have low metadata update rates and we therefore optimize for read operations. Internally indexing nodes maintain multidimensional indices of up to 100 node attributes whereby the node type determines which and how attributes are indexed.

## 2.2 Query Language

Our query model supports both ad-hoc i.e. single result set queries as well as queries initiating data streams. We developed extensions to the SQL to initiate and manage data streams in the network. The language allows us to apply filters on streams and route data throughout the network. We are aware of the restrictions and flaws of SQL as a query language [Dat84]; we chose it due

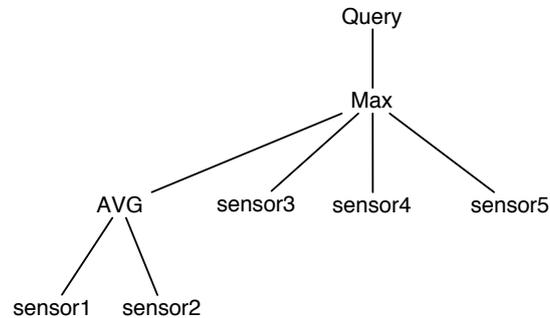


Figure 3: An Example Query

to the wide acceptance among developers and good integration with standard applications, such as JDBC, as well as the general good readability. However, query statements are compiled to a binary format; as soon as they enter the network only this binary format is used. This gives the flexibility to support different query languages by simply changing the compiler; it also improves execution performance and decreases network traffic.

### 3 Query Execution Model

Queries are usually created at a client entity, e.g. human machine interface (HMI) unit, or observe and control terminals. From the client they are passed via gateways to the network where they are executed. Clients receive intermediate and final results of the execution in the form of result sets.

Queries can be injected into the network through known gateways nodes that provide public interfaces (e.g., web services or SQL database drivers). Once injected, queries are parsed and compiled into binary format. In a subsequent analysis an execution plan is created. The plan includes instructions to be executed either locally or remotely to extract and deliver the in the query specified information. Instructions of the plan include resource lookup and discovery, query optimization and sub query creation, query reconfiguration, sub query assignment as well as result delivery.

To illustrate query processing and analysis, consider query depicted in Figure 3. This query states the interest to retrieve the maximum of the values of the sensors three, four, and five, and of the average of the measured value of sensor one and two. The client does not need to have any a priori knowledge about the network topology. All five sensors could be part of just one local network, e.g., within a substation; they could be connected each to a different network, or their connection could be a combination of the first two cases. Furthermore, it is unnecessary for the client to know how the networks are interconnected, which network address they have or whether the measured values are generated on the fly by micro devices out in the field, or are fetched from a database.

On arrival of a query request, it is checked whether the access and location of all in the request specified data sources is known. If there are unknown sources, the first instructions of the plan are lookup requests to the metadata index. Once all sources are known, the query expression

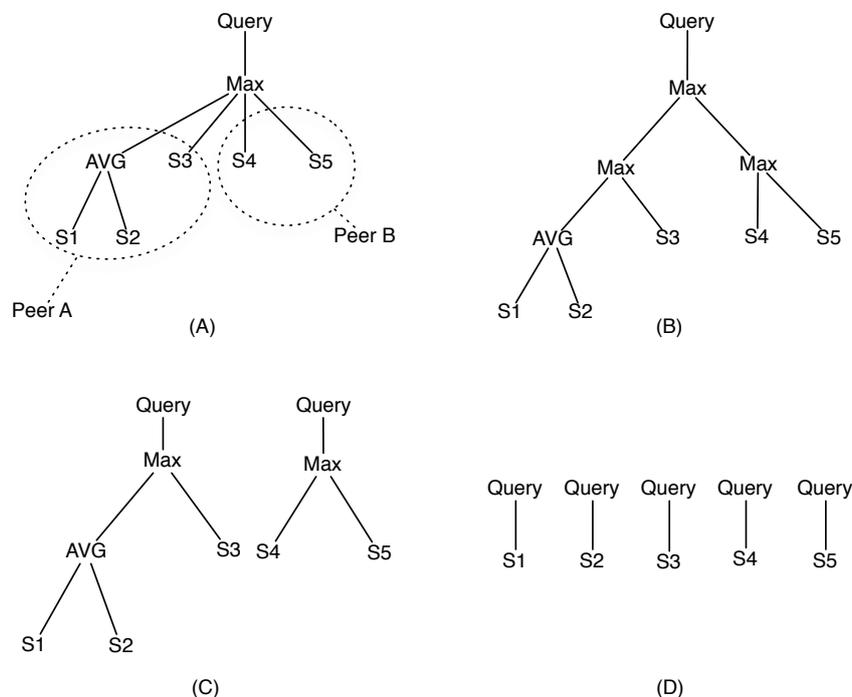


Figure 4: Query Reconfiguration

itself is analyzed. The aim of this analysis step is to reconfigure the expression such that if it includes sources which are not locally available are summarized in subqueries to be sent to respective remote peers. In part A of Figure 4, sensors 1-3 have been resolved to be available at peer A while sensors 4 and 5 are available at peer B. In step B the query is reconfigured pushing the MAX operators for execution to peer A and B respectively. In step C, sub-queries are created and assigned to peer A and B. Eventually they are scheduled locally as per sensor query as illustrated in step D.

Once the execution plan has been created, it is queued in a local scheduler. Execution will begin once resources are available or the specified start time has been reached. Queries in execution are registered in a local query manager and—in case of streams and events—in the central query catalog. The local manager monitors execution and applies necessary cleanup operations once the query has completed or failed.

Execution plans are not static but continuously adapt to changes in the network. E.g., if a node responsible for processing the aggregation of a sub query is overloaded the data stream can be rerouted to an alternative node with more resources available or the query can be further sub divided to more nodes if they join the network. In related literature [SAL<sup>+</sup>96] [OV99], several methods for query optimization have been introduced. Our method is limited to finding alternative data sources once the original peer did fail. Query writers, however, can include their own code to implement more sophisticated optimization algorithms.

### 3.1 In-Network Processing

High frequency sampling of measurements yield high volume data streams. It therefore is rather suboptimal to transfer all data to a single location to be processed. With the domain specific programming language, we can process data close to where it was sampled. Extracted key parameters can then be sent to other nodes for further processing, stored in archives or visualized at control centers. A detailed description of the language is beyond the scope of this paper. It should be noted that it is declarative with imperative elements supporting basic operators and statements found in almost all programming languages. Program 2 displays a very basic example, implementing a PID controller.

```
FUNCTION PID(Pv, Sp)

    Kp -> 100
    Ki -> 0.9
    Kd -> 1000

    Error -> Sp - Pv
    @TotalError -> TotalError + Error

    Pgain -> Kp * Error
    Igain -> Ki * TotalError
    Dgain -> Kd * (Error - @Derror)
    @Derror -> Dgain

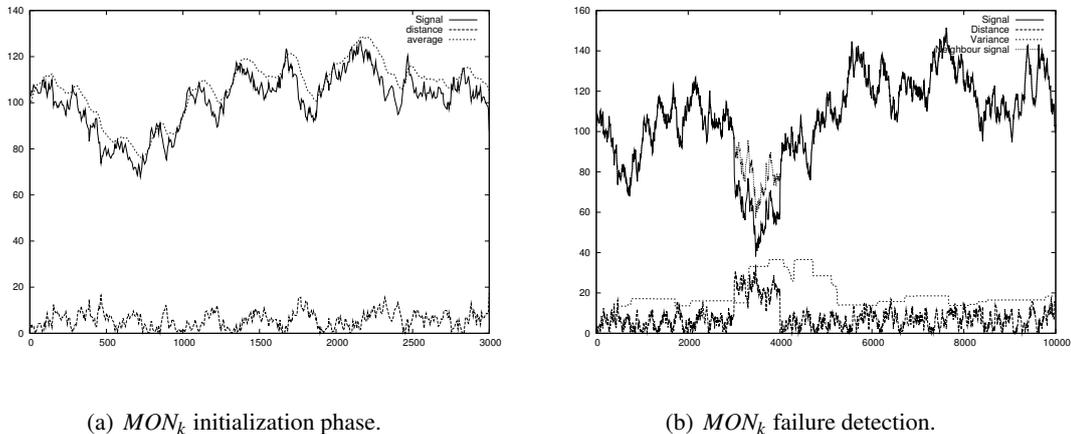
    return Pgain + Igain + Dgain

END
```

#### Program 2: Basic PID controller

In addition to the basic syntax, the language features a set of high level statements for power network specific operations, e.g. *DFT* for *Discrete Fourier Transform* and *MON<sub>k</sub>*, a declarative operator to monitor groups of sensors and detect anomalies. In the following section we use the *MON<sub>k</sub>* operator to illustrate the capabilities and benefits of our architecture. Due to space limitation we omit a discussion on the source code level but rather introduce the conceptual approach and the realization using our query language.

*MON<sub>k</sub>* has been designed to extract relevant information from a large set of globally distributed data sources. In the context of electricity networks it can be used to monitor high voltage power lines or transformers in substations. With the information extracted by *MON<sub>k</sub>*, relays or other control equipment can react to fault situations or take action to optimize network configuration. The operator executes as follows: It first partitions a set of given nodes into sub groups. Group membership is determined by some correlation function. Correlation functions can be as simple as the mean of a measured phenomenon or, e.g., more complex the interplay of several different phenomena e.g. voltage and temperature, over a given time interval. Within sub groups, nodes

Figure 5: Simulation with 5000 nodes;  $k=10$  neighbors

gossip information in respect to the change of the measured phenomenon. As illustrated in section 3, the  $MON_k$  operator is distributed to all nodes of sub group where it is executed locally on nodes<sup>1</sup>. Using a distance metric, each node determines its deviation from the common group state e.g. distance of own measurement to the mean over all measurements within its group. Based on this local computation it is decided whether the node belongs to the query result set like specified in Query 1. Here a subset of nodes is selected and feed to the  $MON_k$  operator which uses the average of the measured phenomena as correlation function and the Euclidian distance to determine its deviation from the stable state.

```

NODES -> SELECT * FROM SENSORS WHERE type='voltage'
SELECT MON_k(*, AVG, DIST, 10, {1, 101}) FROM NODES

```

#### Query 1: $MON_k$ query example

The operator initializes at each node by computing the minimal distance to the given initial means ( $\{1,101\}$ ). It then chooses 10 neighbors at random. Collecting their measured values and applying the average function, the initial average is updated. Neighbors with strong deviations from the average are replaced by randomly chosen nodes from the supplied set.

This initialization process and the convergence of the mean is depicted in figure 5a. The topmost curve, representing the mean, follows the signal quite nicely with a small delay. The bottom curve shows the distance from the mean as measured by one randomly chosen node. In an ideal case, the distance should remain almost constant i.e. a horizontal line. However, as elaborated in [ADG<sup>+</sup>07] and [BGPS05], stability of the algorithm depends on various conditions of the execution environment. Therefore the simulation was done using realistic network models with varying transmission latencies. Furthermore we do not assume synchronized clocks at the nodes and  $MON_k$  is capable to handle failure situations like failing nodes and message loss.

<sup>1</sup> This is implemented using the @node notation, see Section 2

The initialization process finishes once sub groups have stabilized, i.e., the difference of the mean between iterations does not exceed a threshold value.  $MON_K$  computes the variance of the past  $i$  measurements and the distance to the mean to determine whether it measured an anomaly. Concretely it computes  $threshold = \frac{var_i}{2} + distance_{t-1}$  and checks whether  $distance_t > threshold$  is true. This process is illustrated in Figure 5b which shows the results of a simulation with 5000 peers with each  $k=10$  neighbors. Around time frame 3000 an anomaly emerges which is quickly reflected by an increase of the distance to the local mean i.e. the bottom curve. The variance is shown as used to evaluate the current state i.e. shifted by  $i$  time frames. Shortly after the increase, the measured value pokes through the variance curve thus marking the beginning of an anomaly.

## 4 Related Work

Surprisingly little research has been published on the application of distributed databases and GSN middleware for electricity networks. We feel that particularly GSNs provide promising approaches to the challenges arising in the distributed power generation scenario. Results reported by Aberer et al. [AHS07] strongly relate to the concepts presented in this paper. Our architecture, however, was heavily influenced by applicability in current industrial systems. Our sensor model is determined by IEC 61850 which supports an integrative approach with current automation equipment. Compiling queries and programs to binary form allows us to run query engines on low cost embedded devices as well as full features desktop PCs.

A good overview of distributed databases is given in [OV99]. [GO03] provides an overview of current research topics on data stream management. In [KBB07] D. Kucuket et al. introduce a streaming database solution to monitor power quality. Mariposa [SAL<sup>+</sup>96] introduces an architecture for wide area distributed databases in perspective of traditional distributed data management systems (DBMS). The approach includes a micro economic paradigm used for query and storage optimization. AURORA [CBB<sup>+</sup>03], STREAM [GGR07], Cougar [GM04] and others discuss general query processing in sensor networks. AURORA allows users to create queries in a graphical representation. STREAM and Cougar extend the SQL with temporal semantics but not provide support through a programming language.

A considerable body of research is available on P2P systems, including distributed hash tables (DHT) [SMK<sup>+</sup>01], [RD01] and gossip based algorithms [BGPS05]. In contrast to P2P systems deployed in the Internet, in industrial systems, we find a more comfortable environment in some respect: We can expect lower churn rates and less problems with network address translation. However, we must adapt algorithms originally designed for media distribution in the Internet to the resource-constrained environment of industrial systems. Hence, issues such as determinism and real-time capabilities must be addressed. In previous work [SGM08] [FMSF05] we showed already the potential of P2P algorithms in industrial domains beyond content distribution.

## 5 Evaluation

The declarative approach and the use of P2P protocols introduces benefits like scalability, resilience, resource virtualization, uniform data access and adaptiveness to the power network do-

main. Reduction of engineering and maintenance costs is crucial; self configuration and adaption to new devices and topologies yield immediate cost reduction for utilities. Efficient management of complex networks will become even more important in the near future. Up to recently it was not necessary for utilities to be able to assess the complete state of their transmission and distribution network. There was a well defined power flow from a few generators to a large variety of consumers. With the emergence of distributed generation, retrieving detailed real time information on the current network state from thousands or even millions of sensors becomes more and more important as, e.g., reversed power flows can circumvent protection systems and yield equipment failure. Among the challenges of retrieving a correct network state is the extraction of information from vast amounts of sensor data. The declarative approach proposed, allows to calculate key performance indicators in-network thus eliminating the need to transfer high volumes of data to a central point. The primitive program 1 presented in section 2 shows nicely how the complexity of the underlying network is hidden. Queries never use physical network addresses but virtual identifiers. Keywords get resolved and queries reevaluated periodically when messages are sent; thus, our approach supports adaptive reorganization in the event of failure or the appearance of better suited nodes. Processing will still work even if small fractions of nodes are unavailable as is common in large networks. Section 3.1 introduced the  $MON_k$  operator and demonstrated the capabilities and benefits of an in-network processing engine. It adapts system dynamics such that the complexity resulting from the interaction of large numbers of network nodes is hidden under a clean function interface with only few parameters.

Using a centralized metadata index is an efficient and entirely pragmatic approach, but also a potential performance bottleneck. We run extensive tests and simulations to optimize the metadata index performance. The testbed consisted of ten commodity PCs each equipped with relatively low end, Intel Pentium IV, processors and 1GB of RAM. Since our work is in an early state of implementation we are not interested in absolute peak transaction rates, but rather how the system adapts to increased node numbers and update rates. As test data the IEC 61850 based configuration of up to 100000 peers was loaded into the indexing group. For each node 20 attributes were indexed. In the simulation, clients continuously inject queries into the network. Each query requires one lookup request to the index based on one indexed attribute.

The metadata index (implemented in Erlang) was capable to support approximately 10000 concurrent queries on a single machine. The average round-trip time for a query was about 400ms which includes parsing, compiling transferring, processing and receiving results, the peak transaction rate was 24330 transactions per second. However this figures may vary according to the network model used. Our simulations are based on the Kings dataset [KG02]. While this performance is quite remarkable certain applications such as protection system might require faster response times. However, the flexibility of the programming language allows the development of specially tailored solutions that, e.g., use caching mechanisms to reduce lookup and processing times.

Our query execution kernel is event-triggered and operates on incoming and outgoing network messages. This design makes it straightforward to transfer the system to the simulation environment by just exchanging the lower level network protocol stacks. Therefore we conclude that our simulation results are significant in real world environments.

## 6 Conclusion and Future Work

In this paper we introduced a data centric communication architecture and corresponding indexing, routing, and stabilization methods for use in decentralized power transmission and distribution networks. We presented the challenges modern power systems are faced with, and how these are addressed by our approach. We showed how declarative data management can contribute to efficient monitoring and control of power networks. By separating indices and high volume data processing, efficient execution of complex queries on networks with large numbers of nodes becomes possible. Simulations show the performance of the indexing group which scales well to 100000 peers and beyond. Taking everything into consideration we are confident that the proposed architecture is highly beneficial for today's power networks as well as other industrial applications that rely on large, globally distributed networks with thousands of nodes.

Future work will study visualization methods to effectively monitor current system state. Additionally we will address advanced load balancing and query optimization techniques in respect to our target systems.

## Bibliography

- [ADG<sup>+</sup>07] L. Alvisi, J. Doumen, R. Guerraoui, B. Koldehofe, H. Li, R. V. Renesse, G. Tredan. How robust are gossip-based communication protocols? *ACM SIGOPS Operating Systems Review* 41(5):14–18, Oktober 2007.
- [AHS07] K. Aberer, M. Hauswirth, A. Salehi. Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks. *Mobile Data Management, 2007 International Conference on*, pp. 198–205, May 2007.
- [BGPS05] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah. Gossip algorithms: design, analysis and applications. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* 3:1653–1664 vol. 3, 13-17 March 2005.
- [CBB<sup>+</sup>03] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, S. Zdonik. Scalable Distributed Stream Processing. In *CIDR 2003 - First Biennial Conference on Innovative Data Systems Research*. Asilomar, CA, January 2003.
- [Dat84] C. J. Date. A critique of the SQL database language. *SIGMOD Rec.* 14(3):8–54, 1984.
- [FMSF05] T. Friese, J. P. Mueller, M. Smith, B. Freisleben. A Robust Business Resource Management Framework Based on a Peer-to-Peer Infrastructure. In *In: Proc. 7th International IEEE Conference on E-Commerce Technology*. P. 215222. 2005.
- [GGR07] M. Garofalakis, J. Gehrke, R. Rastogi. *Data Stream Management: Processing High-Speed Data Streams (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

- [GM04] J. Gehrke, S. Madden. Query Processing in Sensor Networks. *IEEE Pervasive Computing* 3(1):46–55, 2004.
- [GO03] L. Golab, M. T. Oezsu. Issues in data stream management. *SIGMOD Rec.* 32(2):5–14, 2003.
- [HHH<sup>+</sup>02] M. Harren, J. M. Hellerstein, R. Huebsch, B. T. Loo, S. Shenker, I. Stoica. Complex Queries in DHT-based Peer-to-Peer Networks. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Pp. 242–259. Springer-Verlag, London, UK, 2002.
- [IEC04] IEC/ISO. *IEC61850 Part 7-3: Basic Communication Structure for Substation and Feeder Equipment - Common Data Classes*. IEC, Geneva, Switzerland, 2004.
- [KBB07] D. Kucuk, B. Boyrazoglu, S. Buhan. PQStream: A Data Stream Architecture for Electrical Power Quality. In *International Workshop on Knowledge Discovery from Ubiquitous Data Streams*. 2007.
- [KG02] S. S. Krishna P. Gummadi, S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. *Proceedings of SIGCOMM IMW 2002*, 2002.
- [OV99] M. T. Oezsu, P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1999.
- [RD01] A. Rowstron, P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. *Lecture Notes in Computer Science* 2218:329–350, 2001.
- [SAL<sup>+</sup>96] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, A. Yu. Mariposa: A Wide-Area Distributed Database System. *VLDB Journal: Very Large Data Bases* 5(1):48–63, 1996.
- [SGM08] F. Staerber, C. Gerdes, J. Mueller. A Peer-To-Peer-Based Service Infrastructure for Distributed Power Generation. In *Proc. of 17th IFAC World Congress, Seoul, Korea, Intl. Federation of Automatic Control*. 2008.
- [SMK<sup>+</sup>01] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. Pp. 149–160. ACM, New York, NY, USA, 2001.