November 29, 2018

IMPROVING THE PERFORMANCE OF DATA STREAMS IN LOSSY AND LOW-POWER NETWORKS

Wenjia Wu

Nan Yi

Feiliang Wang

Wenchuan Ji

IMPROVING THE PERFORMANCE OF DATA STREAMS IN LOSSY AND LOW-POWER NETWORKS

AUTHORS:

Wenjia Wu
Nan Yi
Feiliang Wang
Wenchuan Ji

ABSTRACT

Presented herein are techniques to enhance the performance of data streams in Connected Grid Mesh (CG-Mesh) networks.

DETAILED DESCRIPTION

Currently in Connected Grid Mesh (CG-Mesh) networks, channel hopping and Carrier Sense Multiple Access (CSMA) are used to avoid collisions between different nodes. These techniques work well if packet generation is only rare and random. However, in certain scenarios, the data stream(s) associated with one or more nodes may be stable.

For example, consider the situation in which many streams are sent to the same parent node. CSMA is not a viable solution in such situations because there is a need for back-off with every packet. In a worst case scenario, hidden node may become problematic and cause a high loss rate.

Data streams may need a high, continuous bandwidth. Channel reservations may be suitable in such situations (since there is no wasted bandwidth) and, in the reserved periods, there are fewer collisions. However, there currently is no suitable method to utilize channel reservations in current CG-mesh networks. As such, proposed herein are practical techniques for handling such issues.

In general, there are at least three (3) main problems that need to be resolved. These three problems are:

1. Proper slots to allocate for the data stream,
2. Latency and QOS assurance, and

Published by Technical Disclosure Commons, 2018

3. High efficiency channel reservation transaction and synchronization.

In general, a channel is only reserved when there are data streams that need to be sent. After a data stream has been sent, the channel should be freed as soon as possible. For normal random data packets, a vacant slot which is not occupied can usually be found and at least 10% - 20% of channel resources should be reserved for normal data packets. Figure 1, below, illustrates the basic structure of a channel reservation.

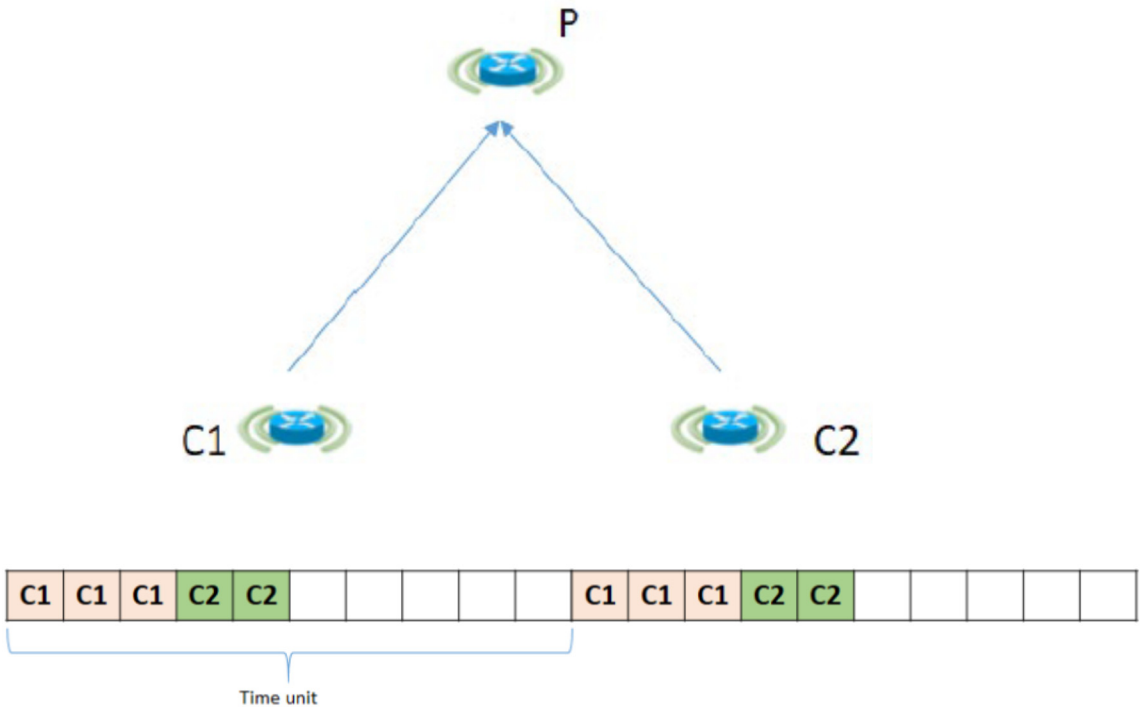

Figure 1

In the example of Figure 1, C1 and C2 are two children nodes of Parent node P. Nodes C1 and C2 would each like to send data streams to the parent node P. As such, they would each occupy some channel resources to help send the data stream fast and smoothly. Accordingly, as shown in Figure 1, each of the child nodes C1 and C2 will try to reserve some time slots for its respective data stream. This would be duplicated for each time unit.

Allocation of Proper slots for Data Streams

It is to be appreciated that it is important to allocate proper time slots to one data stream, where less slots than is required by a node is insufficient and too many slots is a waste of bandwidth. In a common scenario, as shown in Figure 2, below, input data comes from an application (app) and it is sent out through radio-frequency (RF).

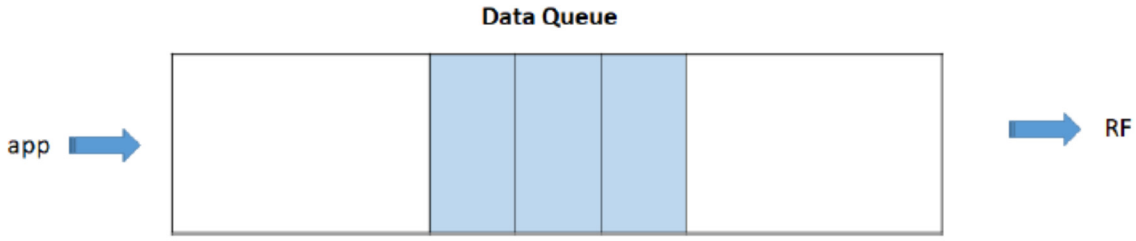

Figure 2

As shown in Figure 2, there is a data queue to buffer the data packet. When the bandwidth is insufficient, the average number of data packets in the queue will expand. Otherwise, the average number of data packets will decrease or remain at a low level. Based on this, the following steps could be performed to dynamically allocate the slots.

1. First, the amount of bandwidth that needs to be allocated is unknown. As such, the process begins with a slow start (e.g., 2 or 3 slots, initially).
2. After monitoring the data buffer queue for a period of time, it is possible to dynamically expand and shrink the time-slots allocated based on the average number of packets in the data queue.

Latency and QOS assurance

Latency

Certain data streams are sensitive to latency and a method is needed to process such streams. This requires a tradeoff between latency and efficient usage of bandwidth. Long series slots will cause possibly high latency, but the waste due to frame size issues is

relatively severe if the slots are short. Consider the situation shown in Figure 3, below, where data streams are received from an app for C1 at t1, t2, t3.

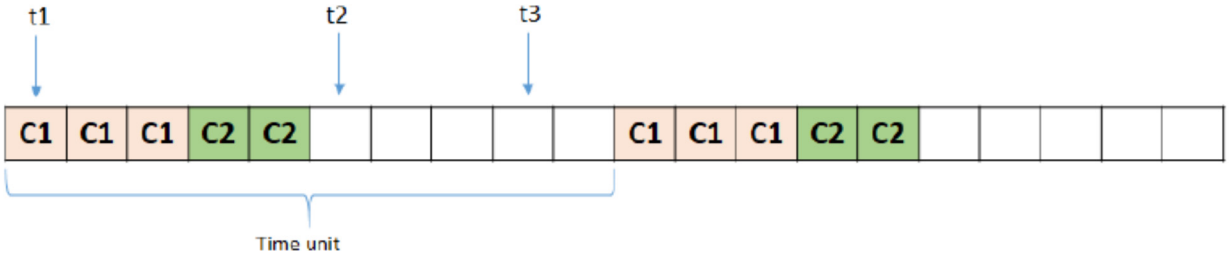

Figure 3

In Figure 3, high latency will occur at t2 and the latency is about half of the time unit. Therefore, if the bandwidth is not limited, the best choice is to separate the long timeslots, as shown below in Figure 4, to decrease the average latency. In this case, the latency for the packet at t2 is very low.

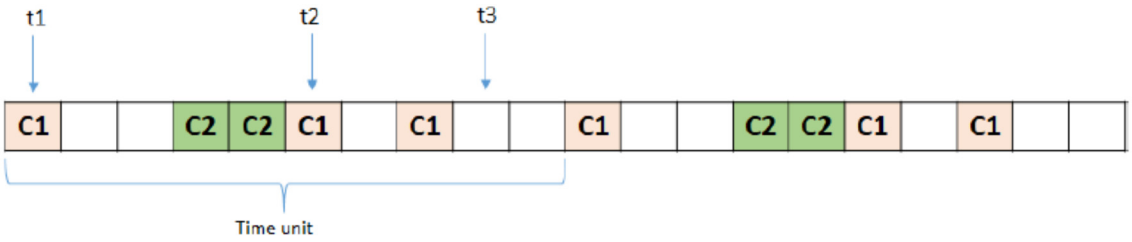

Figure 4

A determination of whether or not high latency is present could be made by monitoring the data message queue. If the average number of data messages is steady, but the maximum number fluctuates severely, then the maximum number of message queue is considered high. It is noted that some packets will have high latency issues.

QoS

A technique for distinguishing the normal random packets from data stream packets is also required. It could be possible to separate the packets based on QOS. For example, there may be a special QoS level for data stream packets that is set by the app via DSCP. After the special QoS is set, the stream could be set up. A higher QoS level means high priority. When all bandwidth is allocated, higher QoS streams could preempt the slots occupied by the lower QoS streams. After occurs, the parent node advertises the information to the node whose slots are preempted.

High Efficiency Channel Reservation Transaction and Synchronization

The above section describes how to allocate/de-allocate time-slots. An efficient way to synchronize among the neighbor nodes is also needed. In one example, the message format is based on "6top", with the addition that two bitmaps will be sent to each node. One bitmap is the occupied slot of the parent node and it contains the slot occupied by each child, slot reserved for the parent node itself, and the vacant part for normal random packets. The second bitmap is the reserved slot for the child node itself.

When some node has a data stream and tries to allocate the time-slots, the parent node will provide the two bitmaps with the node. If any updates, the parent node will attach the bitmaps as an IE in ACK to update the child's parent node.

In addition to sending two bitmaps, each time slot allocation contains a lifetime part. The lifetime could be set based on the current data message that needs to be sent. After the time-slot expires, the time slot will be automatically deallocated. If there are more packets, the lifetime could be extended. The advantage of using lifetimes is that there is no need for redundant messages to deallocate the slot.

Additionally, it is possible to dynamically stream de-allocated messages contained in the packet as an IE. When the data stream is continuous, there is no need for an independent command. Instead, the message will be contained in the following stream packet as an IE.