Fault Tolerant Network Routing through Software Overlays for Intelligent Power Grids

Christopher Zimmer, Frank Mueller
Dept. of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, mueller@cs.ncsu.edu

Abstract

Control decisions of intelligent devices in critical infrastructure can have a significant impact on human life and the environment. Ensuring that the appropriate data is available is crucial for making informed decisions. Such considerations are becoming increasingly important in today's cyber-physical systems that combine computational decision making on the cyber side with physical control on the device side. 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 will enable multi-route discovery in an intelligent power grid. This will be 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.

Keywords

Distributed Networking, Distributed Fault Tolerance

1. Introduction

A cyber-physical system (CPS) governs physical devices through embedded control in a networked environment and has a direct impact on people who rely on such devices (*e.g.*, automotive control, medical devices, power grid). Failure of network equipment in a CPS can result in

- incorrect decisions regarding device failure,
- faulty decisions made due to lack of data,
- system reconfigurations, or
- degradation of system performance.

These types of failures are expensive and cause inefficiencies, especially in smart (power) grids that rely on commodity communication infrastructure. In smart grids, decisions are being made that affect the real world based on data passed within the network. This impact on the real world makes it necessary to improve communication within the smart grid to enable more reliable decision making.

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 of devices can be designed to contain multiple pathways to connect clusters of nodes in a redundant manner. If these pathways exist, a network needs to be able to alternate and utilize them at times of fault.

Traditional networks are composed of a multitude of devices many of which are aiming to maximize their own throughput. Commodity network equipment was designed to provide high levels of throughput. This design choice runs counter to the needs of an intelligent distributed network that will be utilized in 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. Example tasks the power grid must perform, such as distributed load balancing [1], substantiate this need. To accommodate the needs of such a system will require the system to collaborate intelligently at times of fault to insure communication.

In modern network topologies, network failures resulting in message delivery failure may be avoided through smart routing technologies that can bypass faulty equipment. 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. This severely limits the ability of CPS devices to sustain network failures through automatic re-routing over alternate path as switches lack such routing capabilities.

In this work, we 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. In related work, GridStat [2] enables the allocation of node specific redundant pathways for high-level power-grid networks. Our work is orthogonal to GridStat as it

This work was supported in part by NSF grant EEC-0812121 and U.S. Army Research Office (ARO) grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI).

is design 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.

The rest of this paper is structured as follows. Section 2 introduces the FREEDM system that this work originates from. Section 3 provides motivation for smart grids at large. A probabilistic analysis of a theoretical application of this work is outlined in Section 4. Section 5 presents two designs that utilize our approach. Sections 6 and 7 describe the API this work contributes for power micro grids and the distributed visualization tool that coincides with this work. Section 8 summarizes the contributions.

2. FREEDM Project

The NSF Engineering Research Center (ERC) for Future Renewable Electric Energy Delivery and Management (FREEDM) is a multi-institutional project that is investigating the cyber-infrastructure of micro grids. The overall goal of the center is 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 will reduce the world's CO_2 emissions to combat climate change. To overcome these challenges, the FREEDM center is developing a revolutionary power grid that will

- enable plug-and-play of power storage and resource devices,
- allow for distributed intelligent control,
- couple a scalable and secure communications infrastructure,
- improve efficiency,
- and provide stability guarantees [3].

Systems such as these will ultimately replace today's power grids. Today's systems generally operate under centralized control structures and utilize dated technologies and are unable to provide high levels of fault tolerance.

Figure 1 shows a high-level view of the FREEDM project. The goal is to create an Internet for power that will allow 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 load and generation sources becoming available to the consumer.

FREEDM boasts significant CPS challenges. 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 will be 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 will be 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 will feature a communications network through the Reliable Secure Communications (RSC). RSC is investigating ways of integrating a complete communications system into the intelligent power grid. The network will be composed of several different network types to support 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 will be important for many reasons. First, this will allow 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 will enable power generating homes to sell back excess energy to the power grid in times of supply. Third, this will allow the intelligent power grid to shut off non-essential devices in a home during critical times.

3. Motivation

The focus of this work is to ensure 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. By developing techniques that improve the operation of micro grids, this work will inherently benefit if not eventually change the overall power grid.

Conventional power grids utilize centralized command and control structures, *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. Cascading failures have been the most detrimental among these faults.

Cascading failures occur in the power grid when nodes in the physical power system fail. 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. These types of overloads are common in power grids and are responsible for many of the major blackouts that have occurred, such as most recently in 2003 [4]. Conventional power grids provide very little protection against cascading failures. As demand for power increases,

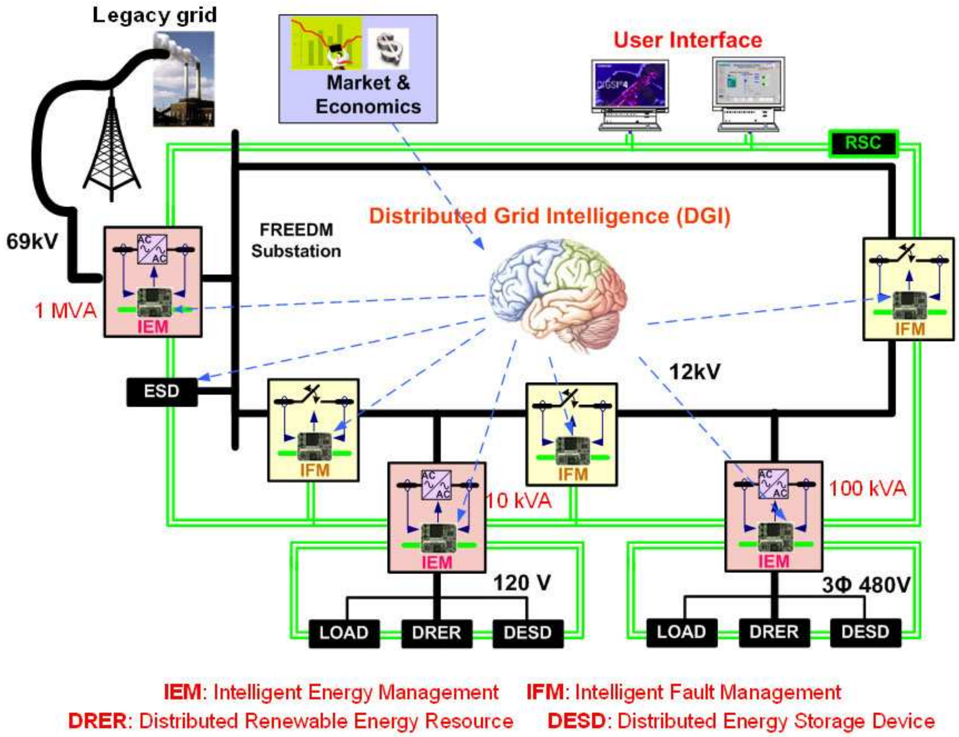

Legacy grid
Market & Economics
User Interface
RSC
69kV
FREEDM Substation
Distributed Grid Intelligence (DGI)
1 MVA
IEM
ESD
IFM
12kV
IFM
IFM
10 kVA
IEM
100 kVA
IEM
120 V
3Φ 480V
LOAD
DRER
DESD
LOAD
DRER
DESD
IEM: Intelligent Energy Management
IFM: Intelligent Fault Management
DRER: Distributed Renewable Energy Resource
DESD: Distributed Energy Storage Device

Fig. 1. FREEDM System

the ability of the power grid to support the increased load is going to lead to more and more blackouts.

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 northeast 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. Ultimately, 256 power plants went offline due to these failures.

Had a smart grid been in place during the cascade of failures, it may have been able to avert much of the problem. This demonstrates that 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 require many more faults to take these systems offline. In the FREEDM model, IEM nodes and IFM nodes 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. 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. IEMs could shut off unnecessary loads to reduce power flow through lines using the in-home Zig-Bee nodes that provide feedback to the IEMs.

For the smart grid to accomplish this, a strong communications network is critical. 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 very difficult to circumvent or operate effectively. To improve upon this paradigm, the communications network must be one of the most robust components of the system. In a robust communications network failure results in the reorganization of communication pathways that still allows for message transmission to all operating nodes in the network.

This provides a more resilient power grid.

The application of fault-tolerant networks is not isolated to power grids. 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 [5]. 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. Both of these examples would benefit from fault tolerant communication infrastructure.

4. Analysis

In this work, 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. 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. 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. We use a tree topology as our network topology. The network tree topology is a good fit for modern power grids that are hierarchically designed.

Using software overlays to improve network resilience is 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. Our work utilizes a similar partitioning for the routing of messages but deviates in that it utilizes this approach in a much smaller local area network (LAN) to facilitate fault-tolerant communication. 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.

In our basic model, the physical network is implemented based on a tree topology. 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. Also present in Figure 2 are a set of horizontal links placed at various points throughout the network. These links, designated as crosslinks, are only intended to be used in fault scenarios. 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. 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. Further detail regarding this model is provided in Section 5.1. The remainder of this section develops a probabilistic analysis showing that the basic model reduces the impact of outages (disconnects) in the network.

The goal of this analysis is to determine the likelihood of isolation for a given single unit failure probability of p. The isolation property signifies the likelihood of a communication (cyber) network outage in power grids. 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.

To facilitate the analysis, we transform the basic model into 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, tree edges T are distinguished from crosslinks C, such that $E = T \cup C$.

As a motivating example, consider the (transformed) graph depicted in Figure 2. We choose a height $h = 4$ for this overlay tree as smaller trees are irregular with respect to crosslinks (see below).

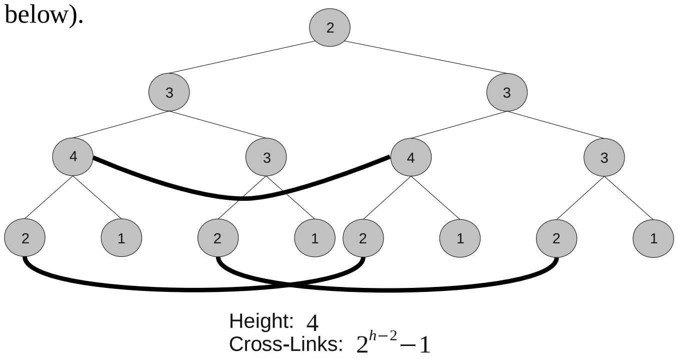


Fig. 2. Overlay Tree

Our probabilistic model is 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, which is depicted as the label of each vertex in Figure 2.

Crosslinks are created at each level $l > 2$ 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 strong guarantee for consistency and resilience under scaling of power grids.

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

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

for a total of 15 nodes and 17 links for the example in Figure 2.

We next consider single, double and triple failures of units on 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. We can enumerate the number of failures in each class that results in graph partitioning (isolation) of at least one vertex (node), as given in Table 1.

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

These are 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), etc. This covers all cases for larger partitions as well (due to the low degree of vertices).

More generally, 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 [7]). 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 (so that some lower cuts are included at higher degrees in Table 1 to simplify the problem). The systematic structure of our overlay graph construction ensures that the number of these cuts remains constant as the height increases.

The overall partitioning (isolation) probability P can then be approximated (by omitting any additive constants) as given as follows, where each term corresponds to the respective entry in Table 1:

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

This result has multiple implications. First, the probability of graph connectivity actually *remains constant* since denominator and numerator grow at the same rate. 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*. These properties become obvious by another simplification step based on the fact that $v \approx e \approx 2^h$:

$$P \approx \frac{1}{2}p + \frac{7}{4v}p^2 + \frac{9}{v^2}p^3 \texttt{ and } \lim_{h\to\infty} P = \frac{1}{2}p \qquad (1)$$

Moreover, we conjecture that fewer crosslinks would actually suffice as along as they were growing at a rate of at least $O(n/log_2 n)$ for our graphs, such that first-order failures (single link/node) increase by only a constant factor. We are considering such a refinement. Another interesting aspect is the placement of crosslinks. In the analysis, an equal distribution of connections across a level is assumed. We are currently developing algorithms for systematic crosslink placement.

5. Software Overlay Network

5.1. Communication Overlay for MIcro Grids (COMIG)

In our first approach based on the basic model, devices are 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 on 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. An example of COMIG is depicted in Figure 3. Utilizing vertical uplinks, our software overlay network represents a tree-based topology.

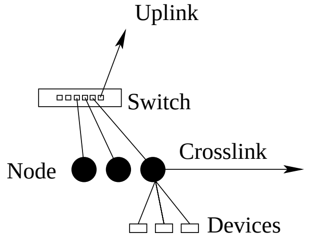


Fig. 3. Device Cluster

Similar work has been performed in the High Performance Computing domain by Varma et al. [8]. Uplinks serve as the default routing path for general message 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, it is necessary to introduce horizontal crosslinks that will serve as secondary paths through the network, as depicted in Figure 4.

This abstract software overlay of COMIG can fit onto 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 maintain appropriate state and to coordinate the actions of system control tasks.

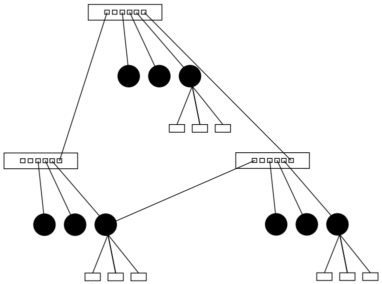
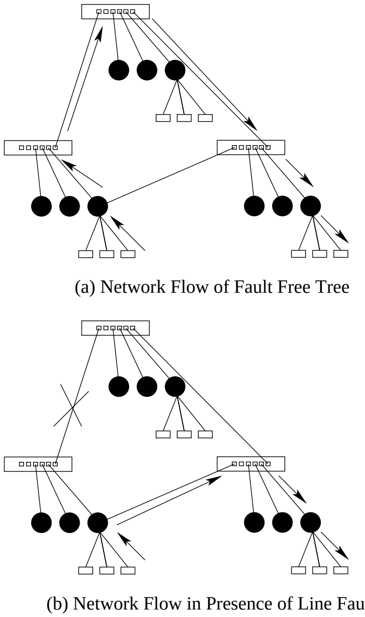


Fig. 4. Cluster Tree

In Figure 5(a), communication pathways are primarily used through the switching interface composed of uplinks. 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. In the event of a loss of an uplink, the abstract network will enter into a reorganization mode. Reorganization describes the actions resulting from a message timeout. In this work, the reorganization mode will explore alternate routes in the network based on meta-information describing the characteristics of the network. 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 and, by utilizing the crosslinks, determine if this is a node failure on the receiving end or a link failure along the switching path. Figure 5(b) depicts the utilization of a crosslink in an attempt to resend a previously failed message. A proper response from the receiving node indicates to the re-organized node that the failure was in the switch link.

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



(a) Network Flow of Fault Free Tree

(b) Network Flow in Presence of Line Fault

Fig. 5. Message Pathways

5.2. SWitch Overlay for Micro Grids (SWOMIG)

A weakness of COMIG is that it relies on an overlay structure imposed on the network even in times when faults do not exist. Overlays impose a trade-off of performance for fault tolerance that may not always be satisfactory. While a power grid may not always have as high of a bandwidth requirement as consumer networks, certain levels must be maintained to insure timely decisions can still be made. With this in mind, a second design was created with SWOMIG. While both designs can operate agnostically of the underlying physical network structure, COMIG forces communication into using abstracted routes while SWOMIG allows for the static communication pathways to be used in times where faults are absent.

SWOMIG's design also utilizes crosslinks that are disjoint from 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 will traverse the list of crosslinks to determine if an alternate route exists to transmit the messages.

Figure 6 shows an example of a commodity configuration that can self-organize via discovery of alternate routes. 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 and would be subject to simultaneous power and cyber network cuts. Generally, the device components themselves are shielded in boxes that protect them from the environment. But links are the most exposed to the elements. These lines running parallel to the

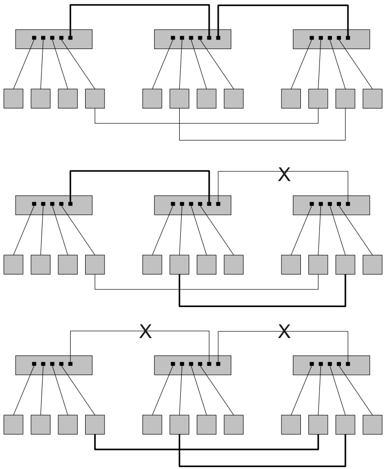


Fig. 6. Device Cluster

power lines represent the default path. Crosslinks can be implemented using commodity Internet connections or specialized lines connecting nodes such as substations, 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 amongst nodes on a switch would fail within the network, but a representative in the network with a crosslink would be able to confirm that failure.

6. API

To aid in the creation and testing of this work, a unified message passing API has been created 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 the power grid network. This API has currently been deployed to other software teams within the FREEDM project and is being used to create applications for load balancing and power management.

The API is non-blocking and asynchronous, similar to Active Messages [9] as implemented in Tiny OS [10]. 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 constructs for

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

Due to frequent faults and large timeouts in a distributed network, non-blocking network abstractions were designed to facilitate this work. 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. It is thus possible to create blocking semantics if desired. This is done using the conditioned wait and condition signaling methods in the API. Through these methods, a running process on a device sends a message and then blocks until being woken up by the receipt of a response message.

The API is being developed using the Mace distributed prototyping language [11]. 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. We have developed a universal FREEDM messaging-passing API and a basic prototype of our proposed system on top of Mace.

7. Distributed Live Monitoring

In a conventional power grid, locating of faults can be very challenging. 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 node should increase the accuracy of locating the fault in the network. To aid in this process, we are developing a distributed live monitoring (DLM) tool that will identify failed power devices amongst other features.

In a distributed network, the ability to understand the structure and status of the network is imperative. 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 based on data provided to the operational model.

Utilizing a centralized server approach in our first prototype, nodes can communicate with the interface server. The server provides a graphical representation of the status of the nodes and current messages in flight. The projected design of DLM will present 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. The information is provided by working nodes that report the status of successful and failed communication attempts within the network.

This work differs from using a commodity tool such as Cacti [12] or Ganglia [13] in that it will be able to provide the visualization for non-IP devices such as those used in a ZigBee platform that are only MAC-addressable. DLM interfaces with IP-addressable nodes to detect any MAC-based devices connected to it and displays their current status.

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

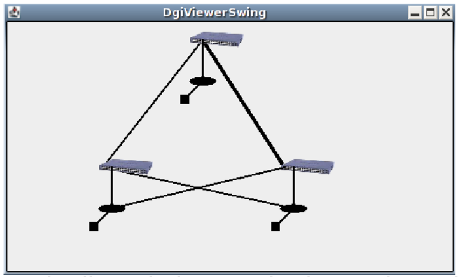


Fig. 7. Distributed Live Monitoring Swing Window

The tool is developed using the Java Swing graphics packages over a network socket API to connect with our distributed prototype written in C++ using the Mace distributed program library [11]. In its current status, it supports:

- setting status,
- defining links,
- defining partitions,
- visualizing paths and
- visualizing messages.

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 DLM. In such a case (*i.e.*, when timeouts occur in the system), the node status is updated by the node that receives the timeout.

8. Conclusion

The vision of this work is to provide fault tolerant communication for micro grids. 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 resilient by exploiting redundancy through utilization of alternate communication paths at the software level. Software re-routing thus allows cheap switching equipment without dynamic routing capabilities to be deployed instead of significantly more costly routers. 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. Overall, our software middleware architecture for fault tolerant network overlays realizes the vision of sustainable, scalable and reliable decentralized energy management on the software side in the FREEDM system and for other CPS domains.

References

[1] R. Akella, F. Meng, D. Ditch, B. McMillin, and M. Crow, "Distributed power balancing for the freedm system," in *in Proceedings of the 2010 Annual FREEDM Conference*, 2010.

[2] K. Tomsovic, D. Bakken, V. Venkatasubramanian, and A. Bose, "Designing the next generation of real-time control, communication, and computations for large power systems," *Proceedings of the IEEE*, vol. 93, no. 5, pp. 965 –979, may. 2005.

[3] "North carolina state university freedm project," http://www.freedm.ncsu.edu.

[4] "Nerc final report," http://www.nerc.com/docs/docs/blackout/ch5.pdf.

[5] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "Pipeneta wireless sensor network for pipeline monitoring," in *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2007, pp. 264–273.

[6] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "The case for resilient overlay networks," in *in Proceedings of the 8th Annual Workshop on Hot Topics in Operating Systems HotOSVIII*, 2001, pp. 152–157.

[7] J. A. Bondy, *Graph Theory With Applications*. Elsevier Science Ltd, 1976.

[8] J. Varma, C. Wang, F. Mueller, C. Engelmann, and S. L. Scott, "Scalable, fault-tolerant membership for MPI tasks on hpc systems," in *International Conference on Supercomputing*, Jun. 2006, pp. 219–228.

[9] T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauser, "Active messages: a mechanism for integrated communication and computation," in *International Symposium on Computer Architecture*, 1992, pp. 256–266.

[10] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," 2005, pp. 115–148. [Online]. Available: http://dx.doi.org/10.1007/3-540-27139-2_7

[11] C. Killian, J. Anderson, R. Braud, R. Jhala, and A. Vahdat, "Mace: language support for building distributed systems," in *ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2007, pp. 179–188.

[12] "Cacti: The complete rrdtool-based graphing solution [online]," 2005, http://www.cacti.net.

[13] M. L. Massie, B. N. Chun, and D. E. Culler, "The ganglia distributed monitoring system: Design, implementation and experience," *Parallel Computing*, vol. 30, p. 2004, 2003.