Technical Disclosure Commons

Defensive Publications Series

September 18, 2019

FLOW CONTROL USING BANDWIDTH ALLOCATION STRATEGY IN LOW-POWER AND LOSSY NETWORKS

Lele Zhang

Chuanwei Li

Ling Wei

Xi Wu

Recommended Citation

Zhang, Lele; Li, Chuanwei; Wei, Ling; and Wu, Xi, "FLOW CONTROL USING BANDWIDTH ALLOCATION STRATEGY IN LOW-POWER AND LOSSY NETWORKS", Technical Disclosure Commons, (September 18, 2019)
https://www.tdcommons.org/dpubs_series/2490

FLOW CONTROL USING BANDWIDTH ALLOCATION STRATEGY IN LOW-POWER AND LOSSY NETWORKS

AUTHORS:

Lele Zhang
Chuanwei Li
Ling Wei
Xi Wu

ABSTRACT

Presented herein are techniques for reduce the risk of transmitting collision in wireless networks. The techniques presented herein use a deterministic bandwidth allocation strategy for a central device (central node). As a result, each neighbor can leverage its own schedule to transmit packets with low risk of collision. The central device spreads schedules via DODAG Information Object (DIO) messages in its neighborhood, while dynamically adjusting distribution, per round, according to predicted traffic of its neighbors. The portion of downward and guest account is determined from the software-defined networking (SDN) controller. The SDN controller can design appropriated ratio for each central device on-demand and configure them as a policy.

DETAILED DESCRIPTION

A tree-based wireless mesh network (WMN), such as a Connected Grid Mesh (CG-Mesh) or a Wireless Smart Utility Network (Wi-SUN), may face a number of communication challenges due to free-competition on the air among neighbors. Furthermore, from the perspective of each device, the WMN may appear as a STAR topology were all neighbors (no matter whether a child or a parent) around a central device are intended to compete for medium access control (MAC) when they are trying to transmit packets at the same time. This is generally represented in FIG. 1, below.

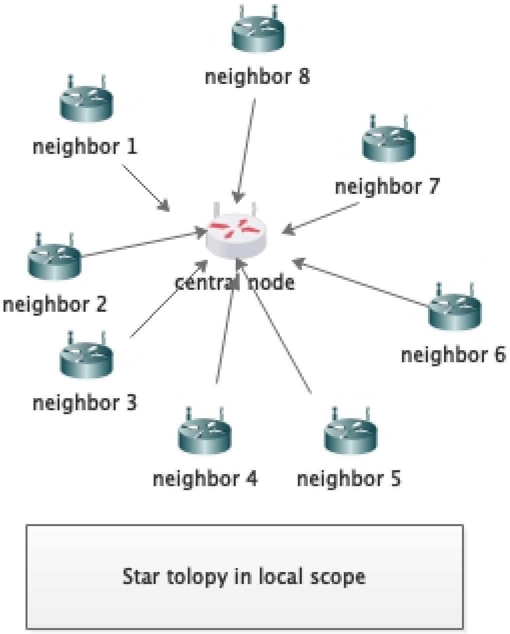


FIG. 1

The above arrangement causes two primary problems. First, if one neighbor takes up too much time during one period, other neighbors will starve to death easily. Second, in the case of concurrent traffic, if multiple neighbors try to transmit packets, the transmissions will have high probability of failure. These problems are generally shown below in FIGs. 2A and 2B, respectively.

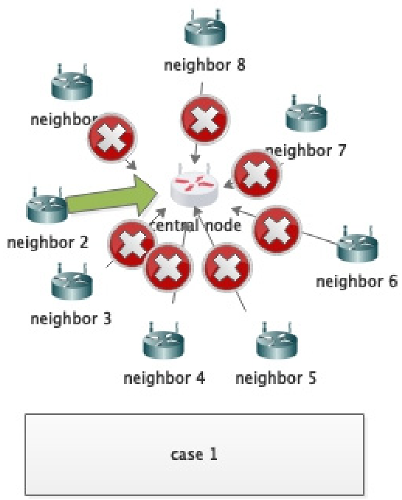


FIG. 2A

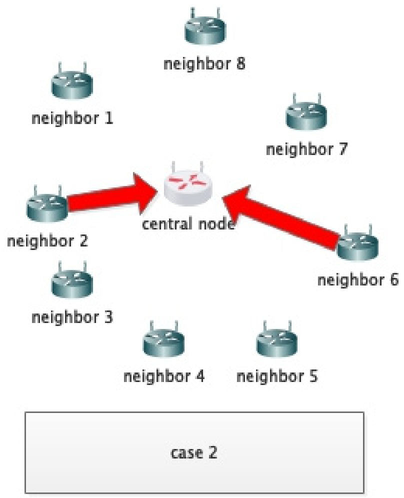


FIG. 2B

Both of the above problems seem to relate to Media access control (MAC) issue rather than bandwidth allocation (BA). However, the techniques presented herein utilize bandwidth allocation in a new manner to address these issues.

There are several existing/known methods to overcome collisions among neighbors of a central device. For example, a Time Slotted Channel Hopping (TSCH) based WMN can be used, but most of such solutions are intended to assign independent time slots for respective neighbors. IPv6 over TSCH (6TiSCH) solutions are based on DetNet principles and could, theoretically, be useful. However, in practice, 6TiSCH solutions add extra overhead for maintenance and management of the synchronization among neighbors of a central device. As such, 6TiSCH may not be suitable for large-scale WMNs, such as a CG-Mesh or Wi-SUN, in which the network environment often changes.

Typically, bandwidth allocation is a solution for distributed scheduling in wired network, such as GPON or EPON. In such examples, the controller often creates different buffer queues for different classes, and it sends out with specific speed per predefined schedule accordingly.

In general, for each central device, the radio access right is exclusive at one moment, which means only one neighbor can send packets to the central device at any one moment. Therefore, for all neighbors, the bandwidth resource could be treated as the respective portion of a unit time duration, as shown in FIG. 3, below.

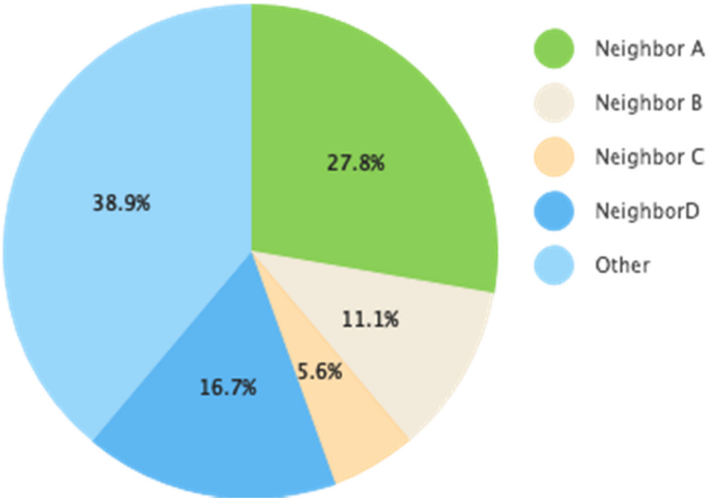
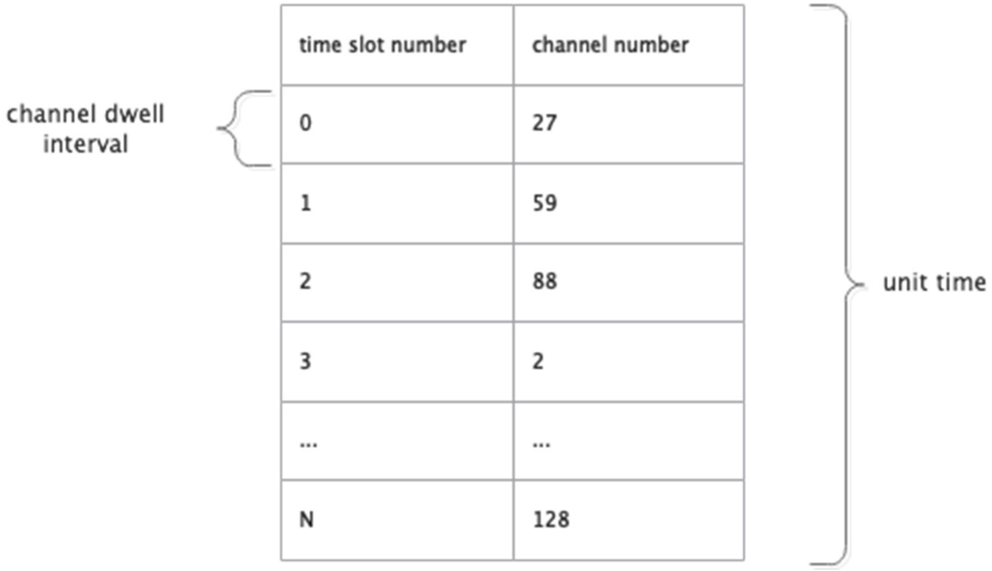


FIG. 3

As shown in FIG. 3, if the central device is aware of the probable duration of each neighbor in one unit interval, it could allocate the required portion of bandwidth to them to avoid the foreseeable collisions. There are some known functions to predict the traffic for neighbors.

In some TSCH based networks, such as CG-Mesh or Wi-SUN, frequency hopping (i.e., channel hopping) is applied, which is a method of exchanging radio signals by rapidly switching a carrier among multiple available channels using a pseudo-random sequence known to both the transmitter and receiver. There are many channel hopping functions could be used, such as TR51 and so on, all available channels could be active once certainly and only once per round.

If it is know that each channel has the same dwell interval, it is possible to define a piece of unit time as being equal to the sum of all available ucast channel dwell intervals, also known as SlotFrame in TSCH (e.g., if there are totally N channels in use for a central device, the unit time will be N*dwell_interval). This is shown in FIG. 4, below.



time slot number	channel number
0	27
1	59
2	88
3	2
...	...
N	128

FIG. 4

In addition, as mentioned above, the central device can predict the throughput for each neighbor in a next unit time period, thus the central device could determine how many time slots are required. For instance, in FIG. 4, neighbor A needs 38.9% of the unit time in the coming slot and the central device could determine the time required by neighbor A. Furthermore, this information could be translated into channels. For example, if there are 129 channels in use (i.e., unit time = 129*dwell_interval), neighbor A will leverage about 51 channels (i.e., 51*dwell_interval). Therefore, the central device could figure out all required portions for all neighbors that want to transmit in the next slot (i.e., children and parent). Presented herein are techniques to inject the calculated channel quantity into a DIO message and then spread this information in the neighborhood when the DIO trickle timer is triggered. The DIO message carries all neighbors' portion schedules with their IDs (i.e., MAC address or eui64). Once a neighbor receives this DIO message, it needs to save its own portion. This processor is generally shown in FIG. 5, below.

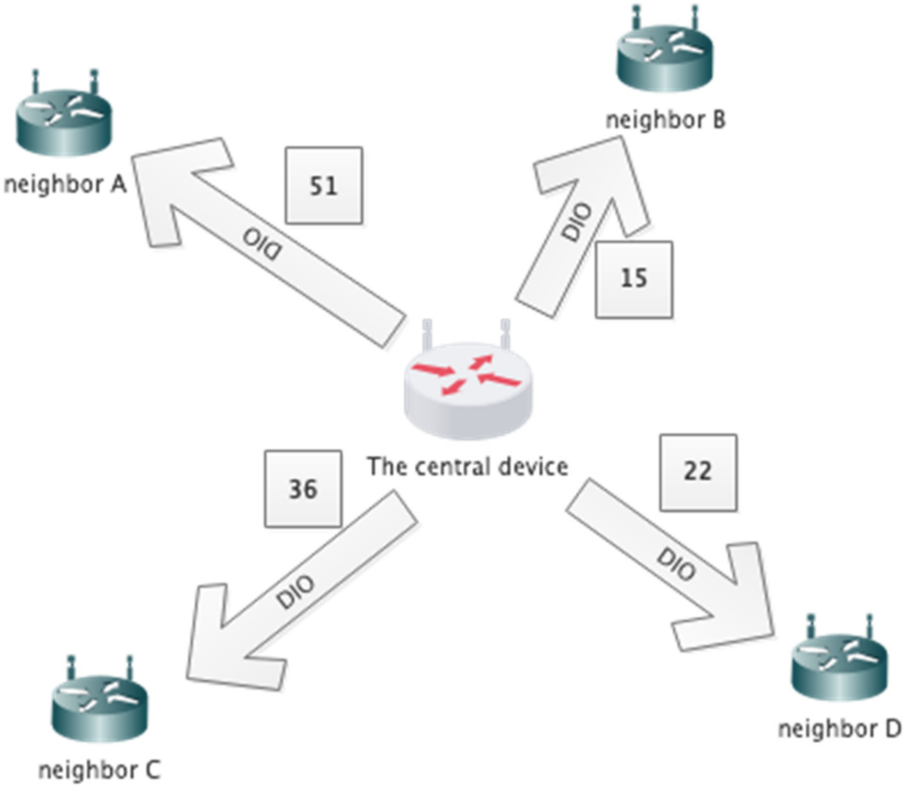


FIG. 5

One exception is the parent of the central device, because it always drops the DIO from its children (i.e., filter by RANK). As such, the techniques presented herein propose the use of a fixed downward bandwidth allocation for the parent node, such as 10% (i.e., channel quantity is 13 per fig 8). The techniques presented herein also propose to use neighbors' ID (e.g., MAC address) as a seed to generate a respective channel list. For example, neighbor A could use 51 channels, so it generates an available channel list CHL[51] = {1, 7, 9, ..., 113} by using a specific function F(x), as also do the other neighbors. F(x) is a pseudo random generator function, which supports to pick up "m" elements from a given set (has "n" elements and $m < n$). This ensures that neighbors will use different channels as much as possible, which could reduce a lot of potential collisions. This is generally shown in FIG. 6, below.

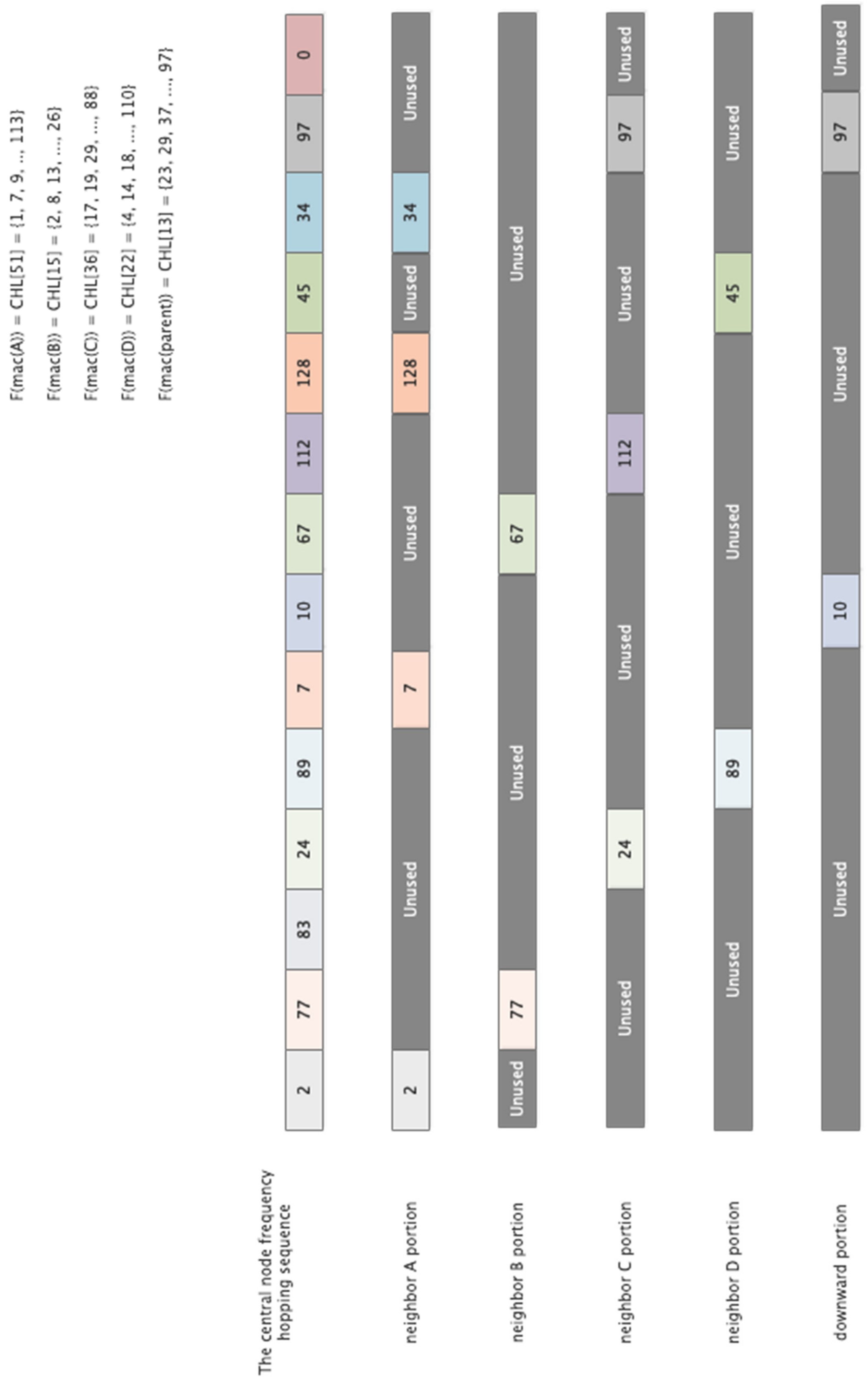

F(mac(A)) = CHL[51] = {1, 7, 9, .., 113}
F(mac(B)) = CHL[15] = {2, 8, 13, ..., 26}
F(mac(C)) = CHL[36] = {17, 19, 29, ..., 88}
F(mac(D)) = CHL[22] = {4, 14, 18, ..., 110}
F(mac(parent)) = CHL[13] = {23, 29, 37, ..., 97}
The central node frequency hopping sequence
2
77
83
24
89
7
10
67
112
128
45
34
97
0
neighbor A portion
2
Unused
7
Unused
128
Unused
34
Unused
neighbor B portion
Unused
77
Unused
67
Unused
neighbor C portion
Unused
24
Unused
112
Unused
97
Unused
neighbor D portion
Unused
89
Unused
45
Unused
downward portion
Unused
10
Unused
97
Unused

FIG. 6

In addition, for CG-Mesh or Wi-SUN, the coefficient of channel utilization is not full load or overload, but rather usually idle (e.g., if the available channels of one central device is 129, neighbors may use only 20 channels while the other 109 channels are idle). Therefore, the techniques presented herein propose that the central device allocates more redundant channels for each neighbor (e.g., if neighbor A just needs 10 channels, the central device can instead assign 50 channels to it rather than 10).

Another new problem is that assigned channels may not meet the requirements for a neighbor. For example, neighbor A may need 60 channels, but is only allocated 50 channels due to limited resources, or neighbor A is allocated 60 available channels, but it has several unexpected re-transmissions, so it needs to extend some channels. However, the preferred central device uses up all bandwidth resources, because, if not, neighbor A should get enough channels. As such, an improved technique is for neighbor A to turn to one of its candidates. In accordance with the techniques presented herein, each central device has a guest account, which could have some reserved channels portion for exceptional access, e.g., 5%. The potential children could leverage this backup portion to alleviate the suffering of concurrent traffic. This is generally shown in FIG. 7, below.

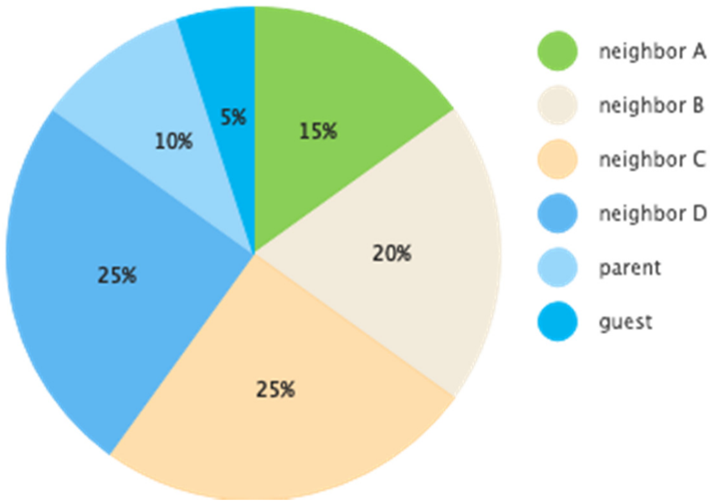


FIG. 7

In certain examples, the central device adjusts the bandwidth allocations per DIO round according to dynamically calculated result. For the ratio of downward and guest portions, it is proposed that they could be configured from the SDN controller as a policy. The SDN controller could adjust the ratio on-demand in practice to optimize the network performance. This is generally shown in FIG. 8, below.

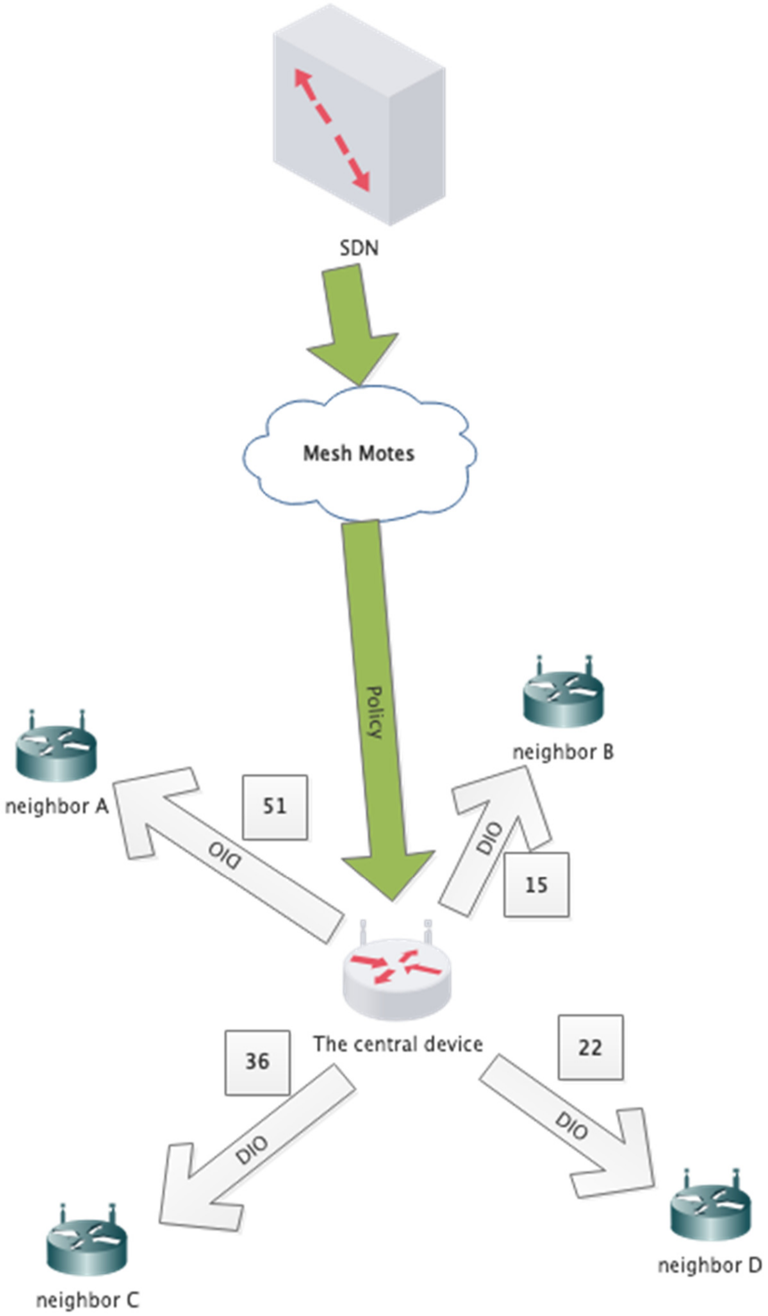


FIG. 8

The techniques presented herein are configured to control disordered competitions among children in a Connected Grid Mesh (CG-Mesh) or a Wireless Smart Utility Network (Wi-SUN), rather than 6Tisch. Both protocols are different from 6Tisch, although they all obey channel-hopping, RPL and IEEE802.15.4 rules. Unlike a deterministic network solution (e.g., 6Tisch), nodes in CG-Mesh or Wi-SUN are greedy and are used to transmit packets immediately as long as their TX queue is not empty. This kind of autonomous WMN is feasible and easy to manage in most cases and is even in certain cases, more efficient than 6Tisch because it does not spend extra overhead on controlling the behavior of children. However, such techniques experience problems with concurrent traffic.

A deterministic network, such as 6Tisch, could provide answers to the above problem. Therefore, the techniques presented herein borrow ideas from DetNet. In a 6Tisch solution (i.e., MSF section 5.1), every child needs to exchange traffic information with its preferred parent to determine whether to add/remove/relocate cells for transmissions, which is not suitable for large-scaled WMN (like CG-Mesh/Wi-SUN) due to many overhead for management. Therefore, the techniques presented herein attempt to find a simple way to reduce concurrent conflicts while not significantly increasing network management overhead.