December 2021

WI-SUN STABILIZATION FOR DISTRIBUTED AUTOMATION GATEWAYS BY DYNAMICALLY CONTROLLING ACCEPTABLE CHILDREN NUMBERS

Lele Zhang

Iris Dai

Xue Yuan

Tonglin Yuan

WI-SUN STABILIZATION FOR DISTRIBUTED AUTOMATION GATEWAYS BY DYNAMICALLY CONTROLLING ACCEPTABLE CHILDREN NUMBERS

AUTHORS:
Lele Zhang
Iris Dai
Xue Yuan
Tonglin Yuan

ABSTRACT

Resilient mesh products (based on, for example, the Wireless Smart Utility Network (Wi-SUN) communication standard) have been available for many years and are widely used within Advanced Metering Infrastructure (AMI) and Distributed Automation (DA) settings. A DA Gateway (which, for convenience, may be referred to herein as a 'DAGW') follows the AMI elements' rules in a Wi-SUN network, but that is not highly efficient while forwarding a significant amount of the third-party traffic. The root cause of such inefficiency is that the child AMI elements are not aware of a change in the available resources in a DAGW. To address these types of challenges, techniques are presented herein that support a DAGW periodically broadcasting its available resources in Destination Oriented Directed Acyclic Graph (DODAG) Information Object (DIO) messages. Aspects of the presented techniques support a DAGW adjusting the number of its attached child AMI elements, thus stabilizing the whole network.

DETAILED DESCRIPTION

Resilient mesh products (based on, for example, the Wireless Smart Utility Network (Wi-SUN) communication standard) have been available for many years and are widely used within Advanced Metering Infrastructure (AMI) and Distributed Automation (DA) settings. For example, one large network equipment vendor has deployed more than 17 million AMI products in customer environments. However, DA products are rarely used as much as AMI products because a DA Gateway (which, for convenience, may be referred to herein as a 'DAGW') is different from conventional AMI products.

Figure 1, below, presents elements of an exemplary mesh environment.

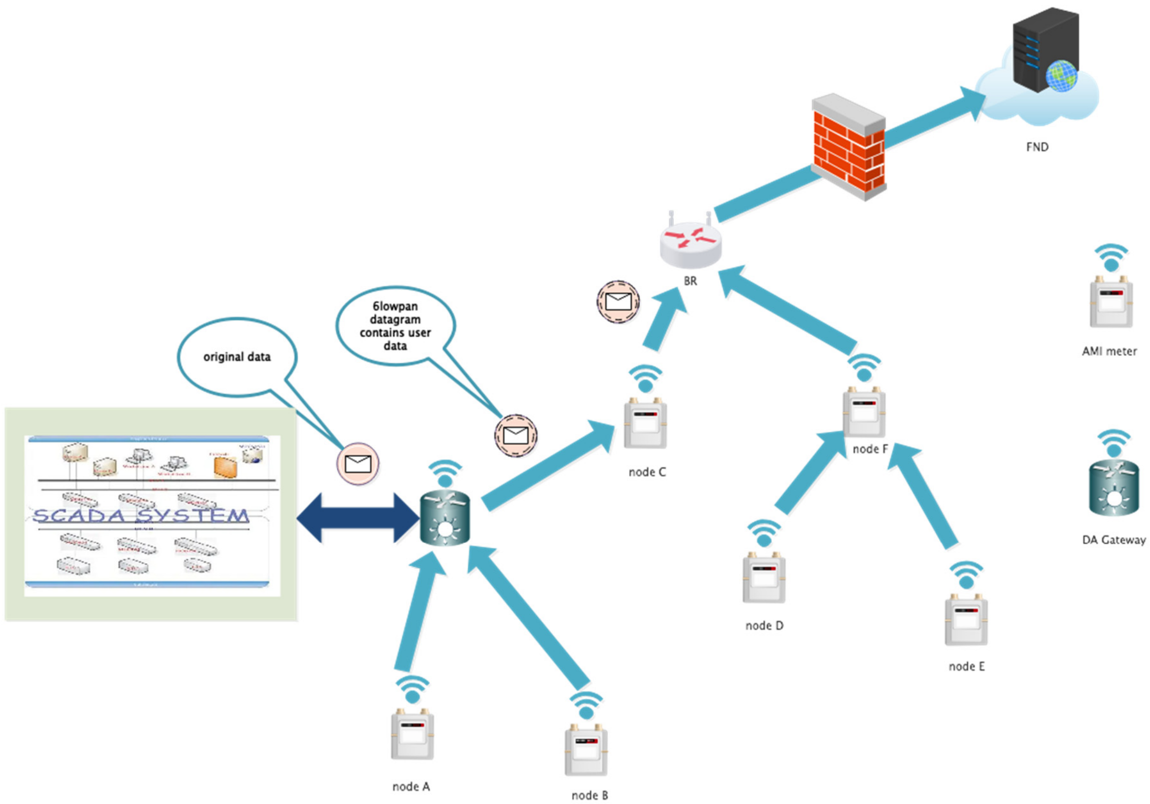

Figure 1: Exemplary Mesh Environment.

Generally speaking, AMI-based environments reside in an idle mode most of the time. Typically, they spend less than 20% of their available cycles transmitting traffic. As a result, the Internet Protocol version 6 (IPv6) Routing Protocol for Low-Power and Lossy Networks (RPL), as defined by the Internet Engineering Task Force (IETF) Request for Comments (RFC) 6550, may be used to establish a tree-like topology network wherein the intermediate nodes have sufficient time to forward uplink and downlink traffic. Rather than the AMI elements, the DAGWs are in charge of forwarding the third-party traffic, which means that their throughput is much larger than the AMI elements.

An arrangement as described above raises assorted challenges. One potential challenge is that a DAGW still can be a coordinator in a field area network (FAN), which means that it can be an RPL routing node for other AMI elements. When a DAGW is not busy, the arrangement will work well. But, if the third-party traffic increases rapidly such burst traffic will likely make the connection worse between the AMI elements and the DAGW, resulting in a significant number of dropped packets.

Another challenge is that a DAGW only can be a leaf node in a FAN, which means that it can only forward the third-party traffic rather than be a routing node for other AMI elements. Such an arrangement will waste DAGW resources because a DAGW will not always be busy.

As will be described and illustrated in the narrative that is presented below, techniques are presented herein that support a DAGW adjusting the number of connectable children to ensure stable and reliable connections while forwarding both the third-party traffic and the child AMI elements' traffic.

Currently, a DAGW is used as a Wi-SUN full functionality node (FFN), which can forward both child AMI element traffic and the third-party traffic such as Supervisory Control and Data Acquisition (SCADA) traffic. Based on observations, there are several actual operational situations that may be considered, as follows:

- Case 1: a DAGW is not busy forwarding the third-party traffic so it can still accept more child AMI elements.
- Case 2: a DAGW employs many of its resources forwarding the (e.g., high priority) third-party traffic and consequently has limited resources to devote to the potential child AMI elements.
- Case 3: a DAGW uses all of its resources forwarding the massive third-party traffic resulting in it needing to disconnect some child AMI elements in order to release resources.

Based on the above description, the second and the third cases require attention in practice.

To address the types of challenges that were described above, techniques are presented herein that enable a DAGW to dynamically adjust the acceptable number of child nodes (based on, for example, the currently available resources) and broadcast such information to its neighbors.

Under aspects of the techniques presented herein a DAGW is able to determine the number of all acceptable child AMI elements based on the currently available resources (such as, for example, memory, bandwidth, etc.). Assume that R_{dagw} represents the total resources of a DAGW, that R_{3rd} represents the resource consumption that is associated with forwarding the third-party traffic, and that R_{ami} denotes the resource consumption of a child

AMI element. The following formula may be used by a DAGW to determine the maximum number of connectable child AMI elements:

$$Num_{max\_children} = (R_{dagw} - R_{3rd})/R_{ami}$$

Using the above formula, a DAGW may calculate the $Num_{max\_children}$ value per each Destination Oriented Directed Acyclic Graph (DODAG) Information Object (DIO) period and then inject that value into the DIO message as a new option. Figure 2, below, illustrates a portion of the DIO Base Object (as defined under Section 6.3.1 of RFC 6550 (i.e., the RPL specification)).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |Version Number |             Rank              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|0| MOP | Prf |     DTSN      |     Flags     |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID                            +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 2: DIO Base Object

Currently, DIO options support the following selections:

`0x00` Pad1

`0x01` PadN

`0x02` DAG Metric Container

`0x03` Routing Information

`0x04` DODAG Configuration

`0x08` Prefix Information

Aspects of the techniques presented herein support a new DIO option that allows a DAGW to broadcast the resource usage information. The following code snippet illustrates aspects of such an option:

```
struct rpl_opt_dagw_children_stats {
  uint16_t max_acceptable_children;  /* the theoretical upper limit of connectable children AMIs */
  uint16_t cur_attached_children;    /* the current connected children AMIs */
};
```

Aspects of the techniques presented herein further support neighboring AMI elements executing a set of rules as long as they receive a DIO message, as described above, from a DAGW. The set of rules will be briefly described below.

A first rule encapsulates that if the node has not yet joined the network, it will determine whether to select the instant DAGW as a parent based on this RPL option. If the maximum acceptable number of children is smaller than the existing number of children, the node will not select this DAGW as a potential RPL parent. Otherwise, the node may consider this DAGW as a parent.

A second rule encapsulates that if the node has been a child of the instant DAGW, but the latest DIO message indicates the existing number of child AMI elements exceeds the limitation of the available resources, then the node will consider detaching from the original DAGW with a certain probability. For example, if a DAGW has ten child AMI elements, but at one moment the available resources only can support eight child AMI elements (e.g., the DAGW needs more resources for increased third-party traffic such as, for example, SCADA traffic), then the DAGW will broadcast a DIO message to notify the child AMI elements. The child AMI elements will know that the DAGW needs to evict two children, so every child has a 20% probability (i.e., (10-8) / 10) of detaching from the instant DAGW.

A third rule encapsulates that if the node is a child of other nodes rather than the instant DAGW, then it will consider this DAGW as a candidate based on the DIO message. If the DAGW still has spare resources, the node may switch to this DAGW when the link quality to the current parent node becomes worse.

Aspects of the techniques presented herein may be further explicated with reference to an instructive example. That example comprises four phases, each of which will be described and illustrated in the narrative that is presented below.

During a first phase, a DAGW has three child AMI elements (as depicted in Figure 3, below) which are nodes A, B, and C.

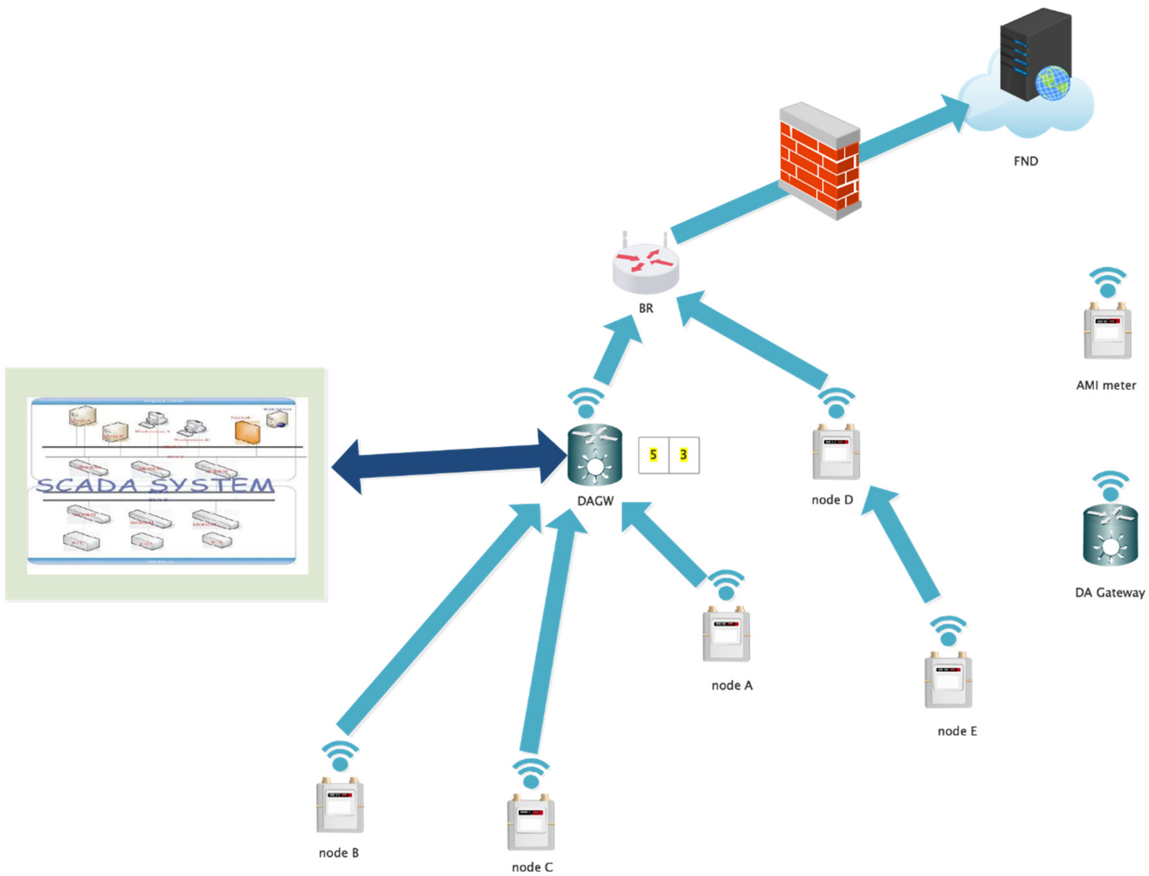

Figure 3: Phase One

As depicted in Figure 3, above, the DAGW declares that it can have at most five children and it has already been attached by three children.

During a second phase, at some point in the future the third-party traffic consumes more resources of the DAGW. Consequently, the DAGW updates this information in the next DIO message. The children (i.e., nodes A, B, and C) receive that DIO and thus all of them have a 33% probability of switching to another node as a new parent. Figure 4, below, depicts this activity.

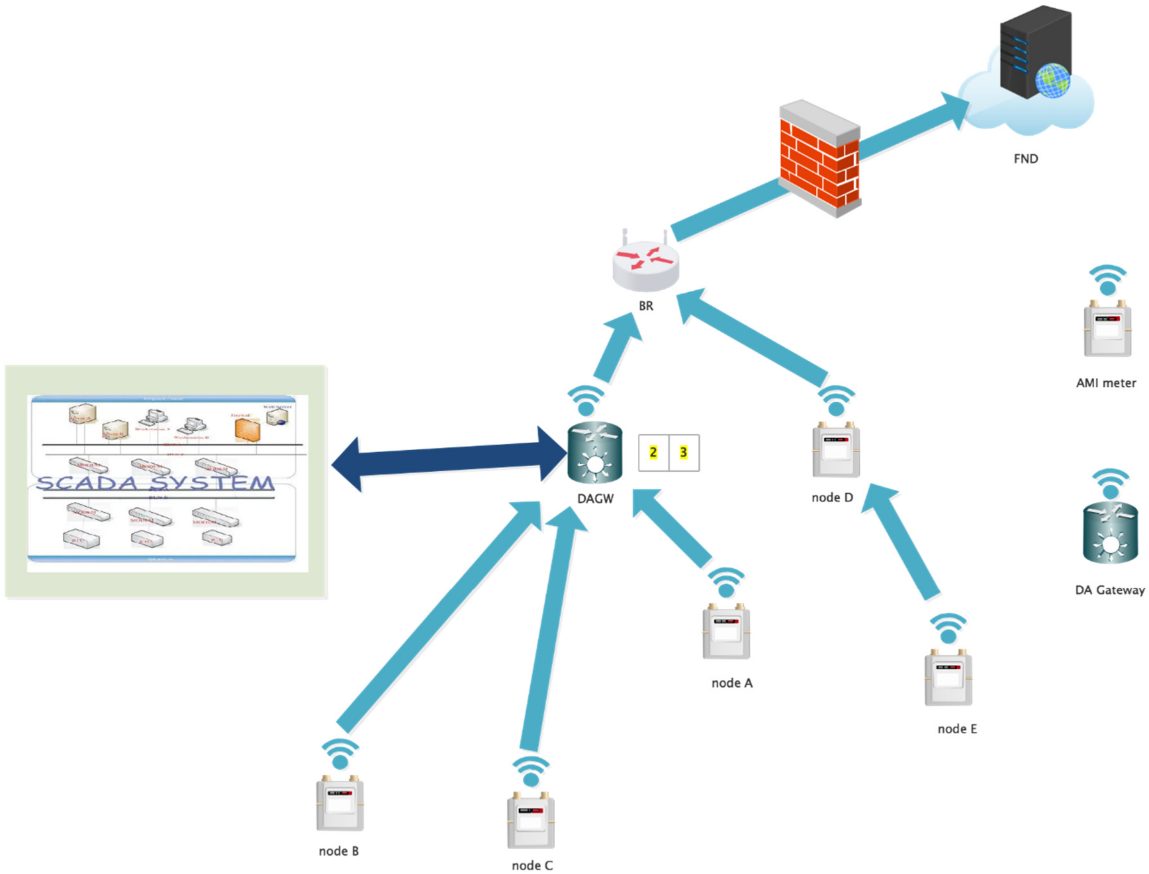

Figure 4: Phase Two

During a third phase, nodes A, B, and C have the same probability of detaching from the DAGW. Consequently, it is possible that all of them (i.e., all three of the nodes), or two nodes, or just one node will leave the DAGW at the same time. The instant example assumes that node A eventually leaves the DAGW, as depicted in Figure 5, below.

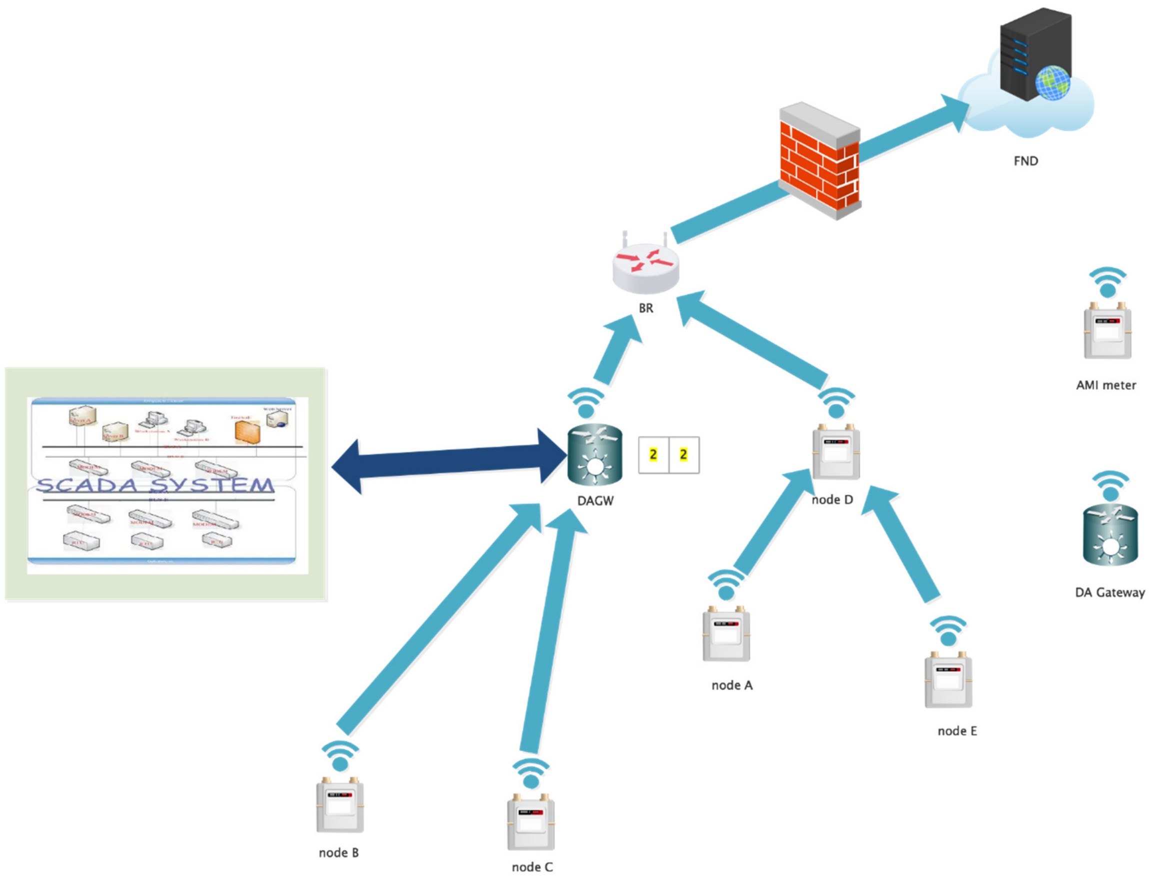

Figure 5: Phase Three

During a fourth (and final) phase, a few hours later the DAGW has more resources for child AMI elements so it updates the DIO information in the neighborhood. The neighboring AMI elements will consider the DAGW as a preferred parent based on the current path link quality. Finally, both node A and node D attach to the DAGW under this scenario, as depicted in Figure 6, below.

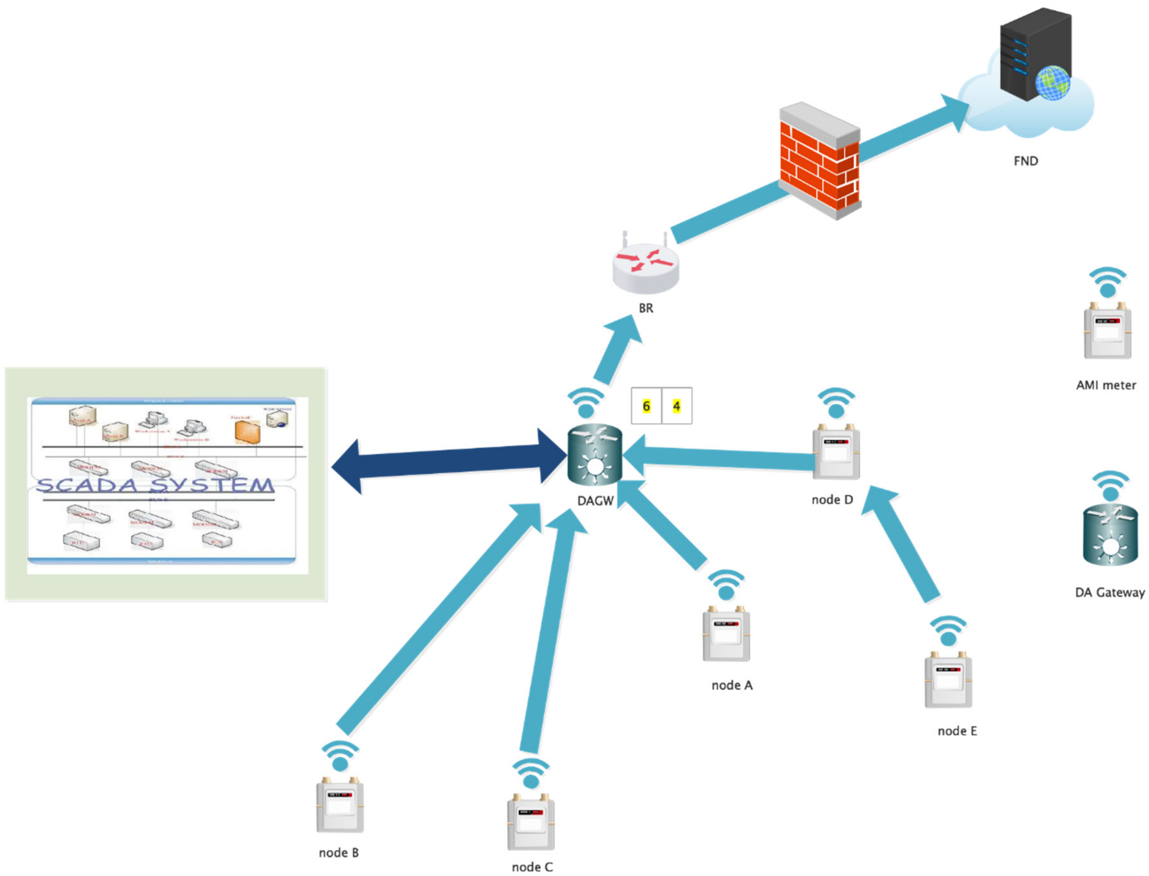

Figure 6: Phase Four

It is important to note that existing solutions just address how a node may select a better parent node when the node is in a join stage. According to the current mechanism, the child node just considers link expected transmission count (ETX) and path ETX for selecting a preferred parent node, which will lead to an unreasonably fat tree topology. Such a mechanism raises no problem because the late coming child nodes cannot evict the existing associated child nodes. For example, for a routing node that can support ten child nodes at most and that already has ten child nodes, an eleventh node would not select it as a parent node.

In contrast, aspects of the techniques presented herein support a DAGW evicting some number of existing AMI elements if the DAGW is short of resources for supporting third-party traffic. For example, if a DAGW can support ten AMI elements at most and it already has ten child AMI elements, but at a particular moment it needs to evict four child AMI elements in order to be able to support forwarding high-priority third-party traffic, under aspects of the techniques presented herein the DAGW may broadcast this information in a next DIO message. Then, all of the received child AMI elements have a 40% probability of leaving the instant DAGW. Consequently, there are three possible results.

Under a first result, there are exactly four child AMI elements that leave the instant DAGW (thus obviating the next steps). Under a second result, there are less than four child AMI elements that leave the instant DAGW (e.g., perhaps just two AMI elements leave). The DAGW will update this information in the next round DIO message and then the eight remaining AMI elements will each have a 25% probability of leaving this DAGW during the next DIO period. Under a third result, there are more than four child AMI elements that leave the instant DAGW (e.g., perhaps five AMI elements leave). The DAGW will update this information in the next round DIO message. Then, the other AMI elements will potentially attach to this DAGW because it still can accept one child AMI element. Finally, the DAGW will have a new balance after several rounds of DIO propagations.

In summary, techniques have been presented herein that support a DAGW periodically broadcasting its available resources in DIO messages. Aspects of the presented techniques support a DAGW adjusting the number of its attached child AMI elements, thus stabilizing the whole network.