January 02, 2019

Peer Node Eviction in Wireless Mesh Networks

Javier Cardona

Peer Node Eviction in Wireless Mesh Networks

ABSTRACT

Wireless mesh networks that include large numbers of peer nodes pose problems for routing frameworks. Multicast messages used for peer discovery can get dropped and synchronization of data between peer nodes does not scale to large numbers of peer nodes. A peer eviction technique for selective eviction of peer nodes is described. The technique can be used to limit the number of nodes for a wireless mesh network. Peer nodes are selected for eviction based on node-specific metrics. The technique ensures stability of the network by restricting the eviction rate and by adjustments to signal thresholds used to evaluate nodes for eviction.

KEYWORDS

802.11s; mesh network; routing framework; peer eviction; eviction rate; RSSI

BACKGROUND

Wireless mesh networks are utilized in many contexts. For example, IEEE 802.11s defines how wireless devices can interconnect to create a WLAN mesh network. Such mesh networks are typically used for relatively fixed (not mobile) topologies and wireless ad hoc networks. In mesh networks, each node of the network has a number of peer nodes, e.g., that are within wireless communication distance from the node. Routing frameworks are built on top of such physical networks.

However, networks with a large number of peers can lead to problems in the use of routing frameworks. For example, some routing frameworks rely on periodic multicast messages to discover peers. The messages include the list of known neighbor nodes. If the list of neighbor nodes is large, such messages become larger than the maximum transmission unit (MTU) for 802.11 frames and get dropped. One possible solution for this is to use 802.11s beacons

In some routing frameworks, message queues are used to synchronize data (e.g., key-value pairs) between peer nodes. Each additional (Nth) peer node increases the number of message queue (MQ) connections by (N-1). Further, as the mesh size increases, the number of peer nodes can exceed the maximum number of 802.11s peers supported by the kernel/driver, posing scalability problems which cannot be addressed by beacons. Therefore, mechanisms to limit the number of peer nodes for routing frameworks for mesh networks such as 802.11s networks are necessary.

DESCRIPTION

A possible solution is to limit the number of peer nodes that are addressable via the routing framework, while leaving the underlying physical network as is. However, such an approach results in two categories of peers: a first category known to the routing framework, and a second category that is not part of the routing framework. Since the second category nodes cannot be used for routing, such nodes are essentially deadweight in kernel memory. Such categorization can also create confusion for network operators. The nodes that do not participate in routing still generate unproductive L2 traffic, e.g., receive and respond to 802.11s path requests, send peer link establishment frames, negotiate security, etc.

Therefore, while this solution can address the problems posed by large numbers of peer nodes in the physical network, a solution that limits 802.11s peer nodes in the network is likely to perform better. Some simple approaches to limit the number of 802.11 peer nodes are:

1. **Limit 802.11s peers by RSSI:** 802.11s networks already include received signal strength indicator (RSSI) based threshold. In a 802.11s mesh network, the promotion of a node to a peer is limited to neighbor nodes whose beacon frames are received with signal strength above a RSSI threshold. However, while this can certainly reduce the number of peers, it does not provide a guarantee that the resulting number of peers will be sufficiently low. For instance, a dense mesh with many nodes near one another can include a large number of peer nodes that meet the RSSI threshold.
2. **Limit 802.11s peers by configuration limit:** The maximum number of peers supported by the kernel can be set via a command, e.g., to a default value of 32 or other value. Neighbor nodes are promoted to peer status on a first-come-first-serve basis. However, such an approach can result in a suboptimal list of peers since nodes with poor signal can become peers before other nodes with better signal.

A dynamic peer eviction method that addresses these problems is described below, with reference to the flowchart on the last page of this document.

Method initialization

At block S10, a target number of peers (target_peers), maximum eviction rate (max_eviction_rate), and minimum number of peers (min_peers) are defined.

At block S15, the RSSI threshold for a node to become a peer is set, and the method is initialized by setting peer count to zero.

At block S20, it is determined if a new peer connection has been established. If yes, the method continues to block S25 (and subsequent blocks, illustrated in yellow) to implement peer eviction. If no, the method continues to block S80 (and subsequent blocks, illustrated in blue) to determine whether a peer has disconnected.

Peer eviction

At block S25, it is determined if the peer count exceeds the target. If the target peer count is not exceeded, no peer is evicted. If a peer is to be evicted, at block S30, it is determined if the last eviction was within an eviction period indicated by the maximum eviction rate. If the eviction was within the period, no peer is evicted. Else, block S35 is performed to identify a peer based on metric. Peers are identified in increasing order of metric.

At block S40, it is determined if the identified peer is connected to gate. If the peer is connected to gate, block S45 is performed to determine if the current number of peers is equal to the minimum number of peers. If the number is not equal (e.g., the current number of peers is greater than the minimum), block S50 is performed to evict the identified peer.

After evicting the identified peer, at block S55, the RSSI threshold is increased. For example, the RSSI is increased to the RSSI of the evicted peer. At block S60, it is determined if more peers are to be evaluated for eviction, in which case block S35 is performed; else, block S20 is performed to detect if a new peer connection has been established.

Peer disconnection

At block S80, it is determined if a peer node has disconnected. If a node disconnection is detected, at block S85 it is determined if the peer count post disconnection is less than the target number of peers. If the peer count is less, at block S90, the RSSI threshold is reduced.

Advantages

This method ensures that a minimum number of peers are always included in the peer list. The method also allows each node to peer with the best peer nodes from the nodes available in wireless communication range. If the number of peers in range is less than the target number (block S25), peer eviction is not performed, thus accepting all peer nodes regardless of signal strength. With excess peers, a peer is identified for eviction (in sorted order of metric, resolving collisions by random selection), and the RSSI threshold is updated to the RSSI of the evicted peer. This ensures that the evicted peer is not able to re-peer immediately after eviction, unless there is a change in signal condition, or until the threshold is lowered again.

The updated RSSI threshold only applies to the peer link establishment exchange. Thus, the increase in the RSSI threshold does not affect established peers, even if their RSSI is below the threshold.

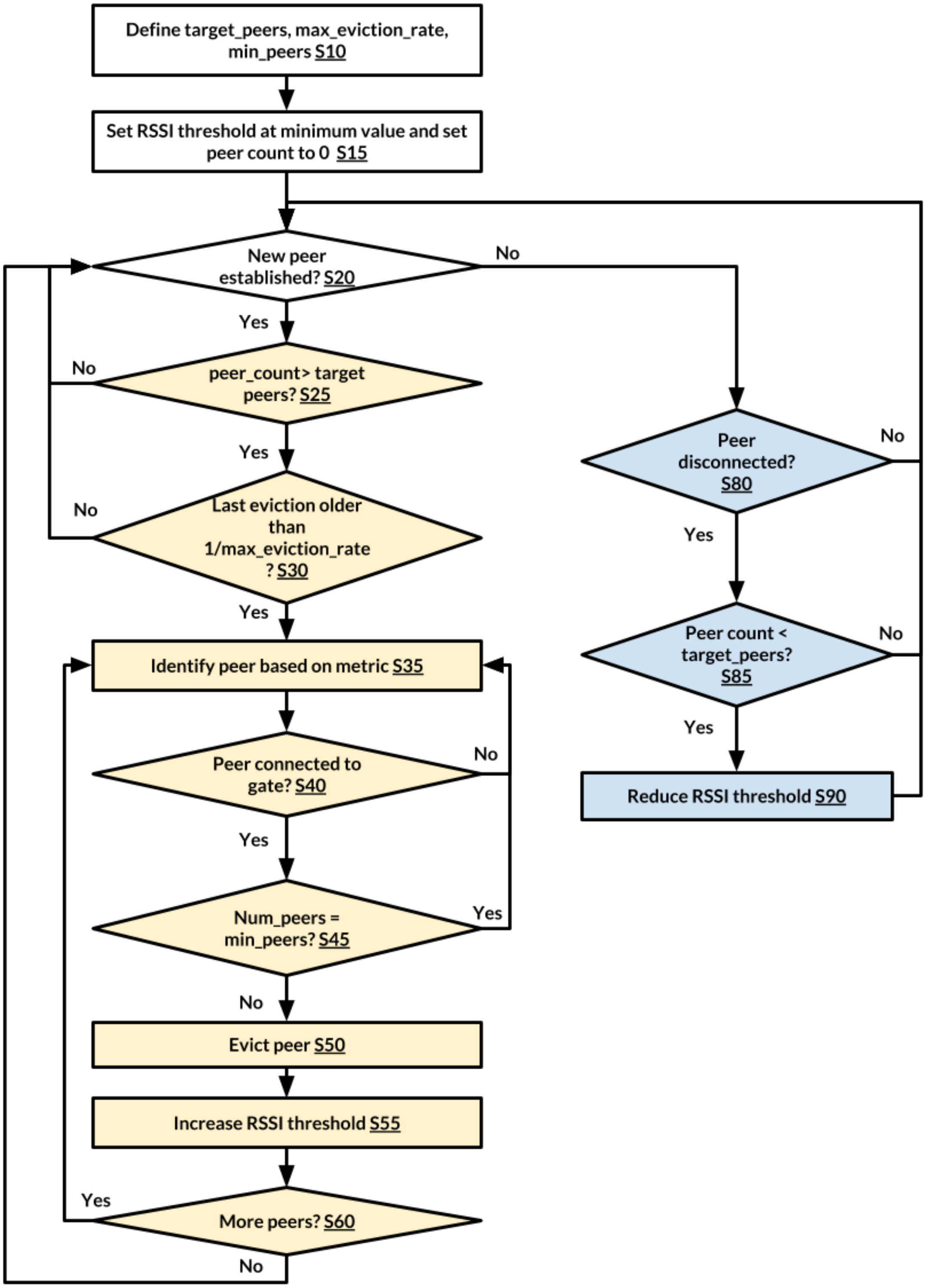
This method of peer eviction is stable even if RSSI variance from peers is high since the RSSI threshold is increased monotonically in case of evictions. If the number of peers in range greatly exceeds the target, the RSSI threshold keeps on increasing. The RSSI threshold is decremented only when a peer link is terminated for a reason other than eviction as illustrated in blocks S80-S90. The reason can include, for example, the peer moving out of range or being switched off. Further, the eviction method limits the rate of changes to the peer table by specifying the maximum eviction rate and limiting the number evictions accordingly (block S30).

The method can be performed periodically, e.g., at a similar rate as the periodic multicast messages. Alternatively, the method can be triggered based on occurrence of events such as teardown of an existing peer connection, or establishment of a new peer connection. Peer management can be performed entirely in the kernel.

CONCLUSION

A peer eviction technique for selective eviction of peer nodes is described. The technique can be used to limit the number of nodes for a wireless mesh network. Peer nodes are selected for eviction based on node-specific metrics. The technique ensures stability of the network by restricting the eviction rate and by adjustments to signal thresholds used to evaluate nodes for eviction.

FIGURE


Define target_peers, max_eviction_rate, min_peers S10
Set RSSI threshold at minimum value and set peer count to 0 S15
New peer established? S20
No
Yes
No
peer_count> target peers? S25
Yes
No
Last eviction older than 1/max_eviction_rate ? S30
Yes
Identify peer based on metric S35
Peer connected to gate? S40
No
Yes
Num_peers = min_peers? S45
Yes
No
Evict peer S50
Increase RSSI threshold S55
Yes
More peers? S60
No
Peer disconnected? S80
No
Yes
Peer count < target_peers? S85
No
Yes
Reduce RSSI threshold S90