October 16, 2018

A VIRTUAL SUB-DOMAIN ROUTING MECHANISM TO IMPROVE ROUTING EFFICIENCY FOR LOW-POWER AND LOSSY NETWORKS

Dechen Xu

Jianfeng Mao

Xiang Fang

A VIRTUAL SUB-DOMAIN ROUTING MECHANISM TO IMPROVE ROUTING EFFICIENCY FOR LOW-POWER AND LOSSY NETWORKS

AUTHORS:
Dechen Xu
Jianfeng Mao
Xiang Fang

ABSTRACT

Techniques are provided herein to optimize the routing path between normal nodes and a fog node in a non-storing mesh network. The transmission between normal nodes and the application fog node is more efficient and little additional calculation or storage is required on the normal nodes.

DETAILED DESCRIPTION

In an Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 mesh network, the routing table is generated based on Routing Protocol for Low-power and Lossy Networks (RPL) and maintained on a grid router. For the upstream traffic, each node sends or forwards the data to its next hop. For the downstream traffic, the route information is added to an Internet Protocol (IP) source routing header. Each node thus knows where the data should be forwarded. In those application where the node only communicates with the server behind the grid router, this kind of routing works well.

However, as illustrated in Figure 1 below, applications where the nodes need to communicate each other in same mesh network are more complex.

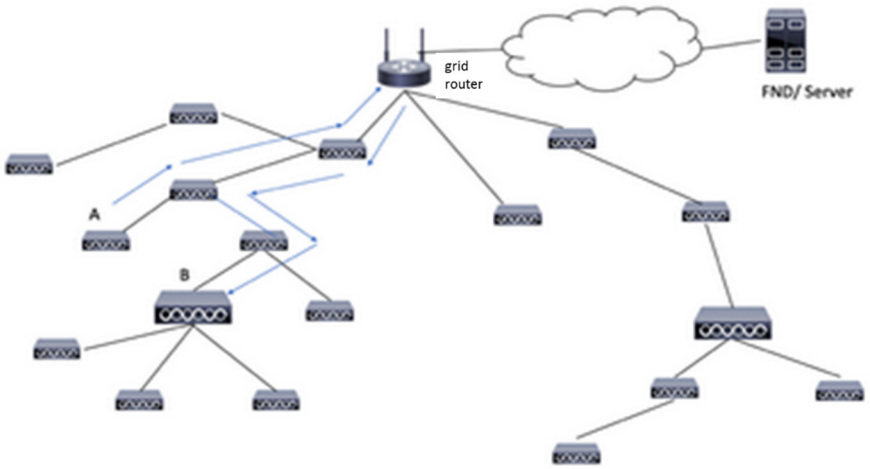

Figure 1

Published by Technical Disclosure Commons, 2018

In this example, Node A needs to send data to Node B. In a non-storing mode mesh network, the data needs to be forwarded to the grid router, which knows how to send the data to Node B. This transmission requires tremendous radio resources. Furthermore, the increment of the node hops will cause a large delay in the data.

For these kinds of applications, the storing mode can resolve the problem, but this requires the nodes in the network to have more hardware resources such as Central Processing Units (CPUs) and Random Access Memory (RAM) capability because those nodes need to store all their downstream routing tables.

In some deployments, the user needs to perform some data processing in certain nodes instead of the head end server, such as fog computing. As shown in Figure 1, Node B, a fog key node, is a device with more powerful hardware resource than other nodes in the mesh network. As such, Node B collects the data from other nodes. As an example involving sensors, in traditional route mode, if Node B is not in the path of the upstream, the data would need to be forwarded to the grid router and then sent to Node B. Considering the large latency and limited bandwidth of Low-power and Lossy Networks (LLNs), more and more users prefer enabling the fog computing in the deployment, which means in the LLNs most nodes are thin nodes which only have very limited CPU/memory and network capabilities to support limited sensor data collections or simple action execution. At the same time, they will deploy some fog nodes in the network, which have more power capability and can support collecting data within some local areas with related nodes, then perform edge computing and consolidate the data to decide which need be sent out to the cloud to minimize the bandwidth requirement. Also, some actions may be taken based on local Artificial Intelligence (AI) to enhance responsibility. Such use cases need the fog node to make a virtual sub-domain within the whole mesh network. In this sub-domain, the fog node can communicate with the related nodes by the shortest possible path.

Presented herein is a mechanism to improve the efficiency of this kind of transmission without requiring additional calculation or storage requirements on normal nodes. Introduced is a routing mechanism in some fog computation application deployment to make some nodes in a mesh network reach other nodes in a short path instead of the traditional RPL tree routing path in order to collect data or issue control commands more

efficiently. These nodes are referred to as fog nodes, and have more powerful hardware resources such as powerful CPU / Microcontroller Unit (MCU) capability and large RAM.

As illustrated in Figure 2 below, in a mesh network, after the nodes go online, a Destination Oriented Directed Acyclic Graph (DODAG) is generated and stored on the grid router and head end server side as a RPL tree.

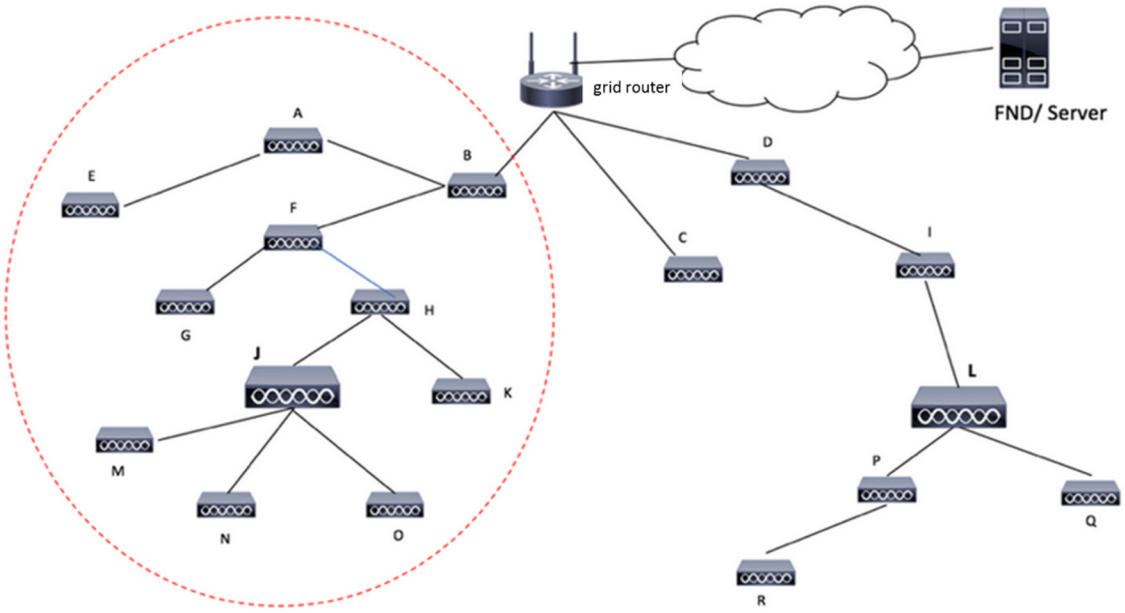

Figure 2

Figure 3 below illustrates the RPL tree.

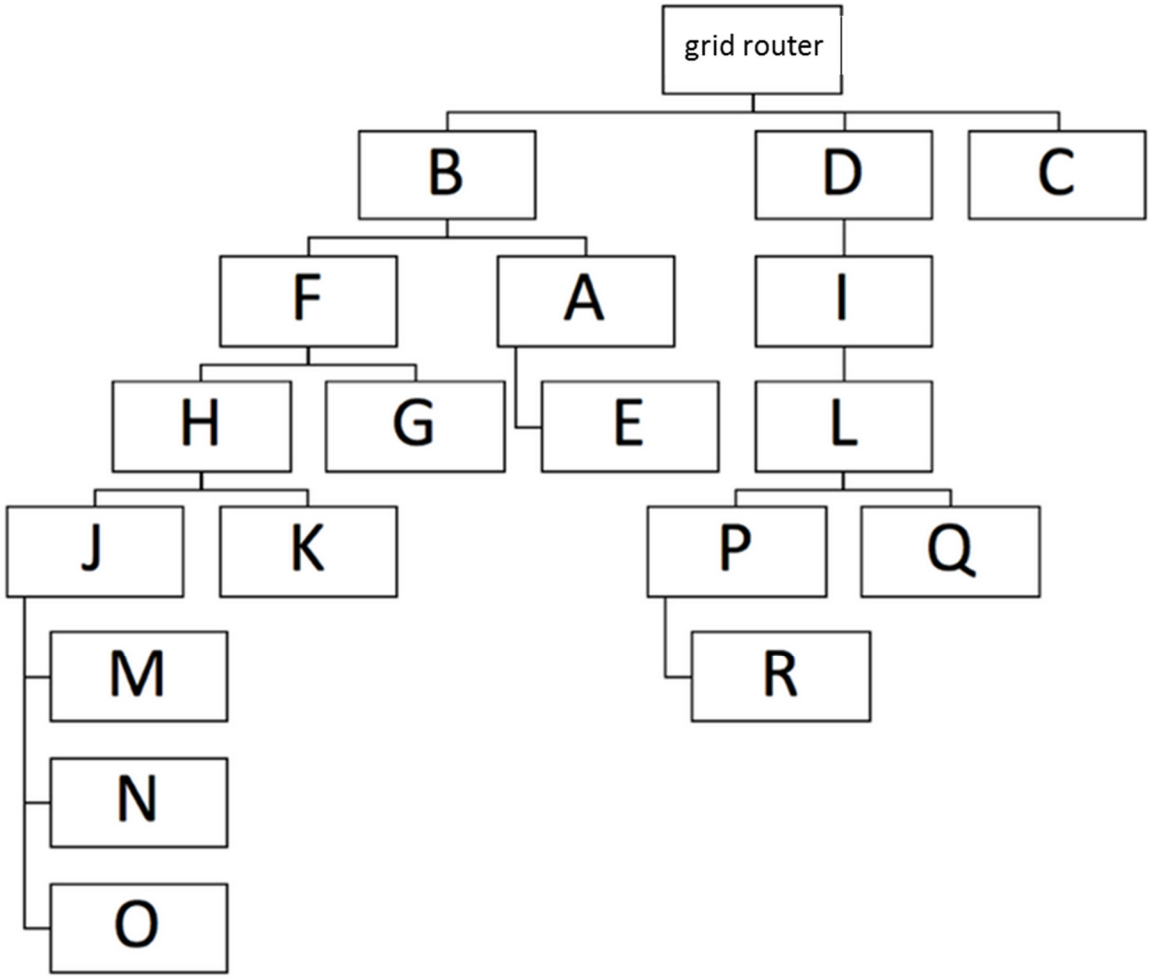

Figure 3

Suppose Node J is the fog node and that it wants to collect the data from devices around it (e.g., the nodes in the red circle in Figure 1).

The first example step involves a DODAG calculation. For Node J, the grid router or Head-End System (HES) needs to generate a DODAG in which device J is the root based on the RPL tree. We call the CGR or HES Path Calculation Element (PCE).

As illustrated in Figure 4 below, the calculated DODAG shows the route from Node J to any other nodes.

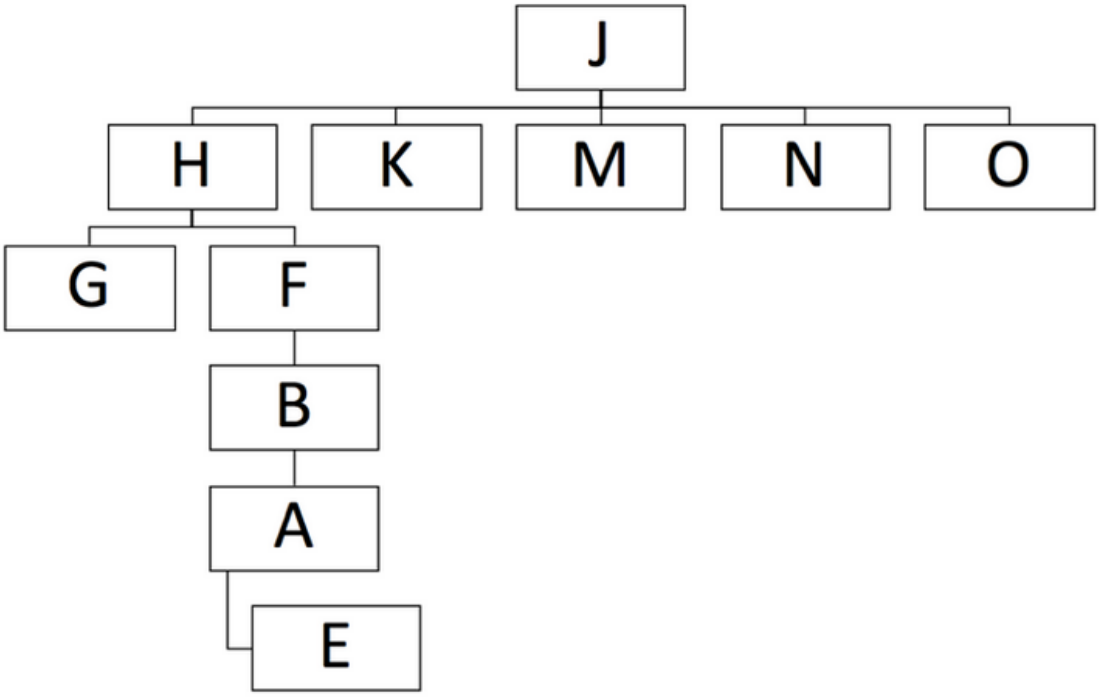

Figure 4

The PCE sends the generated DODAG information to Node J. Node J then knows the route to the other nodes, and this information may be carried in a source routing header in the notification message described below.

A second example step involves special route notification. For those nodes that need to send data to Node J, they cannot use the original next hop as the default gateway. As such, frames whose destination IP is Node J should be forwarded to a secondary gateway. In order to make the nodes learn the secondary gateway used to send data to the fog node, the fog node needs to notify the nodes how to send the data to it.

An Internet Control Message Protocol (ICMP) frame with a source routing header may be used as a notification frame to notify other nodes. Since there may be multiple nodes on a routing path, for each path, there is only one frame required. The final destination is the farthest node. Like the normal downward frames, the source routing header of the frame contains the destination route information. While a node receives such a frame, the node should check the original global IP address in the IPv6 header to

determine whether it is a valid address and then use the source link-local address as its secondary address.

As illustrated in Figure 5 below, Node J sends the route update notification to Node G. The notification frame should be forwarded by Nodes H and F. First Node H receives the frame, and it should update the secondary next hop to Node J. Node H then forwards the frame to Node F, which should update the secondary next hop to Node H and forward the frame to the final destination (Node G). Node G should update the secondary next hop to Node H. After this operation above, each node knows when it needs to send the data to Node J, and which gateway should be used for sending.

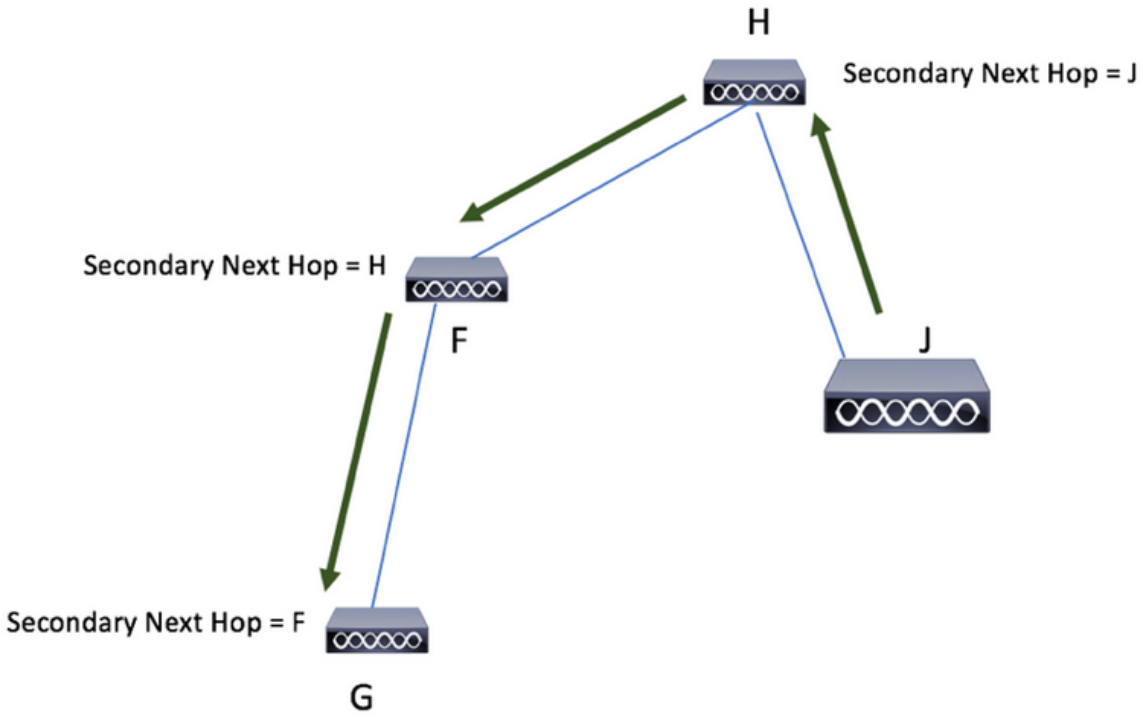

Figure 5

However, in some cases with certain topologies, a frame that needs to pass between two nodes that are close and are neighbors for each other travels via another node between those two nodes. Figure 6 below illustrates a short path mechanism. Here, Node F is the neighbor of Node J. While Node H forwards the route notification frame to Node F, Node H adds its source node information, including address, link Expected Transmission (ETX), and hop value into the IP hop-by-hop option. After Node F receives the frame, it checks whether there is a node in the hop-by-hop option that is also in its neighbor list. If a node

was found in its neighbor list and the ETX is in an acceptable range, the node should use it as its secondary next hop. In this example, Node F should select Node J as its secondary next hop. If there are multiple nodes in the hop-by-hop option in the neighbor list and the ETX are all acceptable, the node should select the node whose hop value is smallest for the secondary next hop. That should be the shortest path to fog Node J.

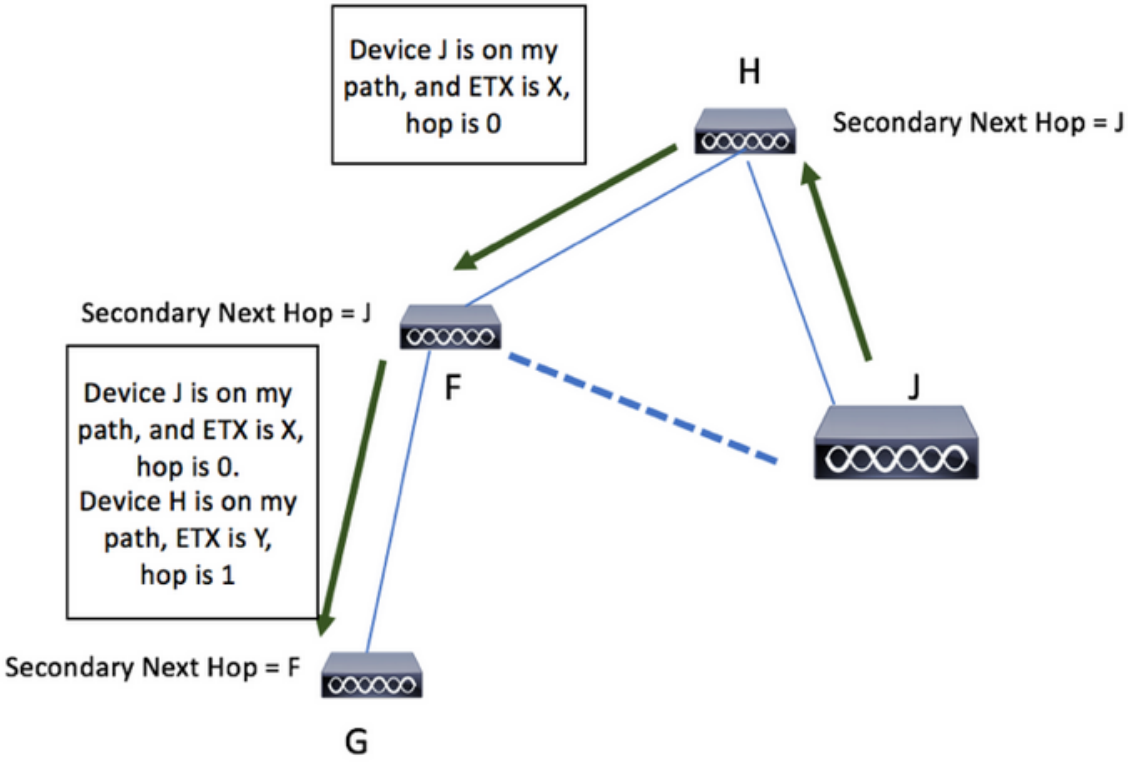

Figure 6

Example step three involves a route notification acknowledgement. After the nodes receive the route notification frame, nodes reply with an acknowledgement in order to indicate to the origin fog node that the route has been updated and whether there any short path exists. The acknowledgement frame is also an ICMP frame with the ICMP option, and the target address is the origin fog node so the acknowledgement will be sent to the secondary next hop. This is illustrated in Figure 7 below.

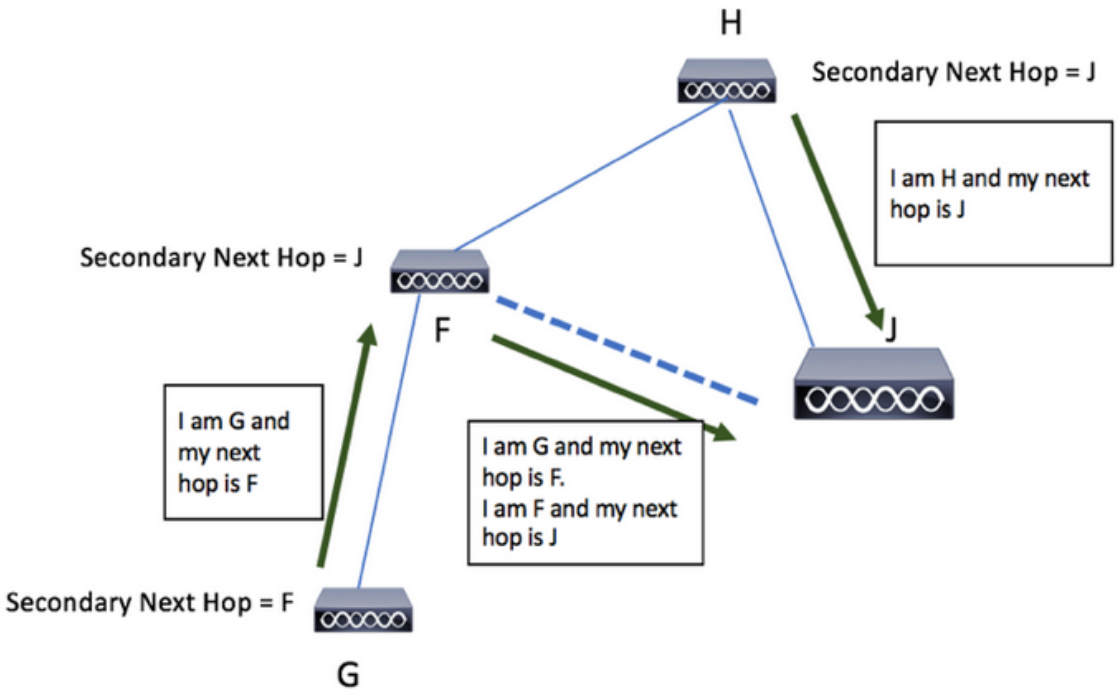

Figure 7

Each intermittent node sets a route notification acknowledgment reply timer after receiving a route notification. The timer value depends on its address hop in source routing. The closer to the destination, the smaller the timer value. The last Node G should reply to the acknowledgment immediately and put its next hop information in the hop by hop option. If Node F receives the acknowledgment within the timer value, it should not send the acknowledgment independently and instead simply insert its next hop information into the same acknowledgment and forward it to its next hop. Because Node F's secondary next hop is no longer Node H but instead Node J, Node H will not receive the acknowledgment from Node F. After the acknowledgment timer timeout, Node H sends its own route notification acknowledgment.

After origin fog Node J receives the acknowledgment, it should also update the route if there is short path. As illustrated in Figure 7, the route to Node J should be updated.

If fog Node J does not receive an acknowledgment from a node, it will send the notification again after a time interval. If fog Node J does not receive the acknowledgment after several tries, the notification should not be sent on this path again. This may be caused by node failure on the upward path, but the Destination Advertisement Object (DAO) is

not updated, so the corresponding DODAG tree is not updated. In this case, Node J should ask the PCE for the new route and resend the notification.

Example step four involves routing table updating. Nodes send the DAO to the grid router to update the DODAG periodically, so there is also an updating period that can be set so that the PCE can send the updating of the DODAG to fog node. The PCE can only send the updated path to the fog node and the fog node should send the new route notification to those nodes whose routing paths are updated (e.g., repeat the third example step).

As illustrated in Figure 8 below, due to a routing update, Node M's next hop changes from Node J to Node N, and the updated DODAG will be sent to Node J.

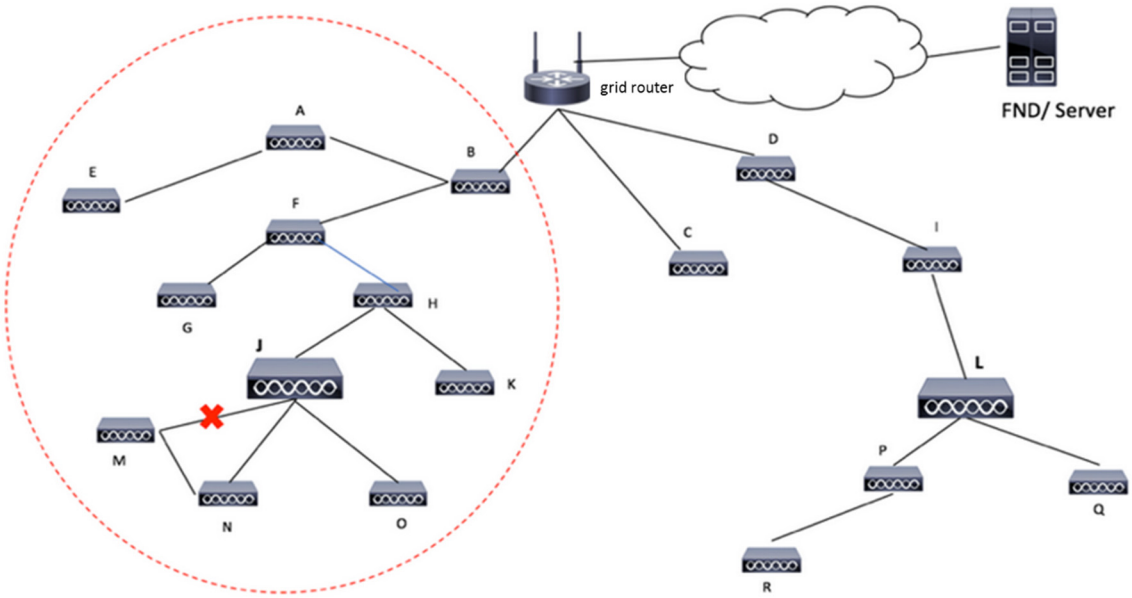

Figure 8

After Node J receives the updated DODAG, it sends the route notification to Node M to update the secondary next hop address.

A fifth example step involves routing failure. A failure may occur during a DODAG updating period. For the upward process (normal nodes to fog nodes), the node knows the sending status. If the sending status indicates failure on secondary next hop, an option will be inserted into the IP hop-by-hop header to tell the fog node there is a failure occurred and then the data will be forwarded to the primary next hop. If the primary next hop node has the secondary next hop, because the destination IP address is the fog node's address, the data will still be sent on the secondary next hop. However, if the secondary next hop is still

unavailable, the failure information will be inserted into the hop-by-hop header and then the frame will be forwarded on primary next hop. If the primary next hop node does not have the secondary next hop, the data will be forwarded on primary next hop, which means the data will follow a normal RPL path to the grid router and then be forwarded to fog Node J.

After fog Node J receives a data frame containing the destination IP header including failure information, it should ask the grid router or HES for a new path and then send the route notification on this path to try to fix the problem. This is illustrated in Figure 9 below.

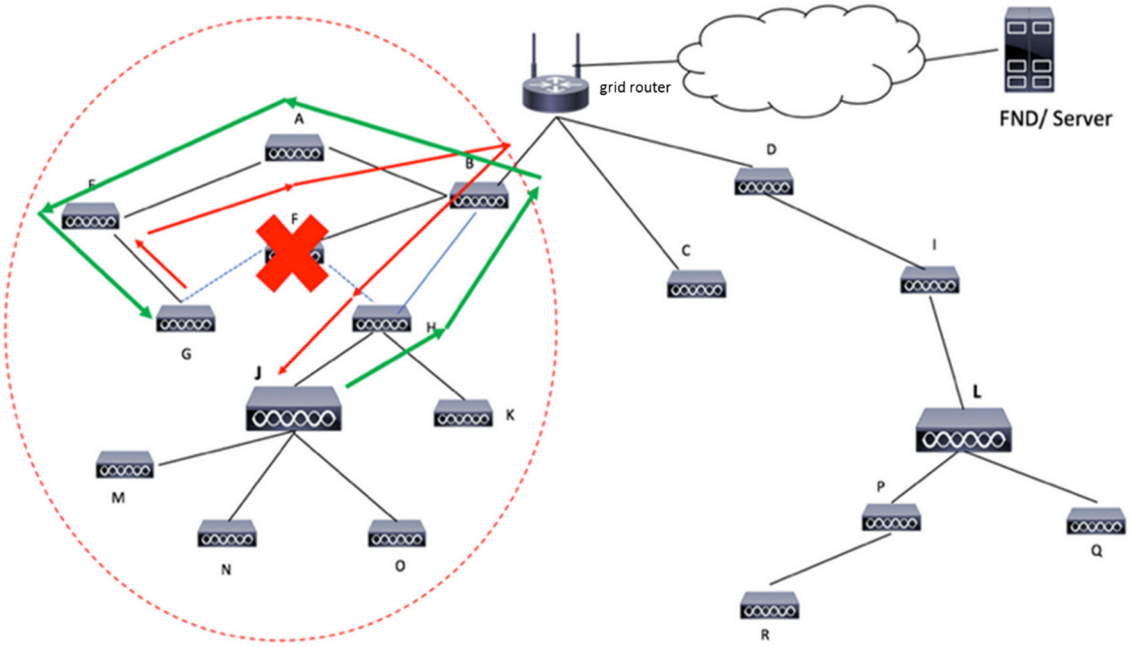

Figure 9

Node G sends data to fog Node J, but because of the Node F failure, the secondary next hop is unreachable. But the primary next is already updated, so the failure information will be inserted into the IP hop-by-hop header and the data will be forwarded by the traditional RPL path to Node J (red line). Then Node J may try to get DODAG information from the grid router or HES, and try to get an updated routing path (green line).

In summary, techniques are provided herein to optimize the routing path between normal nodes and a fog node in a non-storing mesh network. The transmission between normal nodes and the application fog node is more efficient and little additional calculation or storage is required on the normal nodes.