Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 9821289, 13 pages
http://dx.doi.org/10.1155/2016/9821289

*Research Article*

# $\lambda$-Augmented Tree for Robust Data Collection in Advanced Metering Infrastructure

**Joseph Kamto,[1] Lijun Qian,[1] Wei Li,[2] and Zhu Han[3]**

[1] *Department of ECE, Prairie View A&M University, Prairie View, TX 77446, USA*
[2] *Department of Computer Science, Texas Southern University, Houston, TX 77004, USA*
[3] *Department of ECE, University of Houston, Houston, TX 77004, USA*

Correspondence should be addressed to Joseph Kamto; josephkamto@gmail.com

Received 6 September 2015; Accepted 11 February 2016

Academic Editor: Pedro S. Moura



Tree multicast configuration of smart meters (SMs) can maintain the connectivity and meet the latency requirements for the Advanced Metering Infrastructure (AMI). However, such topology is extremely weak as any single failure suffices to break its connectivity. On the other hand, the impact of a SM node failure can be more or less significant: a noncut SM node will have a limited local impact compared to a cut SM node that will break the network connectivity. In this work, we design a highly connected tree with a set of backup links to minimize the weakness of tree topology of SMs. A topology repair scheme is proposed to address the impact of a SM node failure on the connectivity of the augmented tree network. It relies on a loop detection scheme to define the criticality of a SM node and specifically targets cut SM node by selecting backup parent SM to cover its children. Detailed algorithms to create such AMI tree and related theoretical and complexity analysis are provided with insightful simulation results: sufficient redundancy is provided to alleviate data loss at the cost of signaling overhead. It is however observed that biconnected tree provides the best compromise between the two entities.

## 1. Introduction

Advanced Metering Infrastructure (AMI) is a hierarchically heterogeneous communication infrastructure whose aim is to maintain the interaction between electrical power utility and power consumers. Home Area Network (HAN) is the low level of AMI whereby smart appliances (SA) connect within a short range star communication network with the smart meter (SM) that acts as the smart home gateway.

At the intermediate level, in the neighborhood area network (NAN), smart meters rely on each other to report their power demand to the utility [1]. The NAN is segmented in autonomous NAN Building Blocks (NANBBs) and smart homes connect within a tree network configuration of the SMs rooted at the dedicated router gateway (RG). The various RGs at the intermediate level connect in a full wireless mesh network and merge all the various autonomous NANBBs into an AMI Wireless Network (AWN) centered around the dedicated collector base station (CBS) at the substation (Figure 1).

In a WMN topology every node is connected to every other node in the network (Figure 2), so that a new node can be added to an active communication path to remediate another node or link failure without the need of an external action (self-healing and configuring). It is thus immune against single point of failure. Here, every node not only sends its own signals but also relays information from other nodes (for flooding or routing mechanism) which makes the topology very expensive as there are many redundant connections. Consequently, WMN is difficult to maintain and suffers from bandwidth waste as well as delay and signaling overhead due to aforementioned redundancy [2]. Almost 50% of the gross data rate has to be used for overhead to guarantee links stability and uninterrupted data transmission. In addition, another 50% of the available data rate has to be reserved for the backup path. As such, radios are sending the same data twice. Smart meters (SMs) are sensor-like devices that are limited in storage and computational capability and are not good candidates for mesh network structure compared to

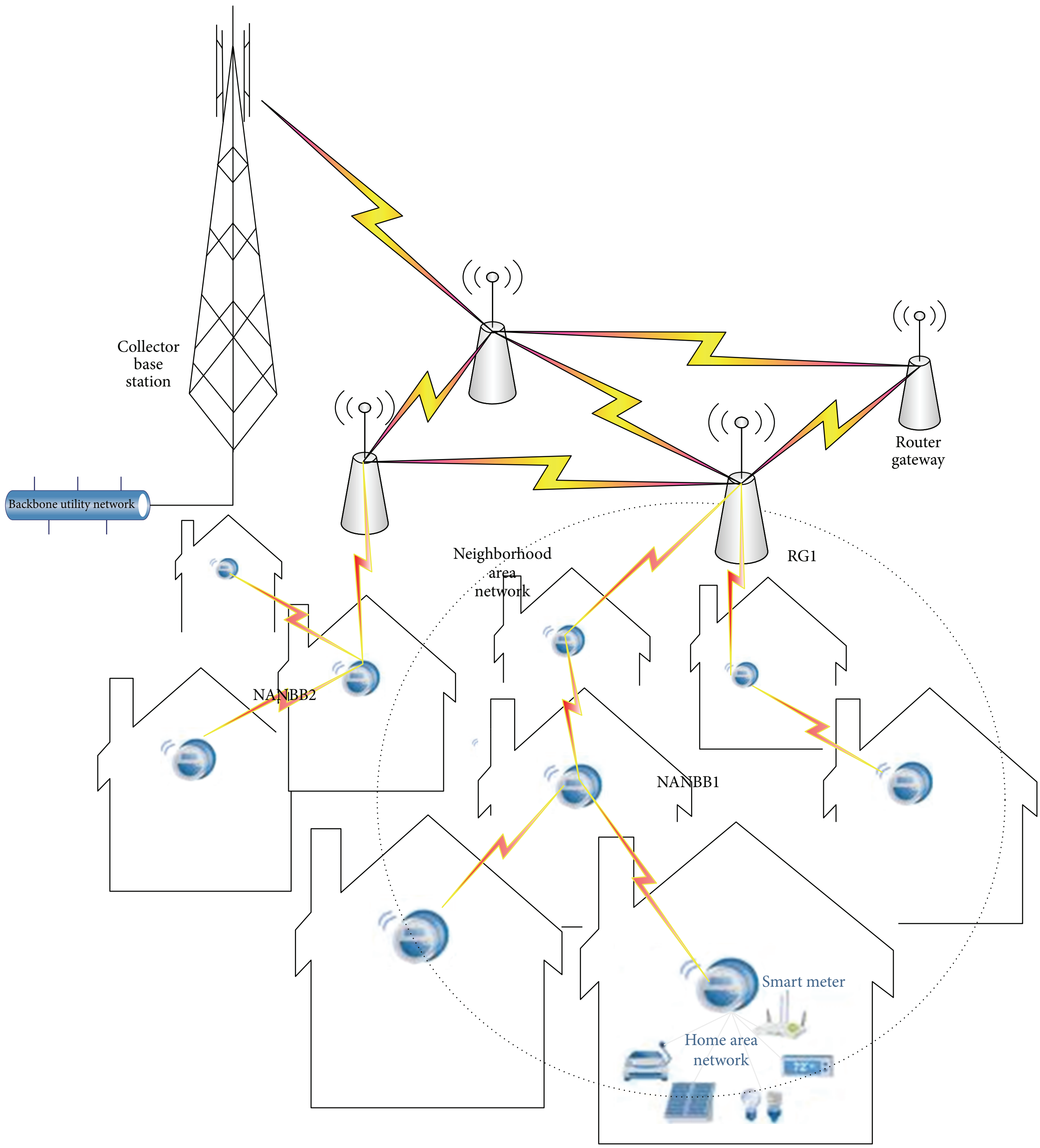


Figure 1: AMI Wireless Network (AWN).

the RGs. WMN is thus unlikely to deliver the speed necessary to fulfill the latency requirement of AMI although it can maintain the connectivity as any SM node or link failure does occur [3–5].

Tree multicast configuration of smart meters is appropriate to maintain a connected network topology and meet the latency requirement for the AMI. It is easy to set up and it is decent in terms of delay, signaling overhead, and bandwidth waste compared to its mesh counterpart. In order to further satisfy the delay constraint of AMI, hop-constrained or shallow trees may be established. However, such network topology is extremely weak as failure of any

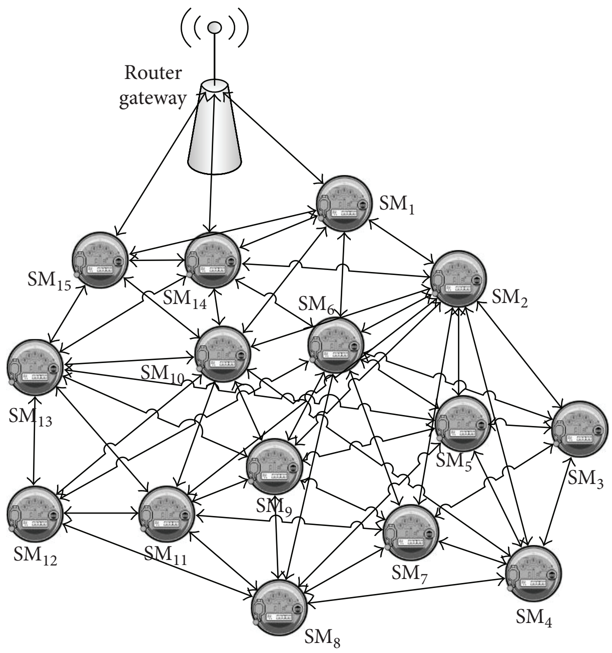


Figure 2: Mesh-based NANBB network.

single node or link could cause the network to partition into separate blocks [2], which may lead to significant loss of data.

This prompts the design of a novel network architecture that merges the advantages of both tree and mesh topology to satisfy the latency requirement while providing a minimum redundancy to ease the connectivity recovery from link failure [6–9] (e.g., Figures 4 and 5). Furthermore, node failure can still have more or less significant impact on such augmented tree connectivity depending on its location: a leaf SM or noncut SM will have a local limited impact compared to a cut SM that will partition the network into disjoint blocks such that some SMs are not able to report to the RG. Such cut SM node is considered of significant criticality. For example, in Figure 9, the network remains connected after any of the leaf smart meters such as $SM_3$ or noncut smart meter like $SM_{10}$ fails. However, the loss of any cut smart meter like $SM_1$ or $SM_{14}$ disconnects the AMI and prevents all the descendant smart meters from reporting to the utility.

Many existing research works on network link connectivity and reliability have focused on the cellular network backhaul augmentation which in the context of AMI corresponds to the data aggregator unit (i.e., Collection Base Station or CBS) interconnection. Dutta and Kubat in [10] propose the use of self-healing rings for the backbone network. In [11], Charnsripinyo and Tipper use a two-phase approach, one to determine the minimum cost network design to satisfy traffic demand and QoS and the other one to consider a problem of incremental network design for minimum cost network to augment the topology with backup link to the backhaul. On the node failure connectivity, existing research works on sensor networks take advantage of the node mobility to propose topology repair schemes in which nodes react to failure by relocating and isolating the failing node to maintain the network connectivity. In [12], Abbasi et al. rely on the view of a node in its neighborhood to design a recovery scheme that relocates the minimum number of nodes while ensuring conservation of path length between any pair of nodes. In [13, 14], predetermined criteria to involve any node in the connectivity recovery effort are maintained in a table for neighboring one- or two-hop node. All the aforementioned schemes rely on the node mobility. In AMI, SM nodes are not movable.

In this paper, we focus our work on the low level end-user's communication network, the neighborhood area network (NAN) reliability: an algorithm is designed to select a set of backup links to augment the NANBB tree based topology from 1-edge-connected tree to $\lambda$-edge-connected tree, where $\lambda > 1$ (an integer) is a design parameter. A set of disjoint paths will be created between the RG and any SM. The topology is said to be $\lambda$-augmented and will sustain the connectivity after as many as $\lambda - 1$ adjacent links fail [15, 16]. For example, if we set $\lambda = 2$, the augmented tree would have a biconnected topology. As a result, there will exist at least two disjoint paths in any cut that separates the RG from any SM nodes as noticeable in Figure 4; likewise, triconnected tree network corresponds to $\lambda = 3$ such that there are no two adjacent communication links whose disability can break the NANBB as can be seen in Figure 5. This scheme has the advantage of significantly reducing the signaling overhead and the radio bandwidth waste compared to mesh networks. However, our proposed scheme lacks in root redundancy so that any RG's defection will disconnect the corresponding AMI building block. Since RGs are well maintained by utility companies, we assume they are free of failures.

In the AMI, SM nodes are not mobile and cannot relocate as proposed in previous works. To address the impact of a SM node failure on the connectivity of the augmented tree network, a proactive topology repair scheme is proposed. It relies on a loop detection scheme to define the criticality of a SM node and specifically targets the cut SM nodes by selecting a set of backup parents to all the immediate children SM nodes. These links come into action immediately after the cut SM fails such that the AMI network mutates and recovers the connectivity by isolating the failing critical SM node.

The rest of this paper is organized as follows. In Section 2, motivating examples are provided to illustrate the minimum redundancy and showcase the ease of connectivity recovery in biconnected and triconnected topologies from link failure. Section 3 studies the basic case of biconnected tree topology for NANBB. A depth first search based algorithm is proposed to solve the biconnected tree augmentation. Section 4 enumerates major change to the DFS algorithm to accommodate the $\lambda$-connected tree algorithm to suit higher level of connectivity. Section 5 presents the topology repair scheme to address the impact of the node failure and connectivity recovery in augmented tree network. In Section 6, a study is proposed to evaluate the complexity of the algorithm in terms of the size of the network and the target connectivity. In Section 7, IEEE 802.11 MAC handshake protocol is employed as basis to simulate and assess the effectiveness of the augmented tree topology in terms of signaling overhead and data loss. Section 8 summarizes the work.

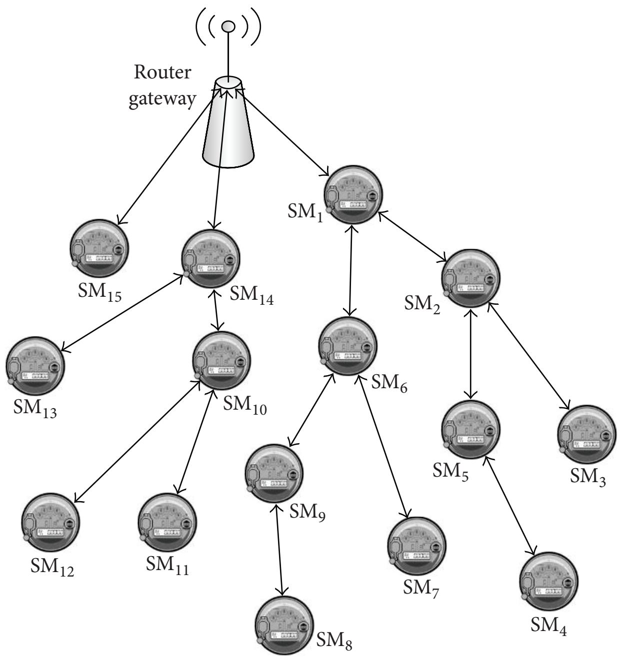


FIGURE 3: Tree based NANBB network (degree 3).

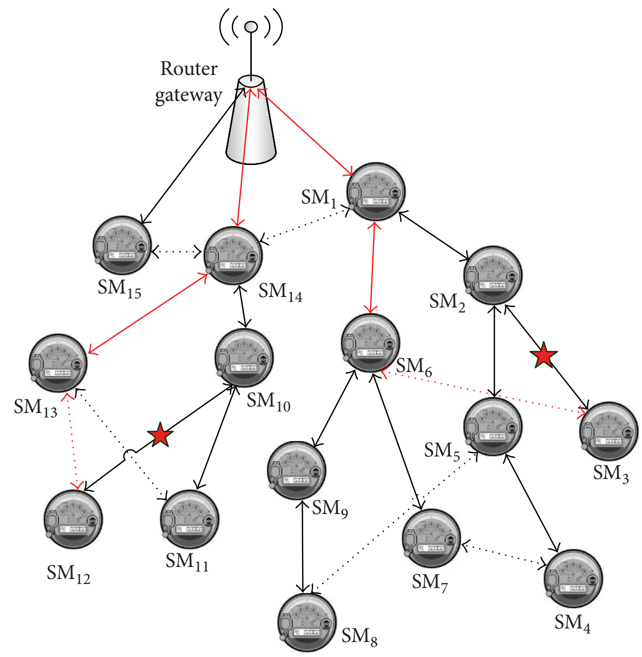


FIGURE 4: Double link failures recovery in a biconnected NANBB.

## 2. Link Failure Connectivity Recovery: Motivating Examples

In this section, we focus on studying two basic augmented NANBB tree based topologies. Specifically we describe and outline the reliability improvement of augmented tree based NANBB for $\lambda = 2$ (biconnected) and $\lambda = 3$ (triconnected).

Connectivity is a predominant concept in a communication network; it refers to the network ability to sustain its full operation despite link failure [17]. For a connected NANBB network, its spanning tree is a configuration through which all the SMs are connected with $\lambda = 1$ (Figure 3).

The goal of this work is to mitigate the weakness of the NANBB spanning tree based communication topology by assigning backup links to thwart against single point of failure and increase its reliability [18]. This may be measured by the probability that the network remains connected despite link failure; that is, any SM somehow remains capable of transmitting its power demand to the RG under link failure, provided that other SMs are operative.

Consider the two augmented tree based NANBBs: biconnected (Figure 4) and triconnected (Figure 5). In the biconnected tree (Figure 4), each SM maintains at least two disjoint paths to the RG. Upon failure of any single communication link (red star), the adjacent backup communication links take over to maintain the connectivity of the corresponding SMs ($SM_3, SM_{12}$) to the dedicated RG through the disjoint path (red dotted links): no single link failure can break the connectivity as it is systematically the case for the spanning tree topology. In general, a biconnected tree can recover from any single communication link failure. In addition, it may recover from more than a single nonadjacent links failure. Examples of recovery by redirecting the traffic through backup links are shown for a double nonadjacent links failure (double red stars) in Figure 4.

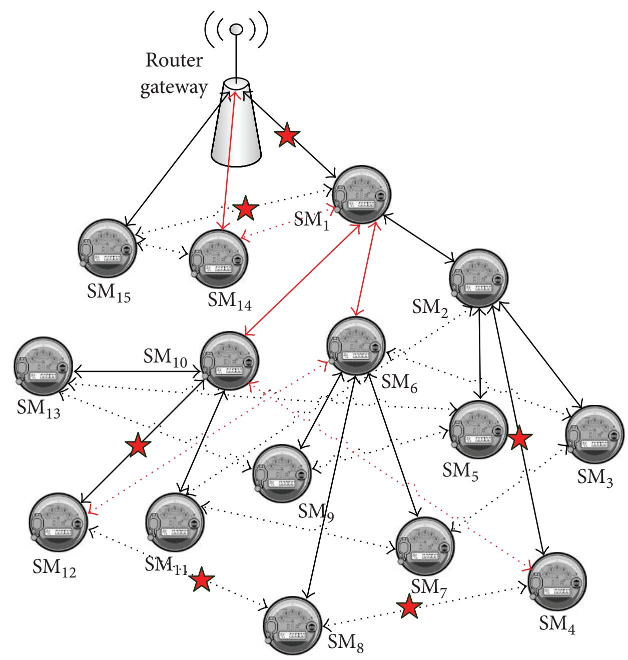


FIGURE 5: Triple link failures recovery in a triconnected NANBB.

Figure 5 shows three-failure recovery for $SM_1$, $SM_4$, and $SM_{12}$ in a triconnected NANBB. Here, there exist at least up to three disjoint paths in between any single SM and the root RG so that two adjacent links failures cannot disconnect any SM from the root RG as it is the case in the biconnected NANBB (Figure 4). In triconnected NANBB (Figure 5), $SM_1$, $SM_4$, and $SM_{12}$ remain connected through the third disjoint communication paths (red link) as both other links fail simultaneously. In the context of this paper, the challenge is to find the minimum $\lambda$-connected tree based topology that improves the reliability of the NANBB communication network by randomly or deterministically assigning backup parent SM.

## 3. Link Failure Connectivity Recovery: Biconnected Tree

In this paper, we make the following assumptions: (1) only one-node SM or communication link failure occurs at a time in the network, the rationale being that multiple simultaneous failures are less likely; (2) a SM node would transmit at its maximum power to increase the chance of being in range with multiple neighbors; (3) the wireless nature of the medium prompts considering link cost generally specified in terms of delay, packet loss rate, bit error rate, guaranteed bit rate, and so on. On the other hand, undesirable connectivity augmentation due to RF signal propagation is worth investigating; communication security scheme involving key distribution and management is a good candidate solution [2]: only SM nodes sharing a secret key can communicate such that radio signals from other nodes in range are simply ignored.

*3.1. Problem Formulation and Definitions.* A NANBB communication network can be represented by a graph $X = G(V, E)$ (Figure 2), where $V$ is the set of smart meters ($\text{SM}_i$) and $E$ is the set of communication links ($l_{ij} = [\text{SM}_i, \text{SM}_j]$) between pair of smart meters. Firstly, depth first search (DFS) algorithm (Section 3.2.1) can be run on the graph $G(V, E)$ to generate a tree $T(V, \Gamma)$ (Figure 3) that spans all the SMs in the NANBB network.

The router gateway (RG) is the root to which any SM directs its data. Given a smart meter $\text{SM}_i$, the path to the root is denoted by $p(\text{RG}, \text{SM}_i)$ and the connectivity of $\text{SM}_i$ is defined as the maximum number of disjoint paths to the RG [19]. Thus, the connectivity is 1 for a NANBB tree topology since no common internal edge exists between the distinct paths.

The tree based NANBB connection reliability problem can be formulated as identifying the minimum set of links to be considered as candidate backup link to a backup parent SM. In other words, the goal is to find additional backup SM through which each individual SM can rely on for connecting to the tree root RG as its original parent SM is out of reach due to link failure. At the detection of the failure, it could take step to activate the backup link and remain connected [15].

The connectivity augmentation in a NANBB tree network $T(V, \Gamma)$ can be defined such that, given a function, $\lambda : V \times V \to N, (\text{RG}, \text{SM}_i) \to \lambda(\text{RG}, \text{SM}_i)$; find the set $K$ of edges in $E - \Gamma$ such as in $\epsilon = \Gamma + K$: $\aleph_{p(\text{RG},\text{SM}_i)} \geq \lambda(\text{RG}, \text{SM}_i), \forall i$, where $\aleph_{p(\text{RG},\text{SM}_i)}$ is the number of disjoint paths. In the case of uniform requirement where $\lambda(\text{RG}, \text{SM}_i) = \lambda$ for all SMs, it simply becomes a problem of $\lambda$-edge-connectivity.

Let $N_T(\text{SM}_i)$ denote the set of neighbors of $\text{SM}_i$ in the NANBB tree network ($T$). The path $p(\text{SM}_i, \text{SM}_j)$ between $\text{SM}_i$ and any $\text{SM}_j \in N_T(\text{SM}_i)$ is simply the edge $l_{ij} = [\text{SM}_i, \text{SM}_j]$. The degree of ($\text{SM}_i$) $\deg_{\text{SM}_i}$ is the size of the neighbor set $N_T(\text{SM}_i)$ (i.e., $\deg_{\text{SM}_i} = |N_T(\text{SM}_i)|$) [18].

The following definitions for NANBB tree network topology are introduced:

(i) A leaf smart meter denoted as (lSM) has no descendant SMs and $L$(lSM) denotes the set of lSMs.

(ii) A cut smart meter (cSM) disconnects the network in as many parts as its degree when deleted.

(iii) A stem smart meter (sSM) has only lSMs as descendant.

From the motivating example described above (Section 2) the following significant observations are made:

(i) Leaf smart meters (lSMs) are exclusively subject to additional backup parent SM for connectivity augmentation.

(ii) Connectivity augmentation requires at least $\deg_{\text{sSM}} - 1$ additional edges per stem smart meter (sSM).

The following categories [15] describe the set from which a backup SM derives:

(i) Uncle: the set of the backup SMs contains the sibling of the parent SM. The root RG has no brother and, as such, its children SMs have no uncle and thus no backup SM either.

(ii) Cousin: this set of backup SMs is made of the uncle's children SMs. Here also, the root has no brother. This makes it impossible to the root's children to have backup SM as well.

*3.2. NANBB Biconnected Tree Link Augmentation.* In this section, we present an algorithm for solving communication link augmentation based on the observation in Section 2 and algorithms in [20]. Prior to that, we describe the depth first search (DFS) algorithm to derive the NANBB tree topology. Breath First Search (BFS) is another search scheme where each node is visited and numbered with access to all of its neighbors at the same time; the neighbors are likewise visited next all together in turn, and so on [21, 22]. In both cases, the root RG is first located where the search starts. They all have similar performance cost in terms of time and space. However, the nodes order produced by the DFS is more appropriate for our scheme in Section 3.2.2 that we survey in [20].

*3.2.1. Depth First Search (DFS) Algorithm .* The root is chosen and its immediate rightmost child SM is explored; the search progresses recursively until reaching a set goal or hitting the last rightmost SM node. From there, the search backtracks and resumes at the last rightmost descendant SM not completely explored [23]. Here each SM is required to have at most $\lambda$ children (i.e., $\deg_{\text{SM}i} = |N_x(\text{SM}_i)| \leqslant \lambda + 1$ (Figure 3) for $\lambda = 2$, $\deg_{\text{SM}_i} \leqslant 3$).

*Data Structures*

(i) lSM: leaf SM.

(ii) cSM: cut SM.

(iii) $T_{\text{cSM}_i}$: subtree rooted at $\text{cSM}_i$.

(iv) sSM: stem SM.

(v) $N_X(\text{SM}_i)$: neighbor SM set of $\text{SM}_i$ in the graph $X$ [18].

(vi) $p(\text{SM}_i, \text{SM}_{i+1})$: communication path between two SMs.

(vii) $\aleph_{p(\text{SM}_i,\text{SM}_{i+1})}$: number of disjoint paths between them.

(viii) *State*: a SM is sequentially *unvisited*, *discovered*, *visited*, and *finished* when the DFS withdraws.

An appropriate description of a DFS for a NANBB network is in terms of the spanning tree of SMs as the search ends. Its mechanism in a NANBB perspective is based on setting and getting the labels of SMs and communication links of the spanning tree.

*3.2.2. Algorithm: NANBB Biconnected Tree Network.* Here, the problem is to find the smallest set of links which in addition to the NANBB spanning tree communication network makes it biconnected so as no single link failure can disconnect any single SM (Figure 4).

Initially the algorithm calls the NANBB network and identifies its root which in this case is the data aggregator unit RG. The NANBB tree network topology is issued resulting from the DFS (Section 3.2.1). The NANBB network is a simple tree that spans all the SMs. A single communication link failure suffices to break the network connectivity, whereas a single SM failure results in the segmentation of the network in a number equal to its degree, whichever case prevents a set of SMs from communicating with the root RG (Figures 3 and 7).

Consider $q[L(\text{lSM})]$ the partitioning of the set $L(\text{lSM})$ of leaf smart meters (lSMs) of the NANBB tree network $T$

$$q[L(\text{lSM})] \subseteq \left\{ L(\text{cSM}_i) \mid \forall \text{lSM}_i \in L(\text{cSM}_i), \exists! \text{cSM}_i \in p(\text{lSM}_i, \text{lSM}_{i+1}), \deg_{\text{cSM}_i} = 3 \right\}. \quad (1)$$

For example, in the NANBB tree (Figure 3), the set of leaves $L(\text{lSM})$ comprises the SMs, $\text{SM}_{15}$, $\text{SM}_{13}$, $\text{SM}_{12}$, $\text{SM}_{11}$, $\text{SM}_8$, $\text{SM}_7$, $\text{SM}_4$, and $\text{SM}_3$ with the partition set $q[L(\text{lSM})]$ below. Each partition set $L(\text{lSM})$ can be augmented to become biconnected (i.e., subset $L(\text{cSM}_2) = \{\text{SM}_3, \text{SM}_4\}$ in Figure 4 is augmented with 3 incident edges: $[\text{SM}_3, \text{SM}_6]$, $[\text{SM}_4, \text{SM}_7]$, and $[\text{SM}_5, \text{SM}_8]$).

Upon augmentation of the first partition $L(\text{cSM}_2)$, the network becomes a biconnected component tree where each node is either a SM or a biconnected component (e.g., biconnected subtree $T_{\text{cSM}_1}$ rooted at $\text{SM}_1$ in Figure 4). The biconnected component is thereafter contracted into its root ($\text{SM}_1$) that will be considered a new leaf SM in the next partitioning followed by a new augmentation

$$q[L(\text{lSM})] \subseteq \{L(\text{RG}) = \{\text{SM}_{15}\}, L(\text{cSM}_{14}) = \{\text{SM}_{13}\}, L(\text{cSM}_{10}) = \{\text{SM}_{12}, \text{SM}_{11}\}, L(\text{cSM}_6) = \{\text{SM}_8, \text{SM}_7\}, L(\text{cSM}_2) = \{\text{SM}_4, \text{SM}_3\}\}. \quad (2)$$

The recursive process proceeds following a rightmost DFS technique until the entire network reduces to a single SM (i.e., RG).

**Input**: NANBB Mesh Network $G = (V, E)$
**Output**: NANBB Tree Network $T(V, \Gamma)$
(1) $\Gamma = \{\}$
(2) **for** $(\text{SM}_i, l_{ij}) \in G(V, E)$ **do**
(3) Mark label $\text{SM}_i$: *unvisited*;
(4) Mark label $l_{ij}$: *unvisited*;
(5) Select RG, DFS's root.
(6) Mark label RG: *visited*
(7) **DFS** $(G, \text{RG})$
(8) **for** $\text{SM}_i \in N_G(\text{RG})$ **do**
(9) Select $\text{SM}_i$ *rightmost* unvisited *descendant*
(10) **if** *Get label* $(\text{SM}_i) =$ *unvisited* **then**
(11) $\text{SM}_i \leftarrow \text{opposite}(\text{RG}, l_{ir})$
(12) Mark label $(l_{ir})$: *discovered*
(13) Add $l_{ir}$ to $\Gamma_i$
(14) Mark label $l_{ir}$: *visited*
(15) Mark label $\text{SM}_i$: *visited*
(16) Add $l_{ir}$ to $\Gamma$
(17) **else**
(18) Set the label of $(l_{ir})$ as: *back-edge*.
(19) **if** $|\Gamma_i| \geqslant 2$ **then**
(20) **DFS** $(G, \text{SM}_i)$
(21) **return** $T(V, \Gamma)$

Algorithm 1: DFS (depth first search).

## 4. Link Failure Connectivity Recovery: $\lambda$-Augmented Tree

In this section, we study the NANBB tree $\lambda$-augmentation based on the requirement that the source NANBB network is a full mesh of at least degree $\lambda$ (i.e., each node including the RG in the NANBB mesh has at least $\lambda$ disjoint communication paths to any other node) and we enumerate the different changes to enhance Algorithms 1 and 2.

*4.1. DFS Algorithm.* DFS Algorithm 1 has to be modified to output a NANBB tree of degree at most $\lambda + 1$.

Specifically, in its line (19), DFS Algorithm 1 has to be changed into **If** $|\Gamma_i| \geqslant \lambda$ **then**.

Figure 7 shows the example DFS output for $\lambda = 3$; here each node is at most degree 4 with at most 3 descendants.

*4.2. NANBB $\lambda$-Connected Tree Communication Network.* Algorithm 2 is upgraded into Algorithm 3 to accommodate higher degree tree topology (i.e., Figure 7 for $\lambda = 3$) and output higher $\lambda$-connected NANBB tree network as desired (Figure 8 for $\lambda = 3$).

The following changes can be seen in Algorithm 3 NANBB $\lambda$-connected tree network:

(1) In line (7), the partition of leaf SM is done with consideration of $\lambda$ for the degree of cSM:

$$q[L(\text{lSM})] \subseteq \left\{ L(\text{cSM}_i) \mid \forall \text{lSM}_i \in L(\text{cSM}_i), \exists! \text{cSM}_i \in p(\text{lSM}_i, \text{lSM}_{i+1}), \deg_{\text{cSM}_i} = \lambda \right\}. \quad (3)$$

**Input**: NANBB Mesh Network $G = (V, E)$
**Output**: NANBB Biconnected Tree $H = T(V, \epsilon)$
(1) $T(V, \Gamma) = \text{DFS}(\text{RG}, V)$ (*RG is the root of Tree $T(V, \Gamma)$*)
(2) $\epsilon = \Gamma$
(3) $H = T(V, \epsilon)$
(4) Mark $l_{ij} \in E - \Gamma$: *unvisited*
(5) Mark $\text{SM}_i \in V - L(\text{lSM})$: *unvisited*
(6) **while** $|L(\text{lSM})| > 2$ **do**
(7) Compute $q[L(\text{lSM})]$ (*partition of $L(\text{lSM})$*)
(8) Select rightmost $L(\text{lSM})_i \in q[L(\text{lSM})]$
(9) **for** $\text{lSM}_i \in L(\text{lSM})_i$ **do**
(10) Compute $L_i = \{l_{ij} \in E - \Gamma, l_{ij}$ incident to $\text{lSM}_i\}$
(11) **if** $|L_i| = 1$ **then**
(12) Add $l_{ij}$ to $\epsilon$
(13) Mark $l_{ij}$: *visited*
(14) Mark $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij})$: *visited*
(15) **else**
(16) **for** $l_{ij} \in L_i$ **do**
(17) Compute $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij})$
(18) **if** $\text{SM}_j$: *unvisited* **then**
(19) Compute weight of $l_{ij}$
(20) Mark $l_{ij}$: *visited*
(21) Add least weighted $l_{ij}$ to $\epsilon$
(22) Mark $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij})$: *finished*
(23) Identify biconnected subtree $T_{\text{cSM}_i}$, (*$\text{cSM}_i$, subtreeroot*)
(24) Contract $T_{\text{cSM}_i}$ into its root $\text{cSM}_i$
(25) Update $H = T(V, \epsilon)$
(26) **else**
(27) Add edge $l_{ij} = [\text{lSM}_i, \text{lSM}_j]$ to $\epsilon$
(28) Mark $l_{ij}$: *visited*
(29) Update $H = T(V, \epsilon)$
(30) **return** $H = T(V, \epsilon)$

Algorithm 2: AMI biconnected tree communication network.

(2) Lines (11) and (12) are added to identify the incident link in the spanning tree.

(3) Line (13) is added to ensure up to $\lambda - 1$ links augmentation per leaf SM.

(4) Lines (28) and (29) are used to create and identify the rightmost $\lambda$-connected subtree.

## 5. SM Node Failure Connectivity Recovery

The scheme developed above can certainly generate multiple disjoint paths to mitigate the weakness of the tree topology and improve the robustness of AMI. However, these paths are seldomly node disjoint as they cross over at numerous SM nodes whose failure can more or less significantly affect the connectivity of such augmented tree network: a leaf SM or noncut SM will have a local limited impact compared to a cut SM that will partition the network into disjoint blocks such that some SMs are not able to report to the RG. Such cut SM nodes are critical for the network connectivity. For example, in Figure 9, the network remains connected after any of the leaf smart meters such as $\text{SM}_3$ or noncut smart meter like $\text{SM}_{10}$ fails. However, the loss of any cut smart meter (critical) like $\text{SM}_1$ or $\text{SM}_{14}$ disconnects the AMI and prevents all the descendant smart meters from reporting to the utility.

In this section, we describe a proactive topology repair scheme to reconfigure the augmented tree network in response to SM node failure. It mostly targets the critical nodes as previously mentioned: a loop detection mechanism is used to define a critical SM node (e.g., $\text{SM}_1$ in Figure 9) as a choking point in the path to the RG and select supplemental SMs as backup parents to its children SMs ($\text{SM}_2$ and $\text{SM}_6$ in Figure 9).

Each SM node in addition to the *routing* table maintains a *loop detection* table to record the packets forwarding activities. The node can use the packet identifier to detect a returning packet and conclude on the existence of a loop.

*5.1. Loop Detection Mechanism.* The loop detection mechanism as depicted in Figure 9 mimics the depth first search algorithm. In this mechanism, the purpose is not to route the packet to the best candidate next hop in the routing table, but instead to check each next hop to determine if it belongs to a loop.

When a node creates a packet that it wants to use for loop detection, it appends its address to a sequence number to

**Input**: AMI Mesh Network $G = (V, E)$
**Output**: AMI $\lambda$-Connected Tree Netw. $H = T(V, \epsilon)$
$T(V, \Gamma) = \text{DFS}(G, \text{RG})$ (*RG root of Tree $T(V, \Gamma)$*)
$\epsilon = \Gamma$
$H = T(V, \epsilon)$
Mark $l_i \in E - \Gamma$: *unvisited*
Mark $\text{SM}_i \in V - L(\text{lSM})$: *unvisited*
**while** $|L(\text{lSM})| > \lambda$ **do**
  Compute $q[L(\text{lSM})] \subseteq \{L(\text{cSM}_i) \mid \forall \text{lSM}_i \in L(\text{cSM}_i), \exists p(\text{lSM}_i, \text{cSM}_i), \deg_{\text{cSM}_i} = \lambda + 1\}$ (*partition of $L(\text{lSM})$*)
  Select rightmost $L(\text{cSM}_i) \in q[L(\text{lSM})]$
  **for** $\text{lSM}_i \in L(\text{cSM}_i)$ **do**
    Compute $L_i = \{l_{ij} \in E - \epsilon, l_{ij} \text{ incident to } \text{lSM}_i\}$
    $\epsilon_i = \{l\}, l \in \Gamma$, incident to $\text{lSM}_i$, $|\epsilon_i| = 1$
    **while** $|\epsilon_i| < \lambda$ **do**
      **if** $|L_i| = \lambda - 1$ **then**
        Add $L_i$ to $\epsilon_i$
        Mark $l_{ij} \in L_i$: *visited*
        Mark $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij} \in L_i)$: *visited*
      **else**
        **for** $l_{ij} \in L_i$ **do**
          Compute $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij})$
          **if** $\text{SM}_j$: *unvisited* **then**
            Compute weight $l_{ij}$
            Mark $l_{ij}$: *visited*
        Add the least weighted $l_{ij}$ to $\epsilon_i$
        Mark $\text{SM}_j \leftarrow$ opposite $(\text{lSM}_i, l_{ij})$: *visited*
    Add $\epsilon_i$ to $\epsilon$
    Identify $\lambda$-connected subtree $T_{\text{cSM}_i}$, (*$\text{cSM}_i$, subtree root*)
    Contract $T_{\text{cSM}_i}$ into its root $\text{cSM}_i$
    Update $H = T(V, \epsilon)$
**else**
  Add edge $l_{ij} = [\text{lSM}_i, \text{lSM}_j]$ to $\epsilon$
  Mark $l_{ij}$: *visited*
  Update $H = T(V, \epsilon)$
**return** $H = T(V, \epsilon)$

ALGORITHM 3: AMI $\lambda$-connected tree network.

create an identifier that it stores in its loop detection table. The identifier is unique and can be used to trace the packet in the entire network. The address of the next node is also stored in the table. However, the destination field of such packet is always loaded with the router gateway (RG) address.

The loop detection is a per node scheme (i.e., each node initiates the detection and determines if it is a choking point). After a node has created the packet as previously mentioned, it forwards it to its rightmost child in the augmented tree NANBB network, where the packet follows a normal forwarding mechanism per the underlying routing protocol. Hence, it is assumed that there exists a control plane that continuously finds and fills up the *routing* table at each SM node with the best paths to the RG. Any time the initiating node receives a packet to forward, it checks in its loop table to match a preexisting identical identifier. If it is a match, it concludes this is a returning packet, and it confirms a loop.

A packet loss can occur during the loop detection scheme in which case the acknowledgment message is not received back at the forwarding node. After a specified waiting time, the node proceeds to reroute the packet by forwarding it to the next child in the augmented tree.

The illustration of loop detection for node $\text{SM}_1$ is shown in Figure 9 and described below:

(i) Node $\text{SM}_1$ creates a packet designated to RG. The packet whose identification is stored in the loop table is forwarded to its rightmost child $\text{SM}_2$.

(ii) Node $\text{SM}_2$ upon reception notices the source address which prevents sending the packet back to $\text{SM}_1$ but an acknowledgment message. It then selects the best next path in its routing table and forwards the packet to node $\text{SM}_3$.

(iii) Node $\text{SM}_2$ fails to receive acknowledgment message after a specified waiting time due to link failure $l_{12} = [\text{SM}_1, \text{SM}_2]$ and chose to reroute the packet. It chose to forward the packet to node $\text{SM}_5$.

(iv) Node $\text{SM}_5$ cannot send the packet back to $\text{SM}_3$ but an acknowledgment message. The packet is thereafter sent to node $SM_7$ from where it naturally flows back through $SM_6$ and $SM_1$ toward RG.

(v) Node $SM_1$ has initiated the loop detection. At the reception of a packet it checks its loop table and detects a loop as it notices a returning packet. It takes action to notify its status as critical to its children SMs.

(vi) All the one-hop SM nodes in the neighborhood update their *loop* table to include the information on the criticality of node $SM_1$.

*5.2. SM Node Failure Detection.* SM nodes periodically broadcast heartbeat messages to neighbors to notify that they are up and operational and as well report changes to their one-hop neighbors. By means of missing heartbeat messages, active SM nodes can conclude the failure of a neighbor.

At the detection of such failure, the one-hop neighbors of the failing SM node would determine the impact, that is, whether the failing node is critical to network connectivity. They will do so by means of lookup in their *loop* table to see if the failing SM node is a choking point as previously described. The children of such critical SM node will react to such failure by redirecting their communication to the backup SM parent node, thus isolating the failing cut SM node (like $SM_1$ in Figure 10).

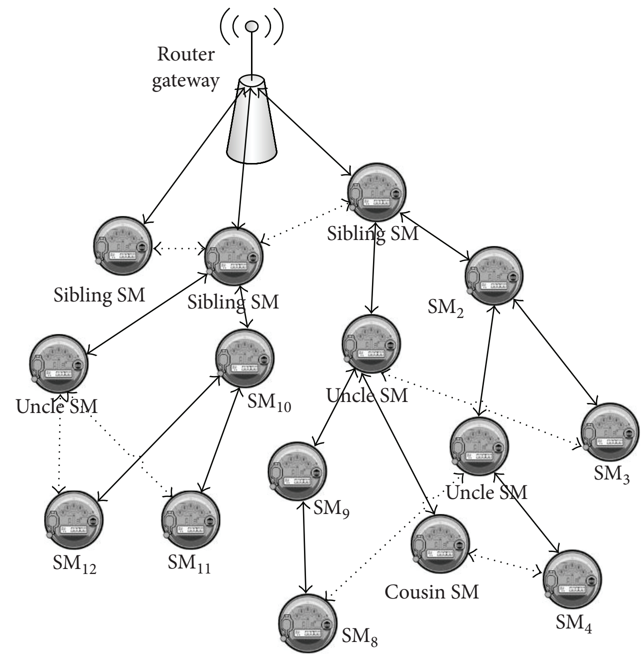


Figure 6: Backup SM category in a biconnected tree NANBB.

## 6. Algorithm Computational Complexity

The above scheme follows the procedure of the depth first search (DFS) algorithm which is a search that sequentially explores the tree network in a descendant fashion. That is, each SM is visited by the algorithm and at each step all the incident communication links to a given cut smart meter (cSM) are explored to determine if they connect to a leaf smart meter (lSM) in which case $\lambda - 1$ back edges are selected to cover the corresponding lSM and create a local $\lambda$-connected subtree. The following is the inventory of operations in a tree with $n$ nodes:

(i) $n$ nodes are visited.

(ii) $n - 1$ links in the tree are visited.

(iii) $\lambda - 1$ links are visited and chosen to cover each lSM.

Consider $|\text{lSM}| = k$, size of the set of leaf SM in the spanning tree. The visitation cost for Algorithm 3 (the $\lambda$-connected tree) can be evaluated as $[n + n + 1 + k * (\lambda - 1)]$. On the other hand, there are $n * (n - 1)$ communication links in the full mesh network with population $n$ nodes, and they are all together visited during the DFS such that the visitation cost due to Algorithm 1 is $[n + n * (n - 1)]$.

The total combined visitation for both Algorithms 1 and 3 will be $[n + n + 1 + k * (\lambda - 1)] + [n + n * (n - 1)] = [n^2 + 2 * n + 1 + k * (\lambda - 1)]$ and the computational complexity is of order $n^2$ (i.e., $\bigcirc(n^2)$).

## 7. Simulation Results

In this section, we assess the performance of the proposed scheme. The analysis is twofold. Firstly, we use the IEEE 802.11 MAC handshake protocol to compare the efficiency of mesh, tree, and augmented tree NANBB network configuration in terms of signaling overhead. We also use Matlab simulation to provide a comparative representation of average data loss due to communication link failure in those NANBB network configurations.

Assuming the handshake request as well as reply packets of size 20 bytes each and also given a 36% successful rate for the aloha handshake protocol, the signaling overhead for a successful handshake can be estimated to $\varphi = 120$ bytes per link [24].

The NANBB tree communication network augmentation is mostly about covering each and every leaf smart meter with up to $(\lambda - 1)$ backup edges (Figures 6 and 8). On the other hand, DFS algorithm is deigned to issue a NANBB tree of degree at least $\lambda$ depending on the level of the desired connectivity $\lambda$ (Figures 3 and 7). It can also be seen that the number of leaf smart meters in a full degree $\lambda$ NANBB tree is estimated to $N = \lambda^{(\lambda+1)}$ such that the augmentation process induces additional $(\lambda - 1) * \lambda^{(\lambda+1)}$ links. Considering the $(n - 1)$ links in the original NANBB tree with $(n)$ SMs, the total link maintained in the $\lambda$-augmented NANBB tree is estimated to $\tau = (n - 1) + (\lambda - 1) * \lambda^{(\lambda+1)}$ such that given a bidirectional communication link the overall handshake signaling overhead can be expressed as follows:

$$\begin{aligned}\text{Overhead}_{\lambda\text{-Aug}} &= \varphi * \left[(n-1) + (\lambda-1) * \lambda^{(\lambda+1)}\right] \\ &= 120 * \left[(n-1) + (\lambda-1) * \lambda^{(\lambda+1)}\right].\end{aligned} \tag{4}$$

In a full mesh network with $n$ nodes, there are $n(n - 1)/2$ bidirectional communication links in between all the active SMs; also each SM individually communicates with the NAN

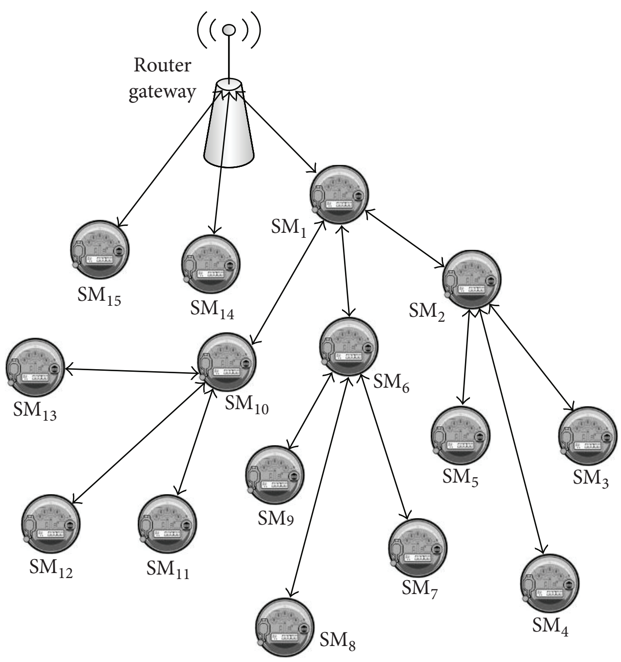


Figure 7: Tree based NANBB network (degree 4).

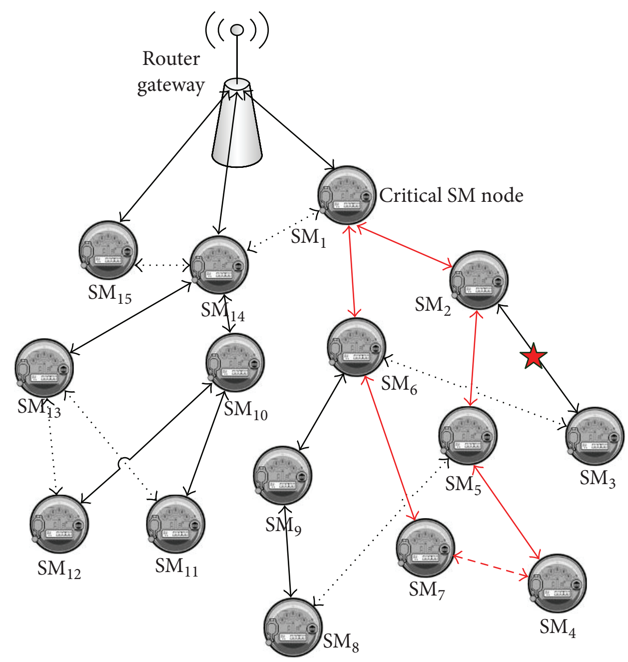


Figure 9: Critical node, loop detection in biconnected tree NANBB.

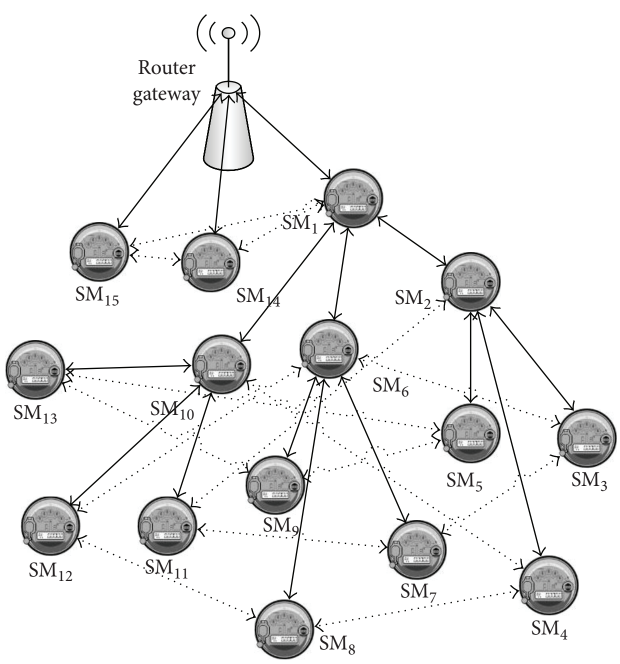


Figure 8: Triconnected tree NANBB.

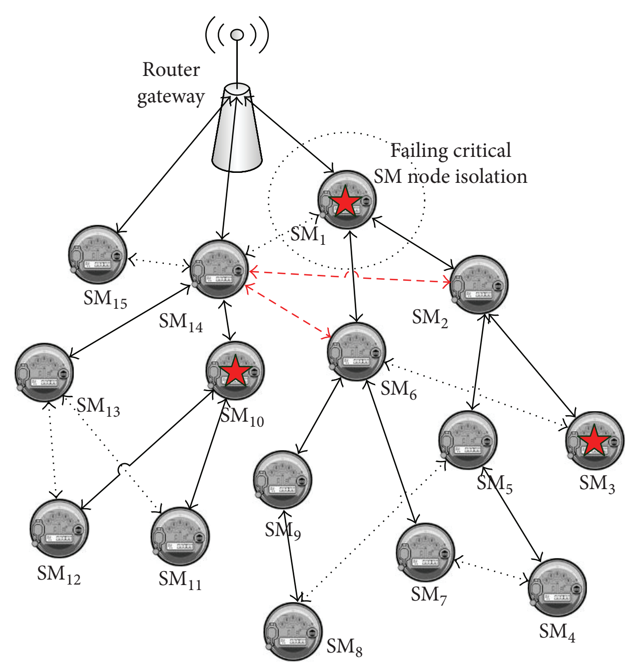


Figure 10: Critical node, failure recovery in biconnected tree NANBB.

gateway data aggregator RG so the summary of the handshake signaling can be expressed as

$$\text{Overhead}_{\text{mesh}} = \frac{\varphi * n(n-1)}{2} + \chi * n, \qquad (5)$$

where $\varphi$ is the SM-to-SM link signaling and $\chi$ is the signaling between the gateway data aggregator and any smart meter. Consider $\varphi = \chi = 120$ bytes.

Figures 11, 12, and 13 show signaling overhead comparison with regard to different size of NANBB and connectivity level. We have the following observations:

(i) Regardless of the size of the NANBB network, the signaling overhead difference between the tree and the biconnected tree is insignificant; in fact it is less than 5%.

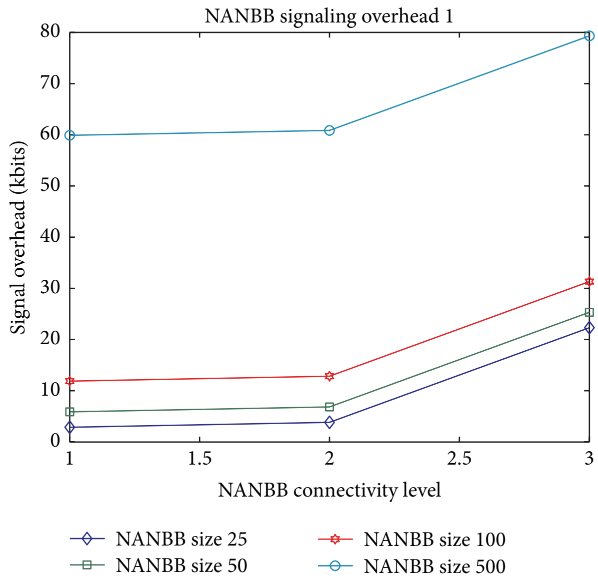


Figure 11: Signaling overhead comparison for $\lambda \leq 3$.

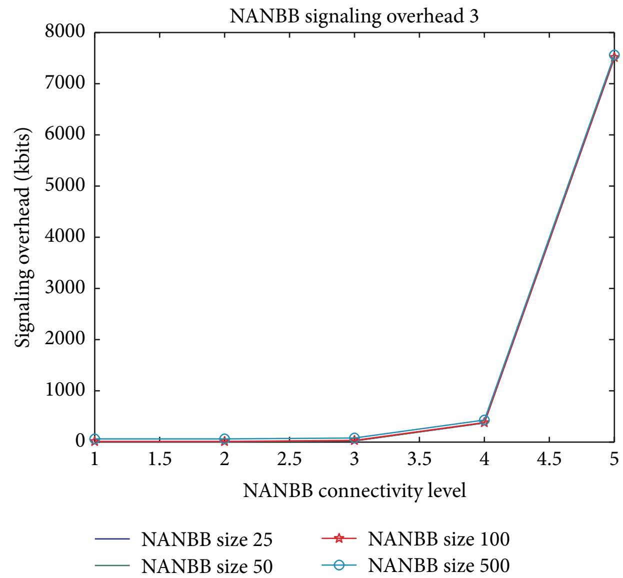


Figure 13: Signaling overhead comparison for $\lambda \leq 5$.

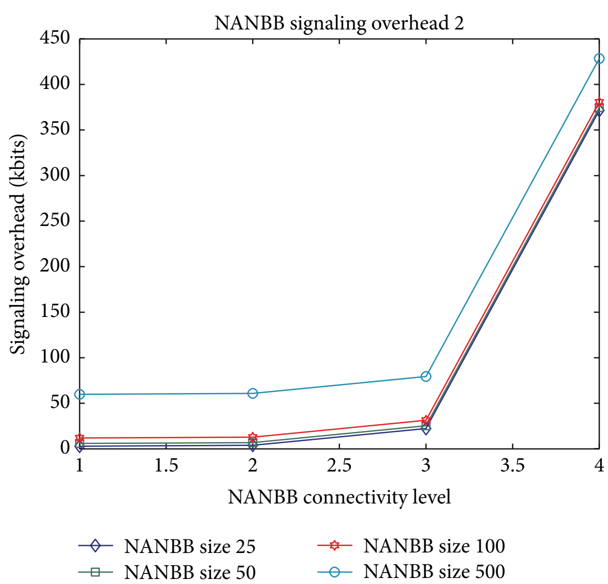


Figure 12: Signaling overhead comparison for $\lambda \leq 4$.

(ii) The signaling overhead increases reasonably for connectivity up to 3 and as connectivity approaches 4 it increases significantly to almost 80% for $\lambda = 5$ in Figure 13.

The simulation result in Figure 14 clearly shows the extremely poor performance of the full mesh NANBB when it comes to signaling overhead. In fact, the overhead increases exponentially with the number of smart meters.

On the other hand, the NANBB tree topology shows a very steady slow overhead increase with the number of SMs. It clearly outperforms its mesh counterpart as it barely approaches the 5% ratio at 100-SM population. The performance of the biconnected tree NANBB topology lies in between with a strong similarity to that of the tree. It barely exceeds 50% as the number of SMs grows to a hundred.

For the data loss, the comparative average packet loss for 1-, 2-, and 3-communication link failure is shown in Figure 15. It can be noticed that, in the case of the biconnected tree NANBB topology, there is no data loss until 2 communication links fail all together whereas, for the spanning tree, data loss occurs as soon as a single communication link fails. The robustness of the biconnected tree remains significantly prevalent compared to spanning tree as the number of link failures increases reasonably to 3.

## 8. Conclusions

A framework for tree based communication network augmentation is proposed to improve the robustness of data collection in neighborhood area network building blocks of AMI. It merges the virtues of mesh as well as tree based network topologies: signaling overhead increases reasonably not exceeding 15% for connectivity up to 3. It grows significantly to almost 80% as connectivity approaches 5 (Figure 13). Sufficient redundancy is also provided to mitigate the tree single point of failure such that no data loss occurs until $\lambda$ adjacent links fail simultaneously. Thus, with the connectivity, while the data loss decreases on one hand, the signaling overhead increases on the other hand, which prompts the search for a trade-off topology to fulfill all together the real-time and availability requirements of AMI. Biconnected tree NANBB ($\lambda = 2$) performance is in close similarity to that of spanning tree in terms of signaling overhead; regardless of the size of the NANBB network, the signaling overhead difference between the tree and the biconnected tree is less than 5% (Figures 11, 12, and 13). It also provides minimum redundancy as data loss occurs only when up to 2 adjacent links fail all together which is rare (Figure 15).

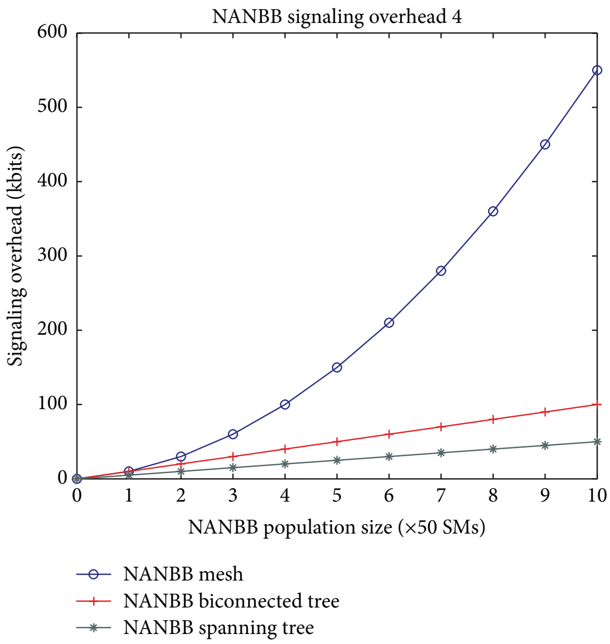


Figure 14: Mesh, biconnected tree, and tree overhead comparison.

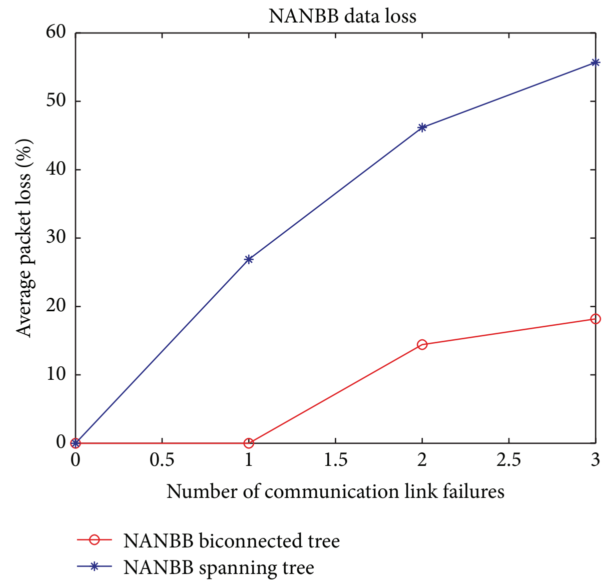


Figure 15: Tree, augmented tree data loss comparison.

A proactive topology repair scheme is proposed to address the impact of SM node failure on the augmented tree network connectivity. It relies on a loop detection scheme (Figure 9) to define the criticality of a SM node and specifically targets the cut SM node by selecting a set of supplemental parents as backup to all its children SM nodes. These links come into action immediately after the cut SM fails. By this means, the topology mutates and the AMI network recovers the connectivity by isolating the failing critical SM node (Figure 10).

The proposed scheme however lacks in root redundancy so that any faulty RG will disconnect the corresponding NANBB. Fortunately, many existing research works such as [11] on network reliability in the context of AMI have focused on the cellular network backhaul augmentation that corresponds to the data aggregator unit (i.e., RG) interconnection.

## Disclosure

The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, DOE, or the Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R&E)) or the US Government.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The first author is supported by the HBGI Scholarship. This research work is supported in part by the US National Science Foundation (NSF 1238859), the US DOE NNSA Sam P. Massie Chair Program, and the Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R&E)) under Agreement no. FA8750-15-2-0119.

## References

[1] F. Ye, Y. Qian, and R. Q. Hu, "Energy efficient self-sustaining wireless neighborhood area network design for smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 220–229, 2015.

[2] J. Kamto, L. Qian, J. Fuller, J. Attia, and Y. Qian, "Key Distribution and management for power aggregation and accountability in Advance Metering Infrastructure," in *Proceedings of the IEEE 3rd International Conference on Smart Grid Communications (SmartGridComm '12)*, pp. 360–365, Tainan, Taiwan, November 2012.

[3] X. Li, L. Qian, and J. Kamto, "Secure anonymous routing in wireless mesh networks," in *International Conference on E-Business and Information System Security (EBISS '09)*, pp. 1–5, May 2009.

[4] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, 2005.

[5] M. S. Siddiqui and C. S. Hong, "Security issues in wireless mesh networks," in *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE '07)*, pp. 717–722, Seoul, South Korea, April 2007.

[6] R. C. Parks, *Advanced Metering Infrastructure Security Considerations*, Sandia National Laboratories, Albuquerque, NM, USA, 2007, http://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/20-AMI_Security_Considerations.pdf.

[7] *AMI System Security Requirements V1.01*, AMI SEC Task Force, 2008, http://energy.gov/oe/downloads/ami-system-security-requirements-v101-1.

[8] T. Ishii, "Graph augmentation problem with diameter requirements," in *Proceedings of the 3rd International Conference on emphNetworking and Computing (ICNC '12)*, pp. 393–398, December 2012.

[9] T. Watanabe, Y. Higashi, and A. Nakamura, "An approach to robust network construction from graph augmentation problems," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 2861–2864, IEEE, May 1990.

[10] A. Dutta and P. Kubat, "Design of partially survivable networks for cellular telecommunication systems," *European Journal of Operational Research*, vol. 118, no. 1, pp. 52–64, 1999.

[11] C. Charnsripinyo and D. Tipper, "Topological design of 3G wireless backhaul networks for service assurance," in *Proceedings of the 5th International Workshop on Design of Reliable Communication Networks*, p. 9, October 2005.

[12] A. A. Abbasi, M. F. Younis, and U. A. Baroudi, "Recovering from a node failure in wireless sensor-actor networks with minimal topology changes," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 256–271, 2013.

[13] M. Younis, S. Lee, S. Gupta, and K. Fisher, "A localized self-healing algorithm for networks of moveable sensor nodes," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 1–5, New Orleans, La, USA, December 2008.

[14] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258–271, 2010.

[15] J. Silber, S. Sahu, J. Sing, and Z. Liu, "Augmenting overlay trees for failure resiliency," in *Proceedings of the Global Telecommunications Conference (GLOBECOM '04)*, vol. 3, pp. 1525–1531, IEEE, December 2004.

[16] Q. Li, W. Ni, Y. Li, H. Zhang, and X. Zheng, "Incremental network design with topology augmentation on backup path provisioning in WDM mesh networks," in *Proceedings of the Asia Communications and Photonics Conference and Exhibition (ACP '10)*, pp. 276–277, IEEE, Shanghai, China, December 2010.

[17] R. Diestel, *Graph Theory*, vol. 173, Springer, 4th edition, 2010.

[18] S. P. Jain and K. Gopal, "On network augmentation," *IEEE Transactions on Reliability*, vol. 35, no. 5, pp. 541–543, 1986.

[19] T. Fukunaga and H. Nagamochi, "Approximating a generalization of metric TSP," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 2, pp. 432–439, 2007.

[20] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," *Journal of the Association for Computing Machinery*, vol. 41, no. 2, pp. 214–235, 1994.

[21] E. Knuth Donald, *The Art of Computer Programming*, vol. 1, Wesley, Boston, Mass, USA, 3rd edition, 1997.

[22] BFS, "Breadth first search and depth first search," February 1996, https://www.ics.uci.edu/~eppstein/161/960215.html.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Depth-first search," in *Introduction to Algorithms*, pp. 540–549, MIT Press, McGraw-Hill, New York, NY, USA, 2nd edition, 2001.

[24] A. Leon-Garcia and I. Widjaja, *Communication Networks, Fundamental Concepts and Key Architectures*, McGraw-Hill, 2nd edition, 2003.