Increasing the Resilience of Critical SCADA Systems Using Peer-to-Peer Overlays

Daniel Germanus, Abdelmajid Khelil, and Neeraj Suri*

DEEDS Group, Computer Science Department, TU Darmstadt, Germany
{germanus,khelil,suri}@cs.tu-darmstadt.de
http://www.deeds.informatik.tu-darmstadt.de

Abstract. Supervisory Control and Data Acquisition (SCADA) systems are migrating from isolated to highly-interconnected large scale architectures. In addition, these systems are increasingly composed of standard Internet technologies and use public networks. Hence, while the SCADA functionality has increased, its vulnerability to cyber threats has also risen. These threats often lead to reduced system availability or compromised data integrity, eventually resulting in risks to public safety. Therefore, enhancing the reliability and security of system operation is an urgent need. Peer-to-Peer (P2P) techniques allow the design of self-organizing Internet-scale communication overlay networks. Two inherent resilience mechanisms of P2P networks are path redundancy and data replication. This paper shows how SCADA system's resilience can be improved by using P2P technologies. In particular, the two previously mentioned resilience mechanisms allow circumventing crashed nodes and detecting manipulated control data.

1 Introduction & Contributions

Supervisory Control and Data Acquisition (SCADA) systems form key components for Critical Infrastructure (CI) trustworthy monitoring and control. The ubiquitous communication developments are also leading to highly interconnected CIs, resulting in large scale and heterogeneous SCADA networks [23]. While the Internet scale ranges to 10^9 nodes, the US powergrid is currently comprised of 10^5 nodes and steadily increasing [10]. The transition from local area to wide area SCADA systems also corresponds to an increasing replacement of proprietary and vendor-specific communication protocols by open standards [17] and Commercial-Off-The-Shelf (COTS) protocols mainly based on Internet hardware and the Internet Protocol (IP) suite. Overall, the wide area SCADA systems entail immense heterogeneity in terms of network technologies and node properties. Heterogeneity mainly manifests in the interconnection of legacy and state-of-the-art devices, both varying in terms of their computational capacities.

* Research supported in part by EU INSPIRE, CASED (www.cased.de), and EU CoMiFin.

The growing usage of low cost COTS components comes at the cost of potentially increasing the vulnerability of SCADA systems to node and communication failures and cyber attacks. The crash of SCADA network nodes usually implies the disturbance of SCADA message flows and consequently may perturb the required trustworthy monitoring and control of the corresponding CI and physical processes. More and more SCADA systems are interconnected through public networks and mainly the Internet. Public networks expose them to cyber threats [7]. Being exposed to cyber threats through the Internet eventually results in data integrity attacks, i.e., the deliberate injection of incorrect data to the SCADA system which may have fatal consequences on the proper operation of CIs. Such SCADA perturbations may result in severe CI failures at the basic level of service disruption to critical impact on public/CI safety. Consequently, SCADA resilience against failures and attacks is essentially needed.

Paper Contributions: To achieve increased SCADA system resilience against cyber threats in large-scale systems, our paper proposes a minimally intrusive and low cost communication overlay onto legacy SCADA systems using Peer-to-Peer (P2P) technologies [3]. In particular, we show that our approach efficiently (i) prevents data loss due to node crashes, and (ii) detects and remedies data integrity attacks. Path redundancy and data replication are two P2P mechanisms that we rely on for this purpose. Path redundancy refers to multiple paths between pairs of peers; data replication implies distributed and redundant data storage across the network.

We propose a middleware-based approach that requires only minimal changes to the existing SCADA software system. Furthermore, our solution is scalable to accommodate ongoing developments of interconnected SCADA systems. In addition, our P2P-based data integrity approach enables avoiding the usage of public key infrastructures. This enables the integration of SCADA nodes with low computational capacities, as the timeliness overhead introduced by cryptographic operations may violate SCADA timeliness requirements. Our simulation results show that the SCADA system remains stable in the presence of the considered perturbations. Our approach is currently evaluated in the INSPIRE EU research project [24] by using both, simulation and testbed experiments.

The paper is structured as follows. In Section 2, the system, data, fault, and attack models as well as the requirements on SCADA protection are introduced. Section 3 presents our solution. The simulation environment and evaluation results are described in Section 4. The related work is presented in Section 5 and the conclusion as well as future work are provided in Section 6.

2 Preliminaries

We now present our system model which describes a large-scale SCADA system along with the data model and a fault/attack model. The system model gives an overview of the node types in the system and how they interact with each other. The data model specifies the different data flows and control loops within the SCADA system. Furthermore, the attack and fault model describes faults

and attacks that are addressed by our solution to enhance the overall SCADA system resilience. Finally, requirements for SCADA systems are presented to provide mitigation for the considered faults and attack classes.

2.1 System Model

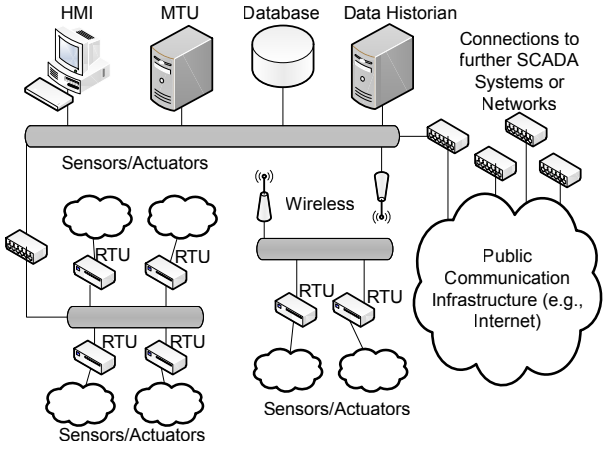

Fig. 1. Basic Components of Interconnected SCADA Systems.

The SCADA system is a network connecting a large number of sensor/actuator clusters with a small set of central control rooms. A generalized SCADA system topology is illustrated in Figure 1. Following the trend to interconnect CIs, the corresponding autonomous SCADA systems are increasingly interconnected in order to facilitate sharing among the different involved authorities, operators etc. Here, autonomy describes a self-contained operator domain, which is not dependent on data exchange with other operator domains to guarantee proper operation. Figure 1 highlights the interconnection of SCADA systems while showing one representative autonomous SCADA system in detail. In the following paragraphs we describe the common components of SCADA systems.

Sensors & Actuators are the monitoring/response components of the SCADA system. Sensors report measurements such as pressure or temperature. Actuators are the controlling elements and conduct operations such as opening or closing valves. Sensors and actuators offer very limited computational and storage capacities.

Remote Terminal Units (RTU) are located "in the field", i.e., they are scattered along the CI. Both, sensors and actuators are attached to a designated

remote terminal unit (RTU), either wired (e.g., through serial interface) or wireless (e.g., through ZigBee [30]). RTUs communicate with superordinate stations using IP encapsulated SCADA protocols.

Master Terminal Units (MTU) collect sensor data from several RTUs and provides the data to other high-level stations (e.g., human machine interfaces (HMI) to give human users a system overview). MTUs also send actuator commands to the appropriate RTU which executes them.

Other stations may reside in a SCADA system as well. The two most prominent ones being a data historian, which preserves relevant information over long durations. The second involves Human Machine Interfaces (HMI) and represents stations that provide human SCADA operators insights and controlling capabilities of the SCADA system.

2.2 Data Model

The communication between SCADA nodes is usually message-based. In each autonomous SCADA system, two message flow directions exist: (i) Upward messages are sent on behalf of RTUs and contain raw or aggregated sensor values, and (ii) downward messages are sent by high-level stations through an MTU to specific RTUs. Examples for downward messages are status requests (e.g., to retrieve a specific sensor value) or actuator commands. Also, different autonomous SCADA systems exchange messages. The message types to be exchanged are individually defined, depending on the participants of an interconnection. Especially, data confidentiality and privacy are considered to address legal obligations, e.g., due to business or political reasons. Therefore, we consider only a subset of data is exchanged that is necessary to support the operation of a distant CI.

Two different control loop classes exist: (i) Safety-critical and (ii) operation-critical. Safety-critical control loops are found between each RTU and its sensors or actuators. Immediate reaction in milisecond ranges is required here to keep the surveilled CI processes stable. On the other hand, operation-critical control loops exist between all SCADA nodes and also across autonomous SCADA system boundaries. Operation-critical control loops have weaker timeliness requirements than safety-critical control loops.

2.3 Fault and Attack Model

In this paper we focus on two fundamental fault and attack classes: Node crashes and data integrity attacks.

Data flow interruptions are usually consequences of node crashes and may conceal crucial monitoring information to decision making SCADA stations. Therefore, node crashes endanger each CIs correct operation. We consider node crashes between source and destination nodes, i.e., the source node is still able to send data, but the transmission path to the destination is disturbed by the crashed node and therefore data gets lost.

Data corruption may be a result of illegitimate data modification on behalf of an attacker. Furthermore, we assume that attackers are not omnipresent and take over a small fraction of peers. The target of data corruption attacks may be any RTU or router in the SCADA system which we assume to be IP based. We do not consider attacks directed against sensors, actuators, or high-level stations. Consequent on a data integrity attack is the provision of incorrect data to the SCADA system which results in an inconsistent system state. The introduced fault and attack classes endanger both safety-critical and operational-critical control loops.

2.4 Design Requirements

In the following, we present the main design requirements demanded to provide protection of large-scale SCADA systems against the discussed faults and attacks. Our objective is to design an add-on protection layer for legacy and evolvable SCADA systems. The protection strategy should besides supervising the SCADA system also provide for immediate and active reaction to reach graceful degradation and a timely recovery of the system in case of faults/attacks.

Flexibility: The solution should be capable to withstand temporary network disconnection or churn effects like frequent entering and leaving nodes. These effects can happen in large interconnected topologies due to node or link failures.

Interoperability: For the multitude of different node types present in a large-scale SCADA system, it is beneficial to mask the nodes heterogeneity. This lowers the customization efforts and eases the solution's deployment.

Minimal Intrusiveness: The desired solution should be minimally intrusive, so existing SCADA applications can be integrated with minimal efforts.

The requirements above show the need for a software layer that mediates between the SCADA applications and the underlying network. This layer should ensure the continuous operation of the applications across the heterogeneous SCADA platforms with their varied network perturbations. This protection software layer is commonly realized by a middleware approach. A middleware should intercept, process and forward SCADA messages, e.g., between the application layer and the network transport layer. In addition, this middleware should fulfil the following crucial requirements.

Protection Enhancement: The new middleware should not introduce new threats to the system, but support it to increase the system's overall resilience against the presented threats.

Resource Frugality: SCADA systems exist for several decades now. Some of these systems employ devices which cannot be regarded as state-of-the-art in terms of computational or storage capacities. Therefore, a solution should not be too resource-intensive in order to meet SCADA's timeliness criteria.

Scalability: Due to many interconnected SCADA networks, a large number of nodes shall cooperate efficiently. Thus, the solution needs to be scalable.

3 P2P-based Middleware for SCADA Protection

In the following subsection, several middleware approaches will be discussed regarding their suitability in a SCADA context. Subsequently, our approach and its concordance to the requirements will be presented.

3.1 Middleware Approach Selection

There are a variety of approaches to build a middleware.

Publish/Subscribe (pub/sub) [13, 4] is an approach that mediates between event/data producers (publishers) and its consumers (subscribers). Pub/sub systems are usually maintained by so called brokers who decide on the dissemination paths. Broker systems represent a weak point of the overall system, since in case they become unavailable for some reason (e.g., an attack), the pub/sub system is rendered ineffective.

Transactional middlewares [9, 2] provide strong data consistency and address reliable data dissemination and persistent storage. Yet, this class of systems requires a noteable amount of system resources and thereby contradicts the requirement of small overhead consumption.

Web Services [19, 15] provide a state-of-the-art communication and data dissemination paradigm. While web services offer high interoperability, the scalability and protection enhancement requirements are violated. Like for the pub/sub approach, web services require central authorities that provide discovery services.

Mobile Agent Systems [21, 27, 15] are autonomous and decentralized systems used for self-organizational tasks. Their primary target is neither data dissemination nor replication, but they may be employed for network or node reorganization during perturbations. Mobile agent systems meet the scalability and flexibility requirements. However, the less provable dependability and security of mobile agent systems made these systems less accepted.

We propose the usage of P2P technology as a solution, because P2P meets all previously introduced requirements. A detailed comparison of our approach with the given requirements is presented in Section TODO 3.3. P2P architectures outperform pub/sub systems in terms of the scalability and protection enhancement requirements. Since the broker system functionality is not distributed among all participating nodes. P2P-based middleware outperforms the transactional and web services middlware approaches as these are "heavy" systems with high computational requirements to reach high data consistency. Mobile agents require complex management and fault-tolerance mechanisms which are easily overcome by P2P.

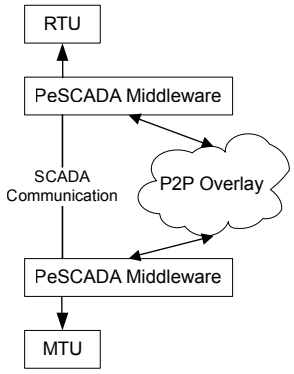

Fig. 2. Integration of PeSCADA into existing SCADA Systems

3.2 PeSCADA Architecture

The network architecture of PeSCADA involves RTUs, MTUs, and high level stations (cf. Figure 1). To increase system resilience, PeSCADA is integrated like a middleware into each of the previously mentioned nodes (cf. Figure 2). It resides between the SCADA application and the IP layer, listens to and extracts SCADA messages of the original SCADA application and finally stores them in the P2P network (cf. Figure 3). The listener component is hooked into the SCADA application communication. A SCADA model describes message formats and relevant payload is extracted from the original messages. Consequently, the extracted payload is forwarded to the local P2P client which stores the data in the P2P overlay.

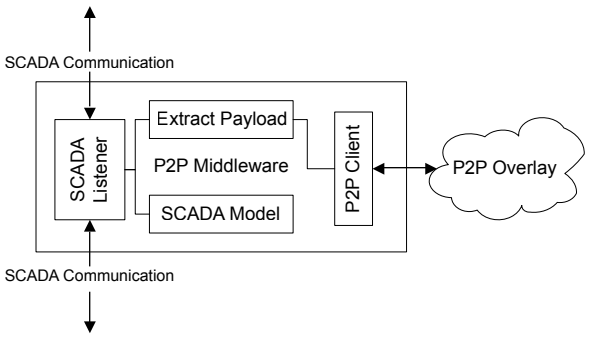

Fig. 3. PeSCADA Middleware Building Blocks

Data storage within PeSCADA is realized with a DHT, which is a distributed associative array that stores key/value pairs and assigns, according to the key's

value, maintenance responsibilities to overlay nodes. The architecture of a DHT is flat, while interconnected SCADA systems are hierarchically organized. To address this architectural difference, each autonomous SCADA system introduces its own *local* overlay network. Local overlays promote legal and performance aspects in interconnected large scale topologies, e.g., data that may reflect corporate secrets is stored locally in the domain of its originating operator. Also, lookup and data retrieval latencies are improved in case overlays are limited to the network of each autonomous SCADA system. Besides local overlay networks all autonomous SCADA systems are part of a *global* overlay network. Data to be shared among different autonomous SCADA systems is specified in filter lists and stored in the global overlay. The notion of local and global overlays is also depicted in Figure 4.

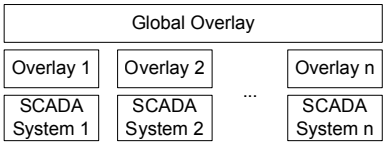

Fig. 4. Arrangement of Local and Global Overlay Networks

Thus, the functionality of the PeSCADA middleware layer is threefold: (i) Overlay management, (ii) data management, and (iii) SCADA communication interception. The overlay management includes the functionality that is required to maintain a P2P network, in the PeSCADA case either a Chord [26] or Kademlia [18] protocol implementation.

3.3 PeSCADA's Concordance with Requirements

PeSCADA is based upon structured P2P networks [18, 26], since they provide good scalability properties and require $O(log\ n)$ routing steps in a network with n peers, whereas unstructured networks require up to $O(n)$ steps.

Structured P2P networks are also capable to handle churn, i.e., frequent entering and departing peers in notable amounts. Therefore, these overlays provide flexibility according to our requirements.

Furthermore, P2P technology meets our interoperability requirement, by providing an identical interface for all peers and thereby masking heterogeneity.

The P2P functionality is inserted into existing SCADA applications as a middleware. This forms a logical layer between the operating system and the SCADA application to intercept, process, and forward SCADA messages. The existing SCADA application needs minimal changes, eventually only a reconfiguration and no source code changes is needed.

Integrating new technology into an existing system bears the risk of introducing new threats. Different P2P-related attacks, like for instance the Sybil

attack [12], require an adversary to newly introduce a notable amount of peers to infiltrate the network. Since SCADA systems are no open networks like for example P2P filesharing networks, attackers are not able to arbitrarily introduce new peers. Furthermore, a secure admission protocol [25] could be applied to disallow entering of unsolicited peers. The proposed P2P-based approach does not replace the SCADA system functionality, but represents a supplement to increase SCADA system resilience.

While the previously mentioned attack class targets the P2P routing mechanism, other attacks against structured P2P networks exist [28, 8], although they also require malicious peers colluding on behalf of an adversary. It is part of the future work to address these attacks in detail. Due to the natural physical distribution, it is unlikely that an attacker is able to hijack all peers that hold replicas of a specific datum at the same time; this property of P2P networks provides protocol inherent resilience.

The overhead requirement is met as well, since no computationally challenging operations are executed. This requirement was the main driver to decide against the application of cryptographic methods, since legacy devices cannot achieve both, cryptographic computations and SCADA timeliness requirements.

3.4 P2P-Inherent Resilience Mechanisms

Path redundancy presents a simple robustness concept in P2P networks: Requests can be sent along different paths to both speed up data retrieval and offer increased resilience against node or link failures. The number of redundant paths to be chosen can be configured. Clearly, using more redundant paths implies lower latencies for data retrieval but generates more network traffic in the system.

We evaluated [16] the choice of suitable P2P technology for SCADA systems. Regarding the requirements of interconnected large-scale SCADA systems, structured P2P networks with Distributed Hash Tables (DHT) are appropriate for three reasons, i.e., (i) low routing latencies, (ii) good scalability, and (iii) data discovery guarantees in DHTs [16].

Data replication in DHTs increases the availability of data throughout the network. DHT entries are stored at k different peers, usually $k = 3$. Large values of k result in increased availability and fault tolerance of the system. The downside is reduced system performance due to increasing network traffic. If a peer p that provides a datum d leaves the network, d is still available at $k - 1$ other peers. Subsequent to peer p's departure, P2P self-organization mechanisms adapt the P2P network's routing tables and choose another peer to store the replicas formerly stored on p.

3.5 PeSCADA Strategies to Increase SCADA System Resilience

In this subsection, we present PeSCADA's anticipation approach for the considered perturbations described in Section 2.3, namely node crashes and data integrity attacks.

Protecting SCADA from Node Crashes: The PeSCADA middleware tracks the reception of sensor messages in MTUs by using expectancy timers to suspect message loss. In case a message becomes overdue, PeSCADA requests the specific sensor message from the P2P overlay network. This mechanism bridges the time between a node crash and its recovery (e.g., by reboot) or until the routing tables are updated through the distributed routing algorithm (e.g., OSPF [22]). This helps to deliver data to the SCADA application during perturbations, i.e., PeSCADA acts as a surrogate data delivery mechanism.

Protecting SCADA from Data Integrity Attacks: PeSCADA is able to discover data corruption attacks, if the location of corruption is between source and destination. We consider corruptions that occur after initial message replication in the overlay, i.e., the corruption occurs on a compromised router. PeSCADA operates as follows: Whenever a SCADA message arrives at an MTU through the conventional SCADA communication channel, the MTU requests the same message via the P2P overlay from q different replica locations and compares it to the initially received message which is accepted if they are identical. The parameter q needs to be less or equal to k which is the system wide replication degree. Choosing large q means that attackers are required to hijack more nodes to successfully spoof the system. On the other hand, choosing small q results in better performance as less messages will be sent through the network.

The DHT data storage does not take the duties of a distributed SCADA data historian. Since autonomous SCADA systems contain up to 10^6 datapoints that send data in second to deca-second intervals, long term storage in devices with varying and potentially very limited resource capacities cannot be realized. Regarding the two previously introduced strategies, the DHT serves for effects like node crashes or to detect ongoing data integrity attacks. These effects require countermeasures near in time, therefore, short term data storage is in favor. Short term storage also decreases the P2P network load because no republishing of DHT entries is performed. Republishing is required to counteract churn effects in P2P networks and thereby to guarantee long-term data availability. [18] proposes to republish data every 24 hours, PeSCADA does not republish data at all. Furthermore, as an extension to [18] we implemented a time to live (TTL) management for DHT entries.

4 Performance Evaluation

This Section first describes the simulation environment. Next, PeSCADA is evaluated for the two considered perturbation scenarios, i.e., data loss mitigation and detection of data integrity attacks.

4.1 Simulation Environment and Settings

We follow a simulation-based evaluation, since our full-scale system model involving many protocol layers and a large set of parameters is unfavorable to break down into an analytical model.

The simulation is implemented using OMNet++ [20], a discrete event simulator. It provides core concepts like message queues, message passing between objects, and an interpreter programming language to define nodes and networks of nodes. Other simulators exist, but the availability of extensions like INET [1] and OverSim [5] revealed OMNet++'s adequacy for PeSCADA. INET provides an implementation of the IP suite to model and simulate large scale SCADA scenarios. Due to lack of open source SCADA scenario generators that could be coupled with OMNeT++, we created our own SCADA scenario generator for OMNeT++, which we made available for the community. Accordingly, we gather simulation results involving the following INET protocol implementations: ARP, IP, TCP, UDP, and OSPF [22].

OverSim builds upon INET and provides the implementation of different P2P overlay networks. We performed simulations using OverSim's Chord [26] and Kademlia [18] protocol implementations. Both protocols provide a DHT as application layer on top of the P2P network. SCADA sensor data is replicated within a local DHT to provide it to an MTU in case the regular SCADA application communication is perturbed. The simulation results provided later in this section show the performance of a local DHT which consists of 8 through 512 RTUs acting as peers and 64 sensors per RTU. At startup each sensor chooses a random but fixed sampling period in the range of 1 to 30 sec. The simulation terminates after 600 sec. Faults and attacks are initiated at $t = 100s$ to provide simulated nodes sufficient time for self-configuration tasks and the P2P network setup. The system wide packet drop rate is set to 10^{-3} to model an unreliable underlay network. The TTL for DHT entries is set to 300 seconds.

The following key is an example for the addressing scheme of DHT key/value pairs: $RAW\_042\_017\_20100102122124$. The key represents raw sensor data of RTU 42's sensor number 17, processed and replicated by the RTU on January, 2nd 2010 at 12:21:24. Clearly, key calculation for a sequence of keys is trivial in case of static sensor intervals. In case that the sequence calculation is not trivial, other mechanisms exist which are omitted here.

4.2 Case Studies

Data Loss Mitigation: A router is set to a dead state. Consequently, the OSPF [22] protocol detects this unresponsive router and initiates a route repair. The time span between router crash and the completion of the reconfiguration process is bypassed via P2P, because packets routed across the dead router get lost. MTUs run expectancy timers for sensor messages, and in case a message is indicated to be overdue, the MTU requests this missing message via the P2P network. Simulations use a replication degree of $k = 2$, i.e., two copies of each DHT key/value pair exist in the network to provide basic redundancy. Therefore, we simulate a SCADA network with a mesh topology, such that alternative routing paths may be taken up to a certain amount of node crashes. In our simulation, MTUs run expectancy timers for sensor messages, and in case a message is indicated to be overdue, the MTU requests after a short waiting period the missing data via the P2P network.

Detection of Data Integrity Attacks: An arbitrary RTU is set to a malicious state leading to sensor messages in transfer being corrupted. For each received SCADA sensor message on an MTU, the same datum is requested from q different replica locations. With the multiple received messages, the receiver can check for their SCADA payload equality. Simulations use a replication degree of $k = 3$, i.e., three copies of each DHT key/value pair exist in the network.

4.3 Metrics

To quantify the benefit of our approach, we define several metrics. The evaluation criteria can be split up according to three dimensions, namely *reliability/security*, *timeliness*, and *overhead*.

Reliability is measured as a percentage of received messages, that would be lost without the P2P mechanism. The metric's formula is:

$$Reliability = \frac{\#Receipts}{\#Requests}$$

$\#Receipts$ is the number of messages received via DHT and $\#Requests$ is the total number of missing messages that have been requested.

Security is evaluated as percentage of discovered corrupted messages. The metric's formula is:

$$Security = \frac{\#Identified}{\#TotalInjected}$$

$\#Identified$ is the number of identified corrupted messages, $\#TotalInjected$ is the total number of corrupted messages that have been injected. Both metrics value ranges are $[0, 1]$, where 1 indicates 100% reception or discovery, depending on the respective scenario.

Timeliness is examined in terms of latency from a request until its completion.

Overhead is evaluated in terms of the number of messages sent/received per peer, and the corresponding incoming and outgoing network traffic per peer. P2P traffic is split up into three subcategories: (i) Application (DHT), (ii) peer discovery (lookup), and (iii) overlay maintenance.

4.4 Simulation Results

In terms of timeliness evaluation, PeSCADA recovers lost messages within the range of 3 to 7 sec using Kademlia [18] and 2 to 5 sec using Chord [26].

Recovery rates for the router crash scenario are given in Figure 5 in terms of the reliability metric. Kademlia provides success rates above 75% with 16 peers or more. Chord ranges between 65% and 95% with 16 peers or more. Recovery failures occur due to two different causes: (i) The P2P network communication

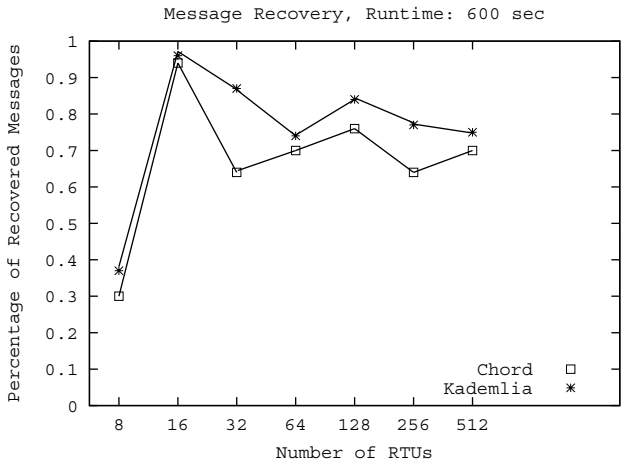

Fig. 5. Success Rate of Lost Message Recovery (Chord & Kademlia)

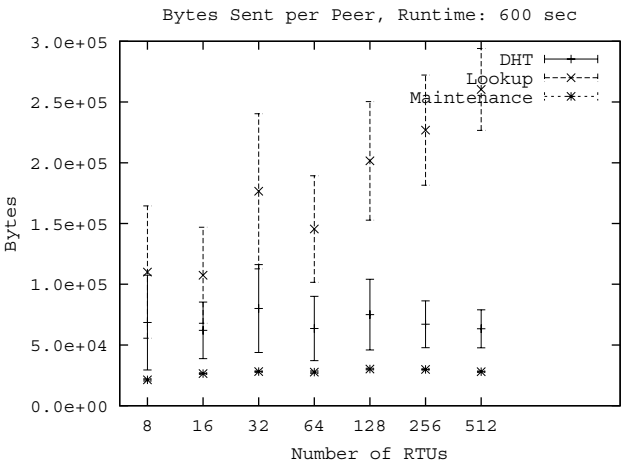

Fig. 6. Sent Messages (Chord)

is partially disturbed due to the router crash and therefore unable to satisfy all requests or (ii) P2P messages get lost due to the packet error rate. Fluctuations in the success rates occur due to PeSCADA's replication and routing scheme: In case the RTUs that are affected by the router crash are responsible for the specific address space range of the requested datum, the MTU cannot retrieve the data. Our simulations show that Kademlia has increased robustness compared to Chord. However, Kademlia communication overhead exceeds Chord. Figure 8 shows the average amount and the 95% confidence interval of Kademlia mes-

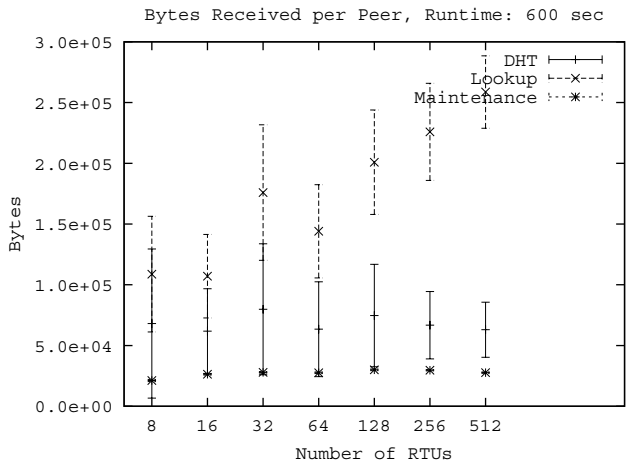

Fig. 7. Received Messages (Chord)

sages sent per peer. Figure 9 shows received messages for Kademlia. Compared to Kademlia, Chord requires an order of magnitude less data bytes for DHT messages (for both sending and receiving), as can be seen for the corresponding RTU numbers across Figures 6 and 7 as well as Figures 8 and 9.

For the data integrity attack scenario, PeSCADA is able to discover 90% of the faked messages by requesting each datum from $q = 3$ different peers for comparison. If the SCADA message corresponds to the message received from at least two of three peers, it is regarded as valid. In other cases, the SCADA message is discarded and counted as corrupted message. Consequently, a message copy that is retrieved from the DHT is used. Traffic consumption in this scenario is clearly higher than in the previous scenario, since each received message is requested threefold.

5 Related Work

Several related approaches to increase SCADA system resilience exist in the literature. The approaches presented in this Section differ in terms of the employed technology (P2P based, pub/sub based, MPLS), requirements (data provision, group communication, fast delivery), and communication paradigm (many-to-one, one-to-many, many-to-many).

Some of the previous works [6, 11, 29] propose a non-intrusive communication infrastructure that is built from scratch. Our approach is minimally intrusive as we build the protection layer on existing already deployed SCADA systems while keeping the change of the existing SCADA system minimal.

P2P for protecting CIs has been proposed before. In [6], the authors qualitatively discuss the advantages and disadvantages of using Chord [26] for the

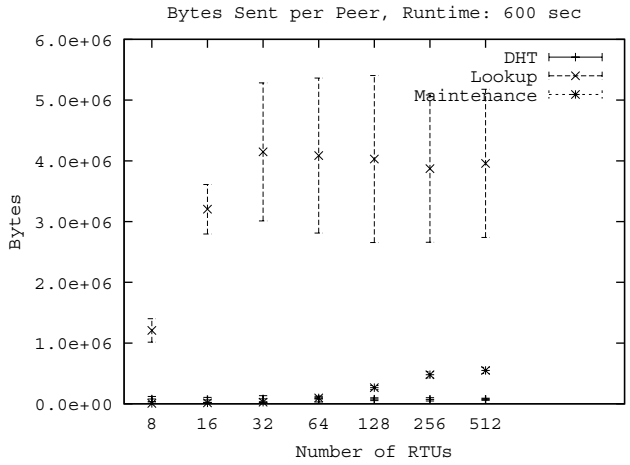

Fig. 8. Sent Messages (Kademlia)

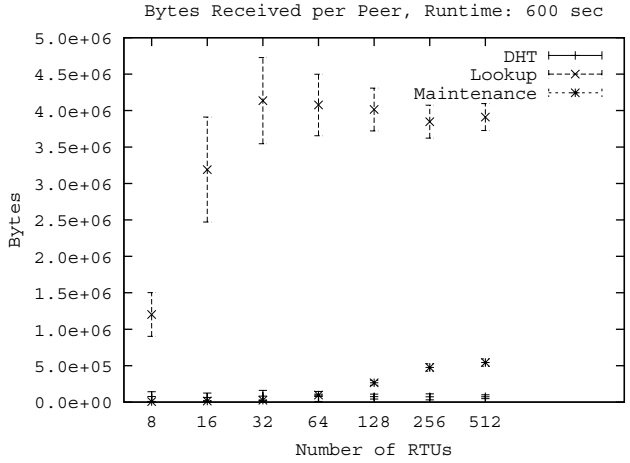

Fig. 9. Received Messages (Kademlia)

surveillance of power grids. The main contribution is an agent-based layer on top of a P2P network to improve message exchange reliability. In [11] the authors propose a hybrid unstructured overlay network without central indexing services to increase power grid surveillance and monitoring resilience. Although the higher timeliness requirements in comparison to structured networks are mentioned, no evaluation is provided to compare both overlay classes.

In contrast to our approach, pub/sub models [14, 13, 4] require message broker nodes which conduct the message reception and forwarding mechanism. The notion of message brokers represent like super peers in hybrid P2P systems potential weak points in the system.

[14] proposes a pub/sub middleware to meet the strong timeliness requirements for data delivery within power grid SCADA systems. It aims at satisfying the needs of the electric power system, i.e., low latency and reliable delivery of data produced anywhere in the network and to multiple interested sites. The middleware is based upon a pub/sub communication model, which provides data item transport from a source (the publisher) to various sinks (subscribers) without requiring the publisher to track its subscribers. Experiments show, that the performance in terms of forwarding latency and reliability is sufficient according to power grid requirements.

[29] focuses on power grids and proposes an information architecture aiming at increased reliability. The architecture is twofold, addressing operational and planning aspects, where the first has stronger timeliness requirements than the second. To meet security and reliability requirements, a potpourri of technologies is applied: Multi-protocol label switching (MPLS), Virtual Private Networks (VPN), and firewalls. Finally, different redundancy topologies are introduced with different scalability properties. [29] provides a solution that requires fundamental changes to the SCADA systems in terms of both, software and hardware.

6 Conclusion & Future Work

We presented PeSCADA, a middleware solution to increase SCADA system resilience in the presence of faults or attacks using a P2P based approach. Our approach consists of building a self-organized structured P2P overlay on top of the SCADA network and in exploiting the inherent path redundancy and data replication to enhance the resilience of the SCADA system. We considered two highly probable fault/attack case studies, namely node crashes and data integrity attacks. In the first case, our solution detects delayed/lost messages through message expectancy timers and requests copies from the overlay network. In the second case, our solution requests for each regularly received SCADA message the same datum from the overlay for comparison, in order to detect data manipulations without any use of cryptographic methods.

As ongoing work, we are customizing PeSCADA to the specific needs of power grids. Furthermore, we are considering the mitigation of further perturbations of SCADA systems through P2P technology.

References

1. INET Framework. http://inet.omnetpp.org
2. PostgreSQL. http://www.postgresql.org/
3. Androutsellis-Theotokis, S., Spinellis, D.: A Survey of Peer-to-Peer Content Distribution Technologies. ACM Comput. Surv. 36(4), 335–371 (2004)
4. Banavar, G., Chandra, T., Mukherjee, B., Nagarajarao, J., Strom, R.E., Sturman, D.C.: An efficient multicast protocol for content-based publish-subscribe systems. In: ICDCS '99: Proceedings of the 19th IEEE International Conference on Distributed Computing Systems. p. 262. IEEE Computer Society, Washington, DC, USA (1999)
5. Baumgart, I., Heep, B., Krause, S.: OverSim: A Flexible Overlay Network Simulation Framework. In: Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007. pp. 79–84 (2007)
6. Beitollahi, H., Deconinck, G.: Analyzing the Chord Peer-to-Peer Network for Power Grid Applications. In: Fourth IEEE Young Researchers Symposium in Electrical Power Engineering. p. 5 (2008)
7. Bowen, C.L., I., Buennemeyer, T., Thomas, R.: Next generation SCADA Security: Best Practices and Client Puzzles. In: Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC. pp. 426–427 (June 2005)
8. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. SIGOPS Oper. Syst. Rev. 36(SI), 299–314 (2002)
9. Codd, E.F.: The relational model for database management: version 2. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1990)
10. Dave Bakken: Smart Grid Data Delivery Service. http://ec.europa.eu/research/conferences/2009/ict-energy/pdf/dave_bakken_en.pdf
11. Deconinck, G., Rigole, T., Beitollahi, H., Duan, R., Nauwelaers, B., Van Lil, E., Driesen, J., Belmans, R., Dondossola, G.: Robust overlay networks for microgrid control systems. In: DSN2007, 37^{th} Annual IEEE/IFIP International Conference on Dependable Systems and Networks. p. 6. Edinburgh, U.K. (June 25-28 2007)
12. Dinger, J., Hartenstein, H.: Defending the sybil attack in p2p networks: taxonomy, challenges, and a proposal for self-registration. In: Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. pp. 8 pp.– (April 2006)
13. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Comput. Surv. 35(2), 114–131 (2003)
14. H. Gjermundrod et al: GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid. Power Delivery, IEEE Transactions on 24(1), 136–143 (2009)
15. Ketel, M.: A mobile agent based framework for web services. In: ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference. pp. 1–6. ACM, New York, NY, USA (2009)
16. Khelil, A., Jeckel, S., Germanus, D., Suri, N.: Benchmarking of P2P Technologies from a SCADA Systems Protection Perspective. In: MOBILIGHT 2010: Inproceedings of the 2nd International Conference on Mobile Lightweight Wireless Systems. to appear (2010)
17. Krutz, R.L.: Securing SCADA Systems. Hungry Minds Inc (2005)

18. Maymounkov, P., Mazières, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems. pp. 53–65. Springer-Verlag, London, UK (2002)
19. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 16(3), 389–415 (2007)
20. Pongor, G.: OMNeT: Objective Modular Network Testbed. In: MASCOTS '93: Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems. pp. 323–326. The Society for Computer Simulation, International, San Diego, CA, USA (1993)
21. Pridgen, A., Julien, C.: A secure modular mobile agent system. In: SELMAS '06: Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems. pp. 67–74. ACM, New York, NY, USA (2006)
22. RFC Standards Track: RFC 2328, OSPF Version 2
23. Rinaldi, S., Peerenboom, J., Kelly, T.: Identifying, understanding, and analyzing Critical Infrastructure Interdependencies. Control Systems Magazine, IEEE 21(6), 11–25 (Dec 2001)
24. Salvatore D'Antonio, Abdelmajid Khelil, L.R.N.S.: INcreasing Security and Protection through Infrastructure REsilience: the INSPIRE Project. In: Proc. of The 3rd International Workshop on Critical Information Infrastructures Security (CRITIS) (October 2008)
25. Sandhu, R., Zhang, X.: Peer-to-peer access control architecture using trusted computing technology. In: SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies. pp. 147–158. ACM, New York, NY, USA (2005)
26. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: 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 (2001)
27. Suri, N., Bradshaw, J.M., Breedy, M.R., Groth, P.T., Hill, G.A., Jeffers, R., Mitrovich, T.S., Pouliot, B.R., Smith, D.S.: Nomads: toward a strong and safe mobile agent system. In: AGENTS '00: Proceedings of the fourth international conference on Autonomous agents. pp. 163–164. ACM, New York, NY, USA (2000)
28. Urdaneta, G., Pierre, G., van Steen, M.: A survey of DHT security techniques. ACM Computing Surveys `http://www.globule.org/publi/SDST_acmcs2009.html`, to appear
29. Z. Xie et al: An information architecture for future power systems and its reliability analysis. Power Engineering Review, IEEE 22(6), 60–60 (June 2002)
30. ZigBee Alliance: http://www.zigbee.org, `http://www.zigbee.org`