Reliable and Scalable Communication for the Power Grid *

Christopher Zimmer, Frank Mueller

Abstract Future smart power grids require constant data availability for actuation of control decisions. The job of ensuring the timely arrival of data falls onto the network that connects these intelligent devices. This network needs to be fault tolerant. When nodes, devices or communication links fail along a default route of a message from A to B, the underlying hardware and software layers should ensure that this message will actually be delivered as long as alternative routes exist. Existence and discovery of multi-route pathways is essential in ensuring delivery of critical data.

In this work, we present methods of developing network topologies of smart devices that enable multi-route discovery in an intelligent power grid. This is accomplished through the utilization of software overlays that (1) maintain a digital structure for the physical network and (2) identify new routes in the case of faults. The resulting cyber network structure is scalable, reliable and inexpensive to build by extending existing infrastructure.

1 Introduction

Today's critical infrastructure often governs control decisions of intelligent devices that can have a significant impact on human life, the environment and the economy. Ensuring that the appropriate data is available is crucial for making informed decisions. Such considerations are becoming increasingly important in cyber-physical systems (CPS) that combine computational decision making on the cyber side with physical control on the device side, let it be the power grid, medical devices or automotive subsystems.

Christopher Zimmer, Frank Mueller
North Carolina State University, Raleigh, NC 27695-8206, mueller@cs.ncsu.edu

* This work was supported in part by NSF grants 1329780, 1239246, 0812121 and U.S. Army Research Office (ARO) grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI). This is an extended version of a prior conference paper [22].

Conventional embedded systems and a CPS differ in that the later governs physical devices through embedded control in a *networked environment* and has a direct impact on people who rely on such devices. Failure of network equipment in a CPS environment may have a number of impacts, such as

- faulty decisions regarding device malfunctioning,
- incorrect actuation (decisions) due to lack of data,
- system reconfigurations/restart, or
- severe performance degradation with missed deadlines.

In smart (power) grids that rely on commodity communication infrastructure, as one example, these types of failures are expensive and cause inefficiencies. Decisions are being made that affect the real world based on data passed within the network of smart grids. This impact on the real world makes it necessary to improve communication within the smart grid to ensure that the *correct* decisions are made in a *timely* manner .

Assuming correct device behavior, the timeliness requirement falls onto the network that connects these intelligent devices. Failures may occur in today's CPS because of a lack of flexibility in routing decisions. Routing decisions are an important part of networking. Commodity networking equipment often relies on static routing techniques within networks. When there is a failure along a static route, any messages sent along that route will time out and result in communication failure. In these scenarios, many systems will assume points along this route to be out of service. This does not have to be the case.

Networks can be designed to be fault tolerant. When nodes, devices or communication links fail along a default route of a message from A to B, the underlying hardware and software layers should ensure that this message will actually be delivered as long as alternative routes exist. Networks of devices can be configured to contain multiple pathways to connect clusters of nodes in a redundant manner. One can ensure delivery of critical data via different network routes, i.e., multi-route pathways need to exist. A network, upon discovery of a faulty route, then needs to be able to utilize an alternate route.

In conventional networks, the main objective is to maximize throughput. Commodity network equipment is designed to provide high levels of throughput. This design choice runs counter to the needs of an intelligent distributed network required for next-generation CPS infrastructure. For example, in a power grid, the guarantee that a message is delivered is more important than high rates of throughput. Sample tasks the power grid must perform, such as distributed load balancing [4], substantiate this need. Thus, system components need to collaborate intelligently upon a component network failure to accommodate sustained communication needs at all times.

Network failures may conventionally result in message delivery failure. This can be avoided through smart routing technologies that can bypass faulty equipment in modern network topologies. However, such fault tolerance is only feasible in situations where the faulty equipment does not constitute a single point of communication failure. Therefore, it is important to maintain redundant pathways through

networks. Another problem with smart routing technology is that in current topologies routers are sparely distributed as their cost is significantly higher than that of switches. Since switches lack routing capabilities, this severely limits the ability of CPS devices to sustain network failures through automatic re-routing over alternate paths.

Contributions: This work develops novel methods for designing network topologies of smart devices that enable multi-route discovery in an intelligent power grid. This is accomplished through the utilization of software overlays that (1) maintain a digital structure for the physical network and (2) identify new routes in the case of faults. To this end, we first present a method of utilizing software network overlays to enable the discovery of additional communication pathways throughout a network. Using abstracted network information, the system is able to react in case of faults and generate new routes through the network in a manner that is transparent to the user by providing a software overlay middleware. In this network, any single node in the system can act as a message-passing agent to dynamically route messages within the network. This paradigm enables us to use inexpensive network devices abundantly within the network and ensure a resilient communication infrastructure at the same time.

The primary aim of this study is to determine appropriate topologies in which to structures the various devices used in Distributed Grid Intelligence (DGI) to insure that the Intelligent Energy Management (IEM) nodes are able to make optimal decisions. By formally creating a network topology in this system we are better able to guarantee critical services that would be delegated through efficient communication of the IEM nodes.

This work further presents a visualization capability to monitor connection states and pathways through the network aimed at helping external entities to understand the states of the network.

Our vision is that the application of this approach in an intelligent power grid will enable IEM and IFM devices to make automated, decentralized decisions and to maintain state of lower-level devices.

2 Related Work

Wauters et al. [20] survey network overlays for computer networks and assess their suitability for smart grids. They identify reliability and cost metrics for different topologies, which are more costly than our work and deliver comparable reliability.

GridStat [18] enables the allocation of node specific redundant pathways for high-level power-grid networks. Our work is orthogonal to GridStat as it is designed for low-level micro grids with switches. We also enable dynamic arbitration over many redundant pathways, which allows for a generic application of redundancy that is resilient to link faults.

Software overlays have also been utilized in the High Performance Computing (HPC) domain, e.g., by Varma et al. [19]. That work focused on structural compres-

sion and reduction of data over a radix tree irrespective of physical topology while COMIG focuses on making physical tree topologies resilient via crosslinks.

Commodity tools such as Cacti [3] or Ganglia [14] provide graphical monitors of networked systems and components for conventional computing ensembles. Our distributed live monitoring (DLM) work (Section 8) differs in that it is able to provide the visualization for non-IP devices such as those used in a Zig-Bee platform that are only MAC-addressable. Our DLM interfaces with IP-addressable nodes to detect any MAC-based devices connected to it and displays their current status.

Berthier et al. [7] discuss the impact of cyber network topology on state estimation for power grids using contingency analysis based on an ad-hoc exploration of this topology. Yan et al. [21] model the vulnerability of power network topologies to cascading failures from the security angle. Our work focuses on cyber network connectivity to sustain or even prevent power outages while most of these prior works focus on network security.

Motivated by early considerations about the power grid [5], Nguyen et al. [15] and Huang et al. [10] assess the impact of clustering due to partitioning for power grids by developing a 1-to-1 and an k-n model, respectively, (with k control nodes and n power nodes) and empirically simulate the partitioning characteristics of different topologies. Other models include multiple-to-multiple [16] and regular allocation [16] assumptions. Our topology is an example of a regular model, but our work differs in that we consider how to retrofit existing cyber networks to create a more robust topology. We also consider an abstract tree overlay, which may or may not match the physical cyber network topology.

This work extends our prior publication [22] by the following contributions: It contains a more detailed motivation of the problem, additional background information, clarifications of its relation to micro grids, more details of micro grid assumptions, cost considerations in retrofitting power systems with additional cyber network paths, more detailed explanations of our technical approach, a refined lower-cost placement of crosslinks plus an algorithm, more comparisons to related work, and a discussion of future work on cyber security specific to CPS.

3 Micro Grids and Renewables

The power grid is currently undergoing a significant transition from a centralized architecture centered around large capacity generation resources of power plants to a future distributed architecture where large generation sources are complemented by many small ones due to renewable generation sources, such as photovoltaic and wind. To address the transitional challenges of our power infrastructure, the NSF Engineering Research Center (ERC) for Future Renewable Electric Energy Delivery and Management (FREEDM), a multi-institutional project, investigates the cyber-infrastructure of micro grids harboring renewable generation sources [2].

In the FREEDM system, power management of green energy is provided in a highly distributed and scalable manner. The system has to ensure that Intelligent

Energy Management (IEM) and Intelligent Fault Management (IFM) devices have the appropriate data to make control decisions for micro grids and with respect to micro grid connectivity to an upstream utility power grid. This is described in more detail later.

At a grander scale, FREEDM is contributing to methods to overcome the looming energy crisis. The objective is to reduce reliance on fossil fuels that are increasingly scarce, reduce reliance on non-renewable sources of energy, and create a system that can reduce the world's CO_2 emissions to combat climate change. To overcome these challenges, the FREEDM center is developing a revolutionary power grid with the following characteristics:

- It supports distributed intelligent control mechanisms;
- it enables plug-and-play of power resource and storage devices;
- it provides stability and reliability of power delivery;
- it improves energy efficiency; and
- it combines scalable and secure communications.

Today's power grids will ultimately transition to become a system with the FREEDM characteristics. Today's systems generally utilize dated technologies and are unable to provide high levels of fault tolerance as they operate under centralized control structures.

A high-level view of the FREEDM infrastructure design is depicted in Figure 1. The goal is to create an Internet for power that supports the incorporation of a variety of power sources and storage devices to operate in a plug-and-play manner. This includes incorporating a variety of green power generation mechanisms, such as photo-voltaic, wind, and hydro-power. In the proposed system, consumers can generate their own power and sell it back to the utility. Such micro grids feature plug-in hybrid electric vehicles (PHEVs), local wind turbines, and other consumer-level load and generation sources.

There are a number of challenges associated with a power grid with distributed generation sources in terms of CPS design in general and for FREEDM specifically. One of the interesting challenges is its Distributed Grid Intelligence (DGI). The goal of DGI is to facilitate the departure from centralized power control in favor of distributed control with multiple control objectives. DGI is developing two types of systems to be utilized in the power grid. The first is the intelligent energy management (IEM) system. IEMs are responsible for enabling the power grid to make distributed decisions, i.e., load balancing and system control. The second type is is the intelligent fault manager (IFM). IFMs are responsible for working with IEMs to detect faults and to make islanding decisions. Islanding refers to temporal isolation of a micro grid where selective loads are still served by local power generation capabilities.

Figure 1 shows the topology of the DGI system and its interface with the IFM and IEM nodes. DGI within the FREEDM system features a communication network through the Reliable Secure Communications (RSC) layer. RSC is investigating ways of integrating a complete communication system into the intelligent power grid. The network is composed of several different network types to support

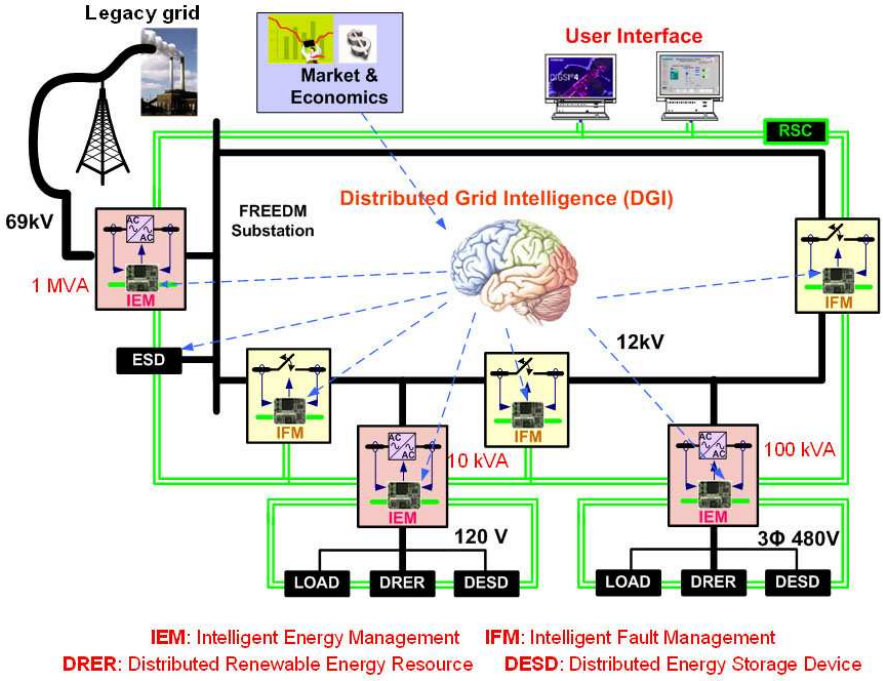


Fig. 1 FREEDM System: Locally generated power is fed into the power grid via solid state transformers; such decentralized generation requires a new level of Distributed Grid Intelligence (DGI) to monitor fault and manage energy in order to ensure sustained reliability compared to centralized power generation.

the scope of devices in the project. The current design of the RSC network is a hybrid network that combines a wireless in-home design with a wired external design. The current model of the in-home design is that of several wireless ZigBee devices that communicate through a StarGate concentrator in the home. This is important for many reasons. First, this allows the IEMs in the system to have a more accurate assessment of the current load of a house. Based on the information, future loads are predicted. Second, this allows the intelligent power grid to shut off non-essential devices in a home during critical times. Third, this enables power generating homes to sell back energy to the power grid in times of excess supply.

4 Distributed Control for the Power Grid

In micro grids with a FREEDM design, CPS control needs to ensure system reliability in the sense of sustained power supply to costumers. To this end, this work focuses on the cyber-networking side of providing fault tolerance for micro grids.

Micro grids are a significant deviation from modern power grids. Today's power grids, particularly their hierarchical control below the substation level, serve as the closest analog for us to envision improvements for the overall power architecture. This work will inherently benefit and eventually change the overall power grid through efforts to develop techniques that improve the operation of micro grids.

The common design of today's power grids follows a centralized command and control structure, *i.e.*, most notably Supervisory Control and Data Acquisition (SCADA) systems relying on human monitors for decision making. SCADA systems provide the mechanism for identifying faults. However, they represent a single point of failure within today's power grid. Even when SCADA systems are running within specified parameters, catastrophic faults can occur.

The most severe faults are cascading failures, which occur when some initial (power) nodes in the physical power system fail at first. Upon failure, their power load is passed to another local node. When this occurs, it can overload the node that received the shifted load. When it fails, its load is passed on in a transitive manner, which may result in cascading effects. Conventional power grids provide little protection against cascading failures. As demand for power increases and the complexity of the power grid increases due to micro grids, the ability of the power grid to support the increased load may result in more frequent blackouts.

The overloads described above are common in power grids and were responsible for historic blackouts, such as in 2003 [1]. In August of 2003, a blackout occurred that affected 45 million people in the US and 10 million people in Canada. Several estimates say the cost of this blackout exceeded $6 billion dollars. The original cause of the power failure occurred due to overgrown vegetation that struck power lines. After the power lines failed, a series of cascading failures occurred resulting in an 80% loss of power in the Northeastern US. The SCADA system within the region of the original blackout failed to detect the faults due to a race condition in its software that caused the centralized system to fail. As a result of the failure of the SCADA system, human monitors failed to be alerted to the problems for over an hour. The missing alerts from the SCADA system caused the human monitors to disregard a phone call that would have pointed them to the cascading failures. Due to these failures, 256 power plants went offline that day.

The problems of cascades like to 2003 blackout might have been averted had a smart grid been in place. In essence, the centralized nature of the SCADA controller creates a single point of failure. This design resulted in a system failure at an inopportune time that led to the Northeastern blackout. Due to the distributed nature of decision making components in a smart grid, it would take many more faults before these systems were to fail.

Trends to decentralize control, such as distributed controllers to locally isolate faults, load power factor corrections and voltage regulators for generation sources, help conventional power systems to reduce the threat of cascading faults. However, we conjecture that with the projected wide-spread deployment of micro grids, additional safeguards are required.

The FREEDM model features IEM nodes and IFM nodes that are distributed throughout the micro grid. In the event of a failure, the IFM nodes would disconnect

the breaker when detecting failures and notify the IEM nodes. If a local controlling IEM node that governs high-level decisions of other nodes fails, the remaining IEMs can distribute the load to accommodate the loss. When cascades occur, the IEMs can identify the fault and circumvent further cascades. They can resort to islanding to isolate the micro grid from either incoming or outgoing faults.[2] This may result in internal partial or complete failure but it would stop further damage to the remaining components. If the failure originated from outside, the micro grid would be isolated from the cascade. Secondly, the IEMs can redistribute the load through immediate control of the transformers used within the network. Zig-Bee nodes inside homes may provide feedback to the IEMs so that IEMs could shut off unnecessary loads to reduce power flow.

The challenge here is to devise a mechanism to support distributed control, a key component of which is a strong communications network. In a conventional network design, routers and switches present a single point of failure, just as in a SCADA network, due to the static nature of routing protocols. If a switch were to fail in a critical location within the network, even if redundant pathways existed, many commodity networks could not exploit such pathways as switches generally do not provide dynamic routing capabilities. In the above example, this could lead to misinformation being disseminated throughout the network. If these types of failures existed at the time of faults within the power network, it would be difficult to circumvent faults or operate effectively in spite of their presence. To improve upon this paradigm, the communications network must be one of the most robust components of the system. In a robust communication network, failure results in the reorganization of communication pathways. This allows messages to still be transmitted to all operating nodes in the network by routing around failed components. Thus, CPS systems in general and the power grid in particular become more resilient.

Let us briefly describe how such distributed control with fault-tolerant networks generalizes to other CPS infrastructure. There is a large scope of applications, especially within critical infrastructure, that could potentially move away from centralized SCADA systems. Researchers are rolling out components in communities to monitor underground water pipes. These devices monitor the flow on the pipe to maintain pressure as well as monitor for slow leaks. The current scope of these devices is to record this data so that it can be collected. But as this technology improves, it will provide real-time feedback to utilities to help them quickly identify failures [17]. Another example is oil refineries that currently use SCADA systems to collect the data complemented by humans monitoring the system. This exposes SCADA to human mistakes with potentially severe consequences. Distributed control shifts responsibility from human operators and creates a decentralized system that reacts to faults in a more robust manner. Fault tolerant communication infrastructure would increase reliability in both of these scenarios.

[2] Incoming faults are predicted cascades that originate outside of the micro grid with a potential to destabilize local power balance (or even damage local power devices) while outgoing ones originate inside the micro grid with a potential to destabilize immediately surrounding power balance, below a substation or beyond (or even damage power devices in this realm).

5 A Resilient Network Model

The objective of this work is to provide resilience in CPS control specifically for the power grid. We consider existing physical infrastructure in terms of its topology, propose a cost-effective topological extension and abstractly analyze its resilience characteristics. To this end, we assume a model that is agnostic to the type of faults affecting the network. In other words, our approach works equally well with, *e.g.*, node failures and link failures, where a node is an IEM/IFM node in the FREEDM model or any other compute node in a power grid control system. The detection of such faults is orthogonal to this work and could be accomplished by timeout-based monitoring, such as in our prototype, assuming fail-stop fault behavior. Fail-stop behavior refers to failures where a compute device stops working altogether, i.e., other devices will not receive any response at all when querying the device. We do not consider Byzantine failures where devices may provide incorrect responses, either because their security has been compromised or due to partial hardware/software failures that produce incorrect results. Any loss of communication in our model is mitigated by attempting to find a route through the network that will bypass the point of failure and still deliver the message using a different route, albeit with potentially different (higher/lower) latency than before and potentially with a reconfiguration delay due to timeouts.

As a starting point, we assume a tree topology as our network topology. The tree topology is a good fit for modern power grids that are hierarchically designed, and power line corridors owned by power utilities often already harbor cyber network lines linking control and monitoring devices along the hierarchical structure.

Resilience can be improved by utilizing redundant physical network paths. Assuming that sufficient paths exist, software overlays may be utilized to improve network resilience, an idea first described by Anderson *et al.* [6]. Their work presented the basis for a resilient overlay network (RON) by partitioning distributed nodes that may contain a different topological perspective than the external, physical network topology. Their work assumed nodes to potentially be geographically scattered across the Internet, i.e., their topology assumed inherent physical link capability for multi-path routing.

Our work differs in that we assume a proprietary network topology that may lack multi-path routes in its current design due to the strictly hierarchical structure of the power grid. Nonetheless, we utilize a similar partitioning for the routing of messages. In contrast to RON, our overlays focus on much smaller local area networks (LAN) to facilitate fault-tolerant communication in a micro grid setting. As such, it complements switches found in LANs due to their low cost with advanced fault-tolerant routing capabilities otherwise only available for expensive routers. When multi-path routes are already available in wide area networks (WAN) at higher levels of the power grid, our method can equally be applied, but our focus in this work is on the local side.

The physical network is assumed to be implemented as a tree network topology by default, which is consistent with today's power (and corresponding proprietary cyber network) infrastructure. Communication in this model follows that of a typical

network in which messages are sent from switch to switch. The standard communication links in this model are referred to as uplinks. As shown in Figure 2, uplinks are the vertical communication lines that create the tree structure. These vertical lines represent today's physical infrastructure.

Our method complements vertical connections with a set of horizontal links placed at various points throughout the network, effectively creating a cross-linked tree, as depicted in Figure 2 (discussed in detail later). These links, designated as crosslinks, are only intended to be used in fault scenarios if we assume that initial communication follows the legacy, vertical links. During link outages, *e.g.*, when nodes start incurring timeouts for sending messages, the nodes incurring the timeouts will transparently morph routing from the path given by the physical switch network to specially designated crosslinks between lateral nodes. These crosslinks facilitate the delivery of messages upon partial link/node failures.

Notice that crosslinks are still periodically monitored by sending heartbeat messages between both endpoints, but these heartbeat messages do not carry any power communication payloads. Such monitoring ensures that crosslink status is known in case of link/node failures to facilitate the discovery of an alternate routing path.

The distinction of vertical and horizontal links here is mainly to illustrate the additional investment required in physical infrastructure. And when new leaf-level (consumer) infrastructure is installed, initial paths are established over vertical links. During 24/7 operation, however, distinction between horizontal/vertical links becomes impertinent as the main objective is connectivity. While latencies due to number of hops may differ, worst-case response times are calculated based on upper bounds considering the longest path though our topology.

Let us next outline the operational model of routing messages in a software overlay over the given cross-linked tree. Each node in the tree contains a prioritized list of nodes containing crosslinks within a predefined radius r relative to their physical location. By enumerating these lists and passing messages via crosslinks, dynamic routes are created throughout the network.

The details on how to implement this model over existing network devices, such as routers and switches, are provided in Section 6.1. Before considering implementation variants, let us first analyze the resilience characteristics of the cross-linked tree, i.e., we are deriving a probabilistic model to study graph partitioning as an indication for disconnected sub-networks (in the cyber sense).

The partitioning/isolation property signifies the likelihood of a communication (cyber) network outage in power grids. Such disconnects in general may result in power outages or reduced capacity islanding due to lack of coordination within the sub-network of the power grid, as detailed next. While the legacy grid continues to supply power during a cyber outage, power efficiency may degrade in the absence of micro-grid control. In the event of simultaneous failure of connectivity to the legacy grid (*e.g.*, when physical power lines and communication lines are clipped simultaneously), outages are unavoidable. In micro grids, in contrast, cyber isolation still allows islanding if a micro grid has generation capabilities, but only for a selected subset of quintessential loads while all other devices remain without power. Hence,

cyber isolation serves as a basis to quantify the reliability of the overall system and that of individual nodes.

Our objective is to keep the probability for such partitioning (the likelihood of isolation and potential outages) low while controlling the cost of software overlays due to retrofitted cross links. Let us assume that a given single unit may fail with a probability of p. For simplicity, we assume an equal probability for node and link failures here.

Let us express the cross-linked tree as an abstract graph $G = (V, E)$ of vertices V and edges E, where the former combines nodes and switches while the latter represents network links. The height of the graph is denoted as h. In the graph, vertical tree edges T are distinguished from crosslinks C, such that

$$E = T \cup C \wedge T \cap C = \phi.$$

Let $v = |V|$ and $e = |E|$ be the number of vertices and edges in G where

$$v = 2^h - 1 \text{ and } e = 2^h + 2^{h-2} - 3 \tag{1}$$

Consider the (transformed) graph representing an overlay tree depicted in Figure 2 with a height of $h = 4$. G has a total of 15 nodes and 17 links for the example in Figure 2, as given by the above equations. Notice that smaller trees are irregular with respect to crosslinks, as discussed below, i.e., the selected height provides the smallest example of an otherwise scalable generalization.

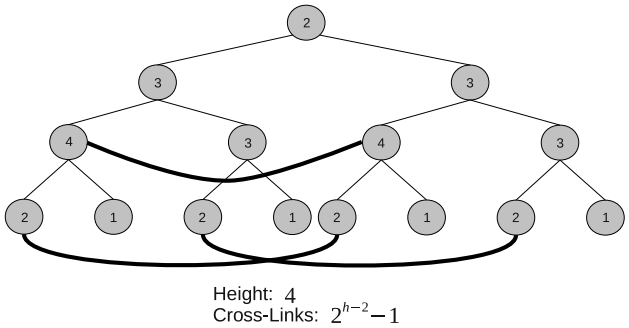


Fig. 2 Sample Overlay Tree: Virtual links overlay the physical cyber network topology with bi-directional connections that form a (in this case) binary tree of height four plus crosslinks (bold links) to increase the reliability. The core tree reflects common physical cyber network topologies close to homes (i.e., the generation sources for micro-grids) while crosslinks represent proposed, additional connections.

We then derive a probabilistic model based on graph analysis and combinatorial theory. Our overlay graphs have a number of unique properties that we utilize: Any vertex has a degree of $d \in \{1, \ldots, 4\}$ (number of edges), including crosslinks. In Figure 2, crosslinks are depicted in bold, and vertex degrees as the label per vertex. At each level $l > 2$, crosslinks are created to connect every other node at the

respective level. Hence, the total number of crosslinks is $2^{h-2} - 1$. This guarantees a uniform distribution of crosslinks that remains proportional to the growth of the overall tree. More significantly, we intend to show that graph connectivity is preserved with at least the same probability as the tree grows, which provides a stability invariant. The consistency and resilience of our model under scaling of power grids to large sizes can thus be exposed via reasoning over strong guarantees for stability and, implicitly, resilience to failure.

Stability is derived in terms of resilience to graph partitioning. Depending on a component location in the graph, partitioning may only occur under a certain combination of simultaneous failures. It suffices to consider single, double and triple failures of units in this model: single link (L), single node (N), double link (LL), double node and single link plus single nodes (LN) failures etc. based on the independent per unit failure probability p. Using combinatorics, the number of failures that results in graph partitioning (isolation) of at least one vertex (node) can be enumerated per class (see Table 1). The table shows the unique failures (omitting identical pairs and isolation of lower degree nodes since units are unordered in their enumeration). For instance, a single-link leaf becomes isolated when its parent or its link fail. A dual-link leaf can be isolated when both its links (1 case), a link and a node on opposite sides (2 cases) or 2 nodes fail (1 case), where multi-partitioning is only counted once. For triple-link nodes at level $h - 1$, two cases each exist with unique partitioning. Any other vertex cannot be isolated by just a dual failure. It would require triple unit failure for degree 3 nodes above level $h - 1$, such as LLL (12 cases), and so on. Due to the low degree of vertices in cross-linked tree (by construction), this covers all cases for larger partitions as well.

Table 1 Enumeration of Isolation Scenarios

# Nodes	**Degree**	case 1	case 2	case 3	case 4
2^{h-2}	1	1 L	1 N		
$2^{h-2}+1$	2	1 LL	2 LN	1 NN	
2^{h-3}, l=h-1	3	2 LL	2 LN	2 NN	
2^{h-3}, o/w	3	12 LLL	12 LLN	12 LNN	4 NNN
2^{h-3}	4	4 LLL	12 LLN	12 LNN	4 NNN

Notice that multi-unit failures are counted only once by ensuring that only (a) nodes on independent paths (without common vertices) and (b) links on edge-independent paths (without common edges) are counted. The former is also captured by the minimum vertex cut while the latter represents the minimum edge cut (see Menger's theorem [8]). All unique cuts need to be counted once, and higher degree cuts subsumed by lower degree cuts can be omitted. However, non-omission only increases the overall partitioning probability insignificantly since higher-degree cuts are significantly less likely than lower ones. Some lower cuts are included at higher degrees in Table 1 to simplify the problem.

Example: Consider a failed link between the root node and its left child in Figure 2. A second simultaneous link failure between this child and its 3rd level left

child would be included in the upper link (single) failure consideration, but also counted separately in our approach to provides a closed formula. This formula still presents a sound upper bound approximation that is tight as argued next.

The systematic structure of our overlay graph construction ensures that the number of these cuts remains constant as the height increases, which is significant for the stability argument in terms of resilience. The approach is thus sufficient to characterize network stability by absence of partitioning. The overall partitioning (isolation) probability P can then be approximated (by omitting any additive constants) as follows, where each term corresponds to the respective entry in Table 1:

$$\begin{aligned} P \approx & \frac{2^{h-2}}{e}p + \frac{2^{h-2}}{v}p \\ & + \frac{2^{h-2}}{e^2}p^2 + 2\frac{2^{h-2}}{ev}p^2 + \frac{2^{h-2}}{v^2}p^2 \\ & + 2\frac{2^{h-3}}{e^2}p^2 + 2\frac{2^{h-3}}{ev}p^2 + 2\frac{2^{h-3}}{v^2}p^2 \\ & + 12\frac{2^{h-3}}{e^3}p^3 + 12\frac{2^{h-3}}{e^2v}p^3 + 12\frac{2^{h-3}}{ev^2}p^3 + 4\frac{2^{h-3}}{v^3}p^3 \\ & + 4\frac{2^{h-3}}{e^3}p^3 + 12\frac{2^{h-3}}{e^2v}p^3 + 12\frac{2^{h-3}}{ev^2}p^3 + 4\frac{2^{h-3}}{v^3}p^3 \end{aligned}$$

The overall partitioning probability has multiple implications. First, the probability of graph connectivity remains constant since denominator and numerator grow at the same rate since $2^h - 1 = 2^{log v} - 1 = v$ (see Equation 1). This indicates that graph connectivity remains *stable* regardless of tree height. Second, for a large number of nodes, partitioning only depends on the probability p for single node/link failure, *i.e.*, our overlay is *scalable*.

Using numerical approximation, these properties become obvious by another simplification step based on the fact that $v \approx e \approx 2^h$ (due to Equation 1):

$$P \approx \frac{1}{2}p + \frac{7}{4v}p^2 + \frac{9}{v^2}p^3 \text{ and } \lim_{h\to\infty} P = \frac{1}{2}p \tag{2}$$

Notice that h is going to grow significantly to accommodate an explosion of devices with a wide-spread deployment of micro grids with IEMs/IFMs.

Moreover, we conjecture that fewer crosslinks would actually suffice as along as they were growing at a rate of at least $O((log_2 n)^2/n)$ total crosslinks for any cross-linked graph, such that first-order failures (single link/node) increase by only a constant factor. Such a refinement is subject to future work. But it may have practical value as a lower constant implies a potential for proportional cost savings when retrofitting trees with crosslinks within the physical infrastructure.

The placement of crosslinks poses another interesting aspect. In the analysis, an equal distribution of connections across a level is assumed. An algorithm for systematic crosslink placement can be constructed by "alternating" their node source/sink such that the subtree rooted in node v with the lowest aggregate cross connectivity distance $c(v)$ is connected to its equivalent neighbor at a distance l linear to the re-

spective height, where $c(v)$ is defined as the sum of the shortest paths from nodes of a subtree (rooted in v) to the nearest crosslink. Let us outline two such algorithms with different growth rates.

Algorithm: At each level at height $h, h \geq 3$, $2h-4$ crosslinks are created, where $l = 2h-3$ and $c(v) = min_{v_c \in V}(|v - v_c|)$, i.e., c is the minimum distance to the nearest cross node v_c in G. The number of crosslinks grows proportionally to n, i.e., it is upper bounded by $2h^2 = 2(log_2 n)^2$, which is less than $O((log_2 n)^2/n)$ total crosslinks (for $h \geq 3$).

The algorithm is configurable in terms of the number of crosslinks. For example, a lower number of just $h-2$ crosslinks could be established per level for the same l and c, subject to the same upper bound. However, any lower rate of crosslinks per level, e.g., $log_2 h$, would violate the bound and result in insufficient alternate routes to ensure stability and, hence, scalability under sustained resilience.

Notice that c can be calculated by depth-first-search (DFS) in linear time. An even more efficient algorithm to calculate can be incorporated into the systematic construction of G using the following steps. (1) The initial c of root is zero. (2) c is then inherited from a parent to the children and incremented by one upon creating a child node. (3) Upon construction of a crosslink, a subtree of height $log(h)$ has to be updated for each node receiving a new crosslink, where such a node is a leaf in the respective subtree. The resulting complexity is only $O(log(h)^2)$ instead of linear DFS complexity.

6 Realistic Network Overlay Designs

Cross-linked trees provide a theoretical basis and a means to reason about reliability. In the following, different design options for mapping and embedding these cross-linked trees into existing networks and with conventional network devices will be developed. The main objective is to provide resilience while keeping the cost of retrofitting networks low. This is mostly a consideration in terms of the required crosslinks as the overlay protocols can be established in software.

6.1 COMIG: A Communication Overlay for MIcro Grids

The first approach considers devices organized into software partitions that are calculated locally based on their IP address. Partitions are created as a side effect of subnet masks. Each partition is assumed to be locally connected to a switch. These partitions are then grouped together in clusters of a certain static size. The combined group of clusters and partitions are interconnected with horizontal crosslinks and vertical uplinks. Figure 3 depicts example of COMIG.

Our software overlay network represents a tree-based topology utilizing vertical uplinks. These uplinks serve as the default routing path for general message

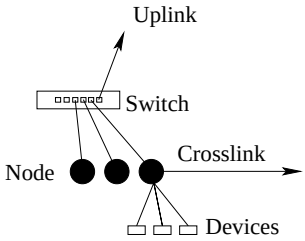


Fig. 3 Device Cluster: Nodes are connected to switches with crosslinks from a node to another tree trunk.

communication in the absence of failures. Figure 4 depicts the vertical uplinks and shows the resulting tree formed by them. Uplinks are necessary to provide inter-cluster communication. They constitute the network backbone of COMIG. To increase fault tolerance, horizontal crosslinks are introduced. Figure 4 depicts these crosslinks, which serve as secondary paths through the network, activated by the overlay network protocol upon primary path failure.

A COMIG overlay is an abstract software overlay that fits arbitrary intelligent power grids. Most importantly, it provides redundant communication pathways and the potential to connect the network in alternate ways in case of faults in the system via its software middleware layer. This capability is crucial for allowing intelligent nodes in the system to coordinate the actions of system control tasks and to maintain appropriate state.

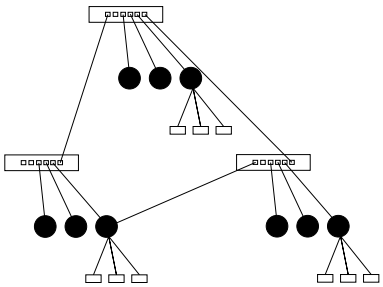
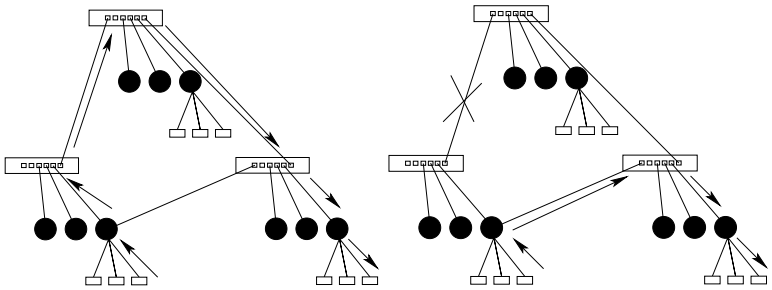


Fig. 4 Cluster Tree: Uplinks connect switches while crosslinks connect a switch to a node on a second network port.

Communication pathways are primarily used through the switching interface composed of uplinks, as depicted in Figure 5(a). COMIG differs from a regular network in the composition of a series of intelligently placed crosslinks that can be implemented as node-to-switch or node-to-node links. The abstract network will enter into a reorganization mode upon loss of an uplink connection.

A message timeout is utilized an indicator for link failure and results in reorganization. The reorganization mode explores alternate routes in the network based on meta-information describing the characteristics of the network. The collection of this meta-information occurs as follows. Nodes can derive partition information from their network overlay data, e.g., to determine its neighbors on a switch and

the partitions above and below it in a tree. A node in reorganization mode can communicate with its neighbors to determine the location of crosslinks. It can further determine if this is a node failure on the receiving end or a link failure along the switching path by utilizing the crosslinks. Figure 5(b) depicts the utilization of a crosslink in an attempt to resend a previously failed message. If the failure was in the switch link, then a switch link is indicated as a failure type by the nature of the response from the receiving node.



(a) Network Flow of Fault Free Tree (b) Network Flow in Presence of Line Fault

Fig. 5 Message Pathways: Without faults, messages travel along the up/down links of the tree; upon failure of an uplink, crosslinks dynamically re-route messages horizontally to another switch before traveling down (or up) to the destination node.

Overall, COMIG provides essential functionality to an intelligent power grid utilizing a distributed network. In case of wide area faults in the power grid, it aids distributed grid intelligence of the micro grid by ensuring reliability through reorganization.

6.2 SWOMIG: A SWitch Overlay for Micro Grids

COMIG has one pitfall: It relies on an overlay structure imposed on the network even at times when faults do not exist, which adds performance overhead. In general, overlays impose a trade-off of performance for fault tolerance that may initially not always be satisfactory but can be refined in terms of minor design changes.

A power grid may require only very moderate bandwidth while the conventional Internet may have higher bandwidth requirements due to consumer needs for streaming services. Nonetheless, certain levels must be maintained to insure timely decisions can still be made in power grids. This observation motivates our second design termed SWOMIG, a switch-based overlay for micro grids. While both COMIG and SWOMIG can operate agnostically of the underlying physical network structure, SWOMIG allows for static communication pathways to be used at times when faults are absent. In contrast, COMIG forces communication over abstracted routes with overhead even in the absence of failures.

Cross-linked trees are also the basis of SWOMIG's design with crosslinks that are disjoint from uplinks representing the static route of the network. This is easily accomplished through default routing configuration tools. In SWOMIG, during normal operation, the network utilizes the static pathways. This provides high throughput. In the presence of a fault, i.e., when a message timeout occurs, an overlay is imposed, but only on the node that experiences a timeout. Each node maintains a list of surrounding nodes' crosslinks. In the presence of a fault, a node determines if an alternate route exists to transmit the messages by traversing the list of crosslinks.

An example of a commodity configuration that can self-organize via discovery of alternate routes is depicted in Figure 6. In the first part of the figure, the commodity network is using the default pathways to enable communication. The remaining portion of the figure explores possible reorganizations using crosslinks. We primarily consider link failure in this example because the communication network may parallel the physical transmission corridors of the power grid. In such a case, simultaneous power and cyber network cuts may occur due to external (physical) intervention, such as fallen trees or unintentional construction-related line cuts, just to name a few.

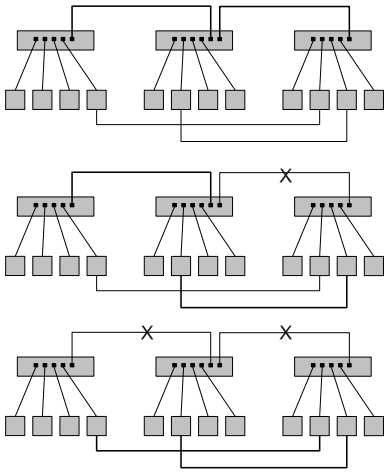


Fig. 6 Device Cluster: Switches are directly connected to one another on the default path; upon failure of links on the default path, traffic is re-routed via crosslinks that connect two nodes with one another across distant switches.

Components of devices themselves tend to be physically protected in hardened enclosures to provide protection from weather and other environmental stress, but may be subject to electrical faults. Links are more exposed to the elements and may experience disconnects when cut. (Recall that the analytical model considers node and link failures to have equal probabilities of failure. If these probabilities were to significantly differ, a refined model is needed, which is subject of future work.) These lines running parallel to the power lines represent the default path. Crosslinks can be implemented using commodity Internet connections or specialized lines connecting nodes such as IEM/IFM nodes in micro grids, optionally not

using above-ground cabling to further protect them. In this example, link failure is used as a cause of fault in the network. Another scenario is device failure, where a switch or a node fails. In the case of switch failure, if subnet partitioning were exploited to provide localization information, this information could help isolate the fault location. In this case, local communication among nodes on a switch would fail within the network. Nonetheless, this failure can be confirmed by a representative in the network with a crosslink as an alternate path to replay a timed-out message.

6.3 Discussion

In practice, the cost of crosslinks depends on the deployment methods. Wired connections between end-points of micro-grid generation sources (homes) may require trenches across corridors not owned by power companies. A more viable method may be wireless connections, which could be realized by short-range Wifi (802.11) or medium range Wimax technology. Wired is more robust and thought to be more secure but the low cost of wireless and ease of installation may be more realistic. If we assume a hybrid deployment of (existing) wired connections along the tree links combined with wireless crosslinks, the default connectivity (the uplink in the tree) would be more reliable than crosslinks. Thus, crosslinks become natural backups for broken uplinks, where the latter can be repaired while the former ensure sustained connectivity. This hybrid approach seems to provide a good cost-reliability trade-off. Inclusion of different reliability levels for uplinks vs. crosslinks is subject to future modeling.

7 Implementation and Interface Definition

We have designed and implemented a unified message passing API that facilitates coordination between nodes ranging from the large and sophisticated IEM nodes to small ZigBee devices. An API is important for the development of applications for a complex system of software, such as a power grid. Using this API, we can guarantee a common messaging-passing standard that will be utilized ubiquitously within micro grids (and possibly above). This API has currently been deployed by other software teams within the FREEDM project and is being used to create applications for load balancing and power management as a proof-of-concept. We have also found the API beneficial for creating and testing the implementation of our overlays.

We design the API to loosely resemble that of Active Messages [9] as implemented in Tiny OS [12]. In particular, messages are non-blocking and asynchronous. This design choice allows less sophisticated devices that simply use a MAC-based designation to be incorporated into the network. Such low-end devices can then be accounted for by more sophisticated nodes. In this message passing API,

a device or node registers a message type to receive a message handler. The handler is then used in sending and receiving messages. The current API provides a number of constructs for basic communication using point-to-point messages. These include

- non-blocking sends and receives,
- conditioned waiting and signaling, as well as
- handle generation.

Non-blocking network abstractions facilitate resilience in network overlays when faults are considered at arbitrary rates and when timeouts are utilized for fault detection in a distributed network. This allows devices within the network to send messages without waiting for acknowledgments before proceeding with other work. The same approach is applied to receives to avoid a need for actively monitoring a queue. In a non-blocking approach, a received message is handled by the network API. When a new message is received, the application is able to use it right away or defer it until a later time. From this asynchronous API, one can thus create blocking semantics of a layered blocking API if desired. This is done using the conditioned wait and condition signaling methods supported by our API. This allows a running process on a device to send a message and then blocks. Once a corresponding message is received, the same process is woken up again.

We utilized the Mace distributed prototyping language [11] to implement this API. Mace is a C++ abstraction that enables the low-level network details to be abstracted from the programmer while leaving significant amounts of flexibility in the message-handling abilities and supporting timeout-based fault detection, which is central to our fault tolerance network overlay approach. The basic prototype of our proposed system on top of Mace comprises a universal basis for message passing over our API in the FREEDM infrastructure.

8 Network Overlay Monitoring

Locating faults in a conventional power grid can present a challenge. Current fault localization practices often require the operating utility to field phone calls that allow it to determine a rough location of where the fault may have occurred. Using such a rough estimation approach can increase the duration of the power failure. Smart grids have the capability to remedy this situation. The distributed nature of control within a smart grid allows agents to detect faults much faster than complaint triangulation. When these faults are detected, the discovering nodes can report the node failure. The identification of the failed (cyber) node should increase the accuracy of locating the fault in the (power) network. We are developing a distributed live monitoring (DLM) tool that identifies failed power devices among other features to aid in fault localization.

The ability to understand the structure and status of the network is imperative, particularly when the network is distributed in nature. A truly global status may often be difficult to obtain due to the distributed nature of the network. To aid the

maintainers of the system in identifying problems and correcting them, our DLM tool provides a real-time view of the state of the networked devices in the system and the dynamic routing through our software overlays. This system provides information regarding a node's current running status as well as a topological layout of the network, both derived from data provided to the operational model.

Nodes can communicate with an interface server in our first prototype, which features a centralized single server. The server provides a graphical representation of the status of the nodes and current messages in flight. The projected design of the DLM presents a fully detailed representation of the underlying LAN. This enables one to monitor system activity, to detect failed components, to observe alternate routing activity, and to sustain partial functionality in the presence of partitioning / islanding of micro grids. As such, one can determine which links have failed and, more specifically, which nodes have failed. Working nodes report the status of successful and failed communication attempts within the network. This information is relayed to and concentrated at the server to allow visualization.

The DLM is used to display the communication paths of three separate devices as depicted in Figure 7. The DLM can be provided with information detailing the locations of software links between nodes to create a graphical representation of the network. The network structure is being coded as a tree network resembling the shape of the network utilized in our Mace prototype in this figure.

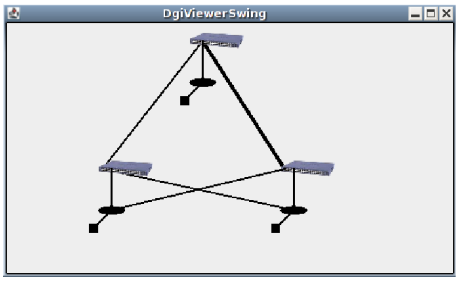


Fig. 7 Distributed Live Monitoring Swing Window

The DLM server and its node components (runtime support / daemons) were developed using the Java Swing graphics packages. We utilize a network socket API to connect with our distributed prototype written in C++. The Mace distributed program library provides the distributed message abstraction and inherently supports fault detection [11]. Our implementation supports a variety of services, such as the ability to

- monitor and set connection status,
- define links and partitions,
- visualize paths and messages, as well as
- topologically arrange output in a hierarchical manner.

The current implementation assumes a cross-linked topology but the principle design supports other topologies for visualization as well. The prototype of the message-passing system is instrumented with calls that relay messages during each critical step in the program communication path. At each step, a command message is sent to the visualizer that renders the information on screen. Failed nodes may not be able to report their current status to the DLM. The status of a node is updated should another node detect link/node failures via timeouts.

9 Future Work

Current limitations of this work in SWOMIG make it very difficult to compose multiple overlays into a single path. A single path would be beneficial in situations where multiple failures occur. Multiple composed overlays could be used to facilitate a single path throughout the network, combining all of the nodes. The problem here is that current localization data is limited to determining neighbors on switches. This provides very little information relating one group position to another. In such a system, to compose two overlays blindly would require a recursive enumeration of crosslinks in the network to identify a single path. This exhaustive type of search is inappropriate in terms of its latency within a network and would not scale well. An alternate approach to this that we will be investigating in the future is to not only provide crosslink information to the nodes in the network. One would also provide localization information that can be used by switches to identify locations of their partition in relation to other partitions surrounding them with a constant set radius. We can then explore statistical means of deriving the best trade-off of composing multiple paths to improve fault detection with a higher radius.

Another future direction of research focuses on scalability by utilizing crosslink offload. In some ad-hoc networks, studies suggest that, after scaling to a certain size, considerable processor time is spent passing other nodes' messages around, which makes it difficult to make computational progress anymore [13]. This type of fault, similar to life lock, could easily affect our current model if the physical network structure had a single crosslink connecting two halves of the network, which could overwhelm the single connecting node. To overcome this challenge, we will study models that include both priority and overhead evaluations. A priority evaluation allows the use of global priority values to be assigned to nodes. In negotiating the use of a crosslink with another node at times of failure, the priority value ensures that important nodes can communicate at the cost of the lower-level nodes that may be shut out. This would also have to account for a load metric that crosslink nodes maintain to insure that their own tasks can be accomplished — unless, of course, the crosslink node is a low priority node, in which case it would have a mode change to serve only as a message-passing link for other high-priority nodes.

10 Conclusion

Today's increased prevalence of intelligent devices in critical infrastructure imposes a need for fault tolerant communication. Automated decisions actuated by devices of such infrastructure can have a direct impact on the environment and human life. In an intelligent energy grid, this may include decisions on supplying power to critical devices, such as medical life support systems, while shutting down power to non-critical devices of a hospital.

This work contributes a fault tolerant communication mechanism for micro grids at low cost and high scalability. Such a provision enables IEM and IFM nodes to communicate, even in the event of multiple link failures. The first step to accomplish this is through introducing increased but intelligently distributed redundancy in the links of the network. We introduced a framework of middleware components that utilize software overlays to support fault-tolerant communication. The network overlay proves to be resilient by exploiting redundancy through utilization of alternate communication paths at the software level, and it is shown to provide stability in terms of sustained resilience as networks are scaled up in future micro grids.

The work further allows cheap switching equipment to be deployed as a means to complement legacy hierarchical network topologies. Since switches lack dynamic routing capabilities, our middleware realizes re-routing using the software overlay. This is significantly less costly than deployment of routers instead of switches. Our development of low-overhead route detection algorithms to assist in the presence of single and multiple link failures constitutes the key contribution to provide such fault tolerance in a transparent manner to other control software. Our middleware layer provides the means for higher-level distributed grid intelligence (DGI), such as providing hierarchical control schemes within this software overlay architecture. Thus, the vision of sustainable, scalable and reliable decentralized energy management on the software side in the FREEDM system and for other CPS domains in general is provided by our software middleware architecture for fault tolerant network overlays.

References

1. NERC final report.
   http://www.nerc.com/docs/docs/blackout/ch5.pdf
2. North carolina state university freedm project.
   http://www.freedm.ncsu.edu
3. Cacti: The complete rrdtool-based graphing solution [online] (2005). Http://www.cacti.net
4. Akella, R., Meng, F., Ditch, D., McMillin, B., Crow, M.: Distributed power balancing for the freedm system. In: in Proceedings of the 2010 Annual FREEDM Conference (2010)
5. Albert, R., Albert, I., Nakarado, G.: Structural vulnerability of the north american power grid. PHYSICAL REVIEW E **69**(2, Part 2) (2004). DOI 10.1103/PhysRevE.69.025103
6. Andersen, D., Balakrishnan, H., Kaashoek, M.F., Morris, R.: The case for resilient overlay networks. In: in Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems HotOSVIII, pp. 152–157 (2001)

7. Berthier, R., Bobba, R., Davis, M., Rogers, K., , Zonouz, S.: State estimation and contingency analysis of the power grid in a cyber-adversarial environment. In: NIST Workshop on Cybersecurity for Cyber-Physical Systems (2012)
8. Bondy, J.A.: Graph Theory With Applications. Elsevier Science Ltd (1976)
9. von Eicken, T., Culler, D.E., Goldstein, S.C., Schauser, K.E.: Active messages: a mechanism for integrated communication and computation. In: International Symposium on Computer Architecture, pp. 256–266 (1992)
10. Huang, Z., Wang, C., Nayak, A., Stojmenovic, I.: Small cluster in cyber physical systems: Network topology, interdependence and cascading failures. Parallel and Distributed Systems, IEEE Transactions on **PP**(99), 1–1 (2014). DOI 10.1109/TPDS.2014.2342740
11. Killian, C., Anderson, J., Braud, R., Jhala, R., Vahdat, A.: Mace: language support for building distributed systems. In: ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 179–188 (2007)
12. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., Culler, D.: Tinyos: An operating system for sensor networks. pp. 115–148 (2005). DOI 10.1007/3-540-27139-2_7. URL http://dx.doi.org/10.1007/3-540-27139-2_7
13. Li, J., Blake, C., De Couto, D.S., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 61–69. ACM, New York, NY, USA (2001). DOI http://doi.acm.org/10.1145/381677.381684
14. Massie, M.L., Chun, B.N., Culler, D.E.: The ganglia distributed monitoring system: Design, implementation and experience. Parallel Computing **30**, 2004 (2003)
15. Nguyen, D., Shen, Y., Thai, M.: Detecting critical nodes in interdependent power networks for vulnerability assessment. Smart Grid, IEEE Transactions on **4**(1), 151–159 (2013). DOI 10.1109/TSG.2012.2229398
16. Shao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Cascade of failures in coupled network systems with multiple support-dependent relations. CoRR **abs/1011.0234** (2010). URL http://dblp.uni-trier.de/db/journals/corr/corr1011.html#abs-1011-0234
17. Stoianov, I., Nachman, L., Madden, S., Tokmouline, T.: Pipeneta wireless sensor network for pipeline monitoring. In: IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks, pp. 264–273. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1236360.1236396
18. Tomsovic, K., Bakken, D., Venkatasubramanian, V., Bose, A.: Designing the next generation of real-time control, communication, and computations for large power systems. Proceedings of the IEEE **93**(5), 965 –979 (2005)
19. Varma, J., Wang, C., Mueller, F., Engelmann, C., Scott, S.L.: Scalable, fault-tolerant membership for MPI tasks on hpc systems. In: International Conference on Supercomputing, pp. 219–228 (2006)
20. Wauters, T., De Turck, F., Develder, C.: Overlay networks for smart grids, pp. 1–27. IEEE Smart Grid Research: Communications. IEEE (2013)
21. Yan, J., He, H., Sun, Y.: Integrated security analysis on cascading failure in complex networks. Information Forensics and Security, IEEE Transactions on **9**(3), 451–463 (2014). DOI 10.1109/TIFS.2014.2299404
22. Zimmer, C., Mueller, F.: The freedm architecture of fault tolerant network routing through software overlays. In: FREEDM Conference (2009)