September 25, 2018

DYNAMICALLY MITIGATING BOTTLENECK EFFECT TO GUARANTEE QUALITY OF SERVICE IN LOW-POWER AND LOSSY NETWORKS

Mingyu Xie

Jianfeng Mao

Xueying Yang

Xiang Fang

DYNAMICALLY MITIGATING BOTTLENECK EFFECT TO GUARANTEE QUALITY OF SERVICE IN LOW-POWER AND LOSSY NETWORKS

AUTHORS:
Mingyu Xie
Jianfeng Mao
Xueying Yang
Xiang Fang

ABSTRACT

Techniques are described herein for providing an intelligent and dynamic routing policy for Quality of Service (QoS) based on Routing Protocol for Low-Power and Lossy Networks (RPL) Directed Acyclic Graph (DAG). This helps mitigate the bottleneck effect in a connected grid mesh by forecasting the capacity of the routing path. Each sender device may be able to forward packets based on QoS requirements to the proper next hop before RPL DAG updates by Expected Transmission Count (ETX) change. With this approach, the QoS of latency sensitive or low packet loss tolerance services can be better satisfied in the connected grid mesh network.

DETAILED DESCRIPTION

A connected grid mesh for smart grids may support various applications (e.g., Advanced Metering Infrastructure, Distribution Automation, etc.). In this type of Low-power and Lossy Network (LLN), utility enterprises usually aim to conserve precious bandwidth resources while requesting the minimum delay.

Routing Protocol for LLNs (RPL) is used for building up Destination Oriented Directed Acyclic Graph (DODAG) as the packet routing path in LLNs. In the connected grid mesh implementation, each device is capable of keeping three as the maximum number of parents in a Destination Advertisement Object (DAG). The upward traffic first goes to the primary parent and if all retransmissions fail, it goes to secondary parent. The downward traffic based on Destination Advertisement Object (DAO) priority also first goes to the primary parent, and then tracks the secondary parent if all the attempts to primary parents fail in RPL storing mode.

One device may have several child devices in DODAG. In this case, the common ancestor becomes the bottleneck when its successors have concurrent packet transmission

Published by Technical Disclosure Commons, 2018

or the ancestor also has a packet to send to other devices. The Expected Transmission Count (ETX) calculation on the child devices may help them switch the primary and secondary parent. However, this is after packet loss or retransmission. Before that, retransmissions on the child device occupies the buffer resources and causes packet drops on the child device itself, which makes the child device become another bottleneck for its successor. Particularly for the latency sensitive or low packet loss tolerance services, the current implementation does not satisfy Quality of Service (QoS) requirements.

To address these challenges, the techniques described herein provide dynamic routing optimization by switching traffic path proactively based on the indication from its parents. This may guarantee that the latency sensitive traffic and low packet loss tolerance service can be forwarded on a better path before the device detects the ETX changes. This introduces a method for proactively mitigating the bottleneck effect to guarantee QoS and limit congestion in RPL-based LLNs.

These techniques include three components. The first component is that the device estimates and forecasts the path capability of forwarding per QoS requirements. The second component is that the device sends the path capability of forwarding in an acknowledgment (ACK) frame when it receives a unicast packet. The third component is to dynamically choose the routing path per packet Class of Service (CoS) based on the path capability of the forwarding device.

A bitmap is defined as a QoS policy tag to indicate the forwarding capability for different QoS patterns on itself. It should consider the possible factors that may shortly impact packet forwarding, such as Central Processing Unit (CPU) utilization, packet buffer vacancy, queue size, and congestion error patterns on the device. A set of thresholds may be set for the device to be aware of how busy it is for processing the certain type of QoS packet. For example, when the packet buffer vacancy is less than 20%, the QoS policy tag for high priority packets may be set to “False.”

Each device calculates the path QoS policy tag by a bitwise AND its own tag with the one received from the primary parent. The path QoS policy tag is carried in the unicast ACK frame to indicate different QoS traffic forwarding capabilities on the routing path. The device maintains the QoS policy tag for possible routing paths. Every time a new unicast packet is received, the device updates the tag in the ACK if necessary.

A time window may be predicted based on the receiver's resource status and also indicated in the ACK frame to let the sender know how long the busy situation might remain. The forecasted time window may be calculated by the remaining packet and data rate. The sender may carry the maximum length of the window in the path in the ACK.

Based on the indication, the sender may determine the proper next hop for the next following packets with certain QoS requirements. After the time window finishes, the sender may send the packets to the original next hop. For example, if the QoS policy tag indicates some device on the primary parent path is too busy to handle high priority packet, but the backup parent's path is good enough, the sender may send the high priority packets to its backup parent as the next hop.

If the sender receives the negative tag on all possible routing paths, it may choose to lower the data rate and follow the original path built by the RPL DAG. The downward traffic path can be optimized by the same mechanism when the downward route has a different next hop to the same destination in RPL storing mode.

Figure 1 below illustrates a basic scenario with upward traffic. A/B is the primary parent for C/D. C/D is the primary/backup parent for E. F has upward traffic, the original traffic path based on RPL DAG is F->E->C->A->CGR. Since F has only one parent E, the mechanism cannot work for F. E has two paths for relaying traffic from F, and E follows steps to dynamically switch high priority packet to the more reliable path.

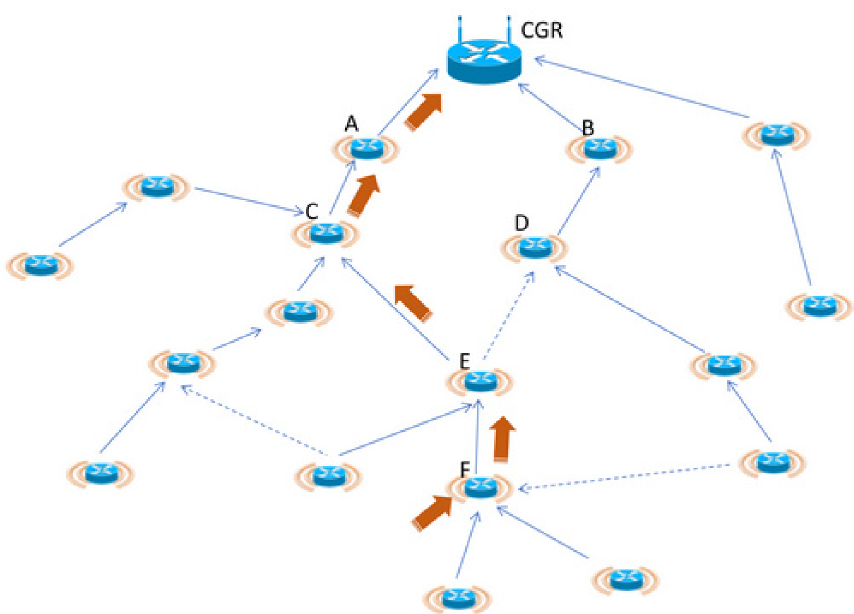

Figure 1

At the first step, the resource occupation (e.g., CPU utilization) on device A is 60%, and has enough resources to forward traffic. It sets the QoS policy tag bitmap to 111 (1 means forward traffic, and H means high priority traffic). With this same rule, the forwarding capability bitmap on C (80%) is 011, which means C is too busy to deal with high priority traffic but can handle medium and low priority traffic. At the second step, E sends the upward high priority traffic to C.

Figure 2 below illustrates the first and second steps. The QoS policy tag for Path X is 011 (111 and 011), and for Path Y is 111. C adds the Path X QoS policy tag 011 into its ACK message to inform E that the current traffic path is too busy to forward high priority traffic for next 500ms.

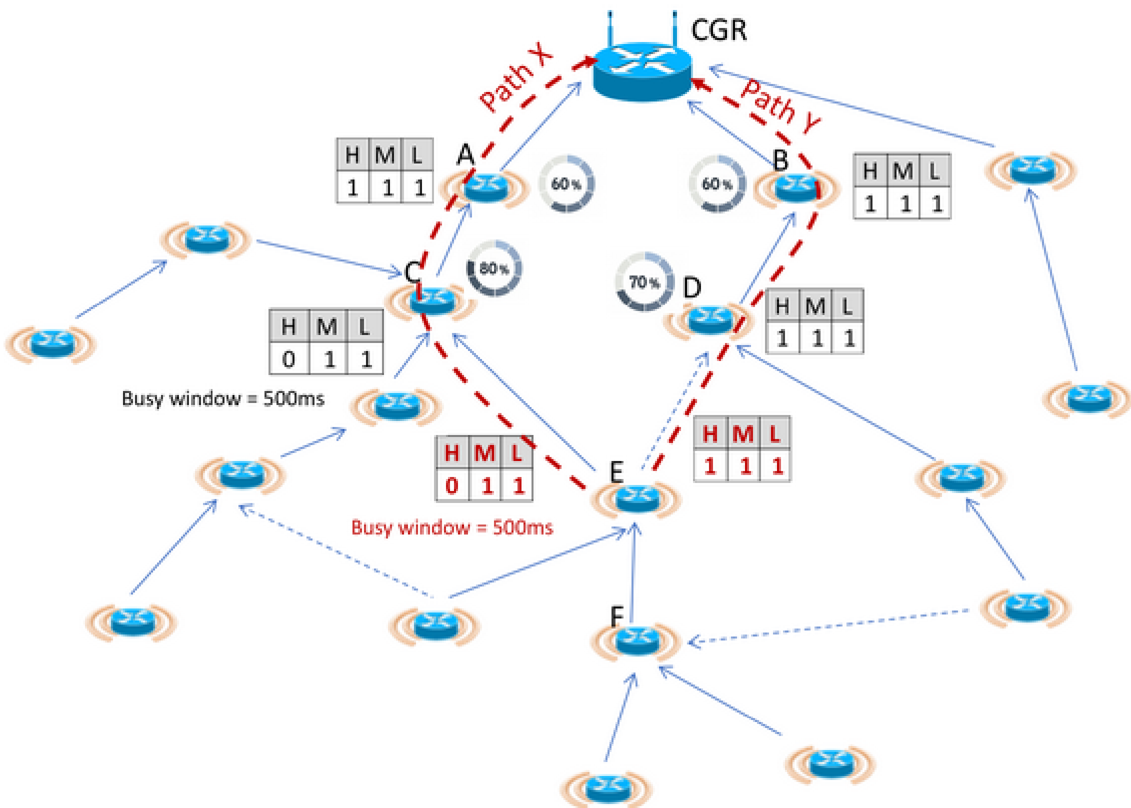

Figure 2

At step three, during the next 500ms, the high priority traffic is forward to D following Path Y. Medium and low priority traffic may still be routed via original Path X. Thus the potential congestion on C is released and high priority traffic is routed via a better path while the medium and low priority packets are still forwarded to the original Path X to C. At step four, after the 500ms time window has expired, E reverts high priority traffic to C and follows Path X as the RPL DAG.

Figure 3 below illustrates the third and fourth steps.

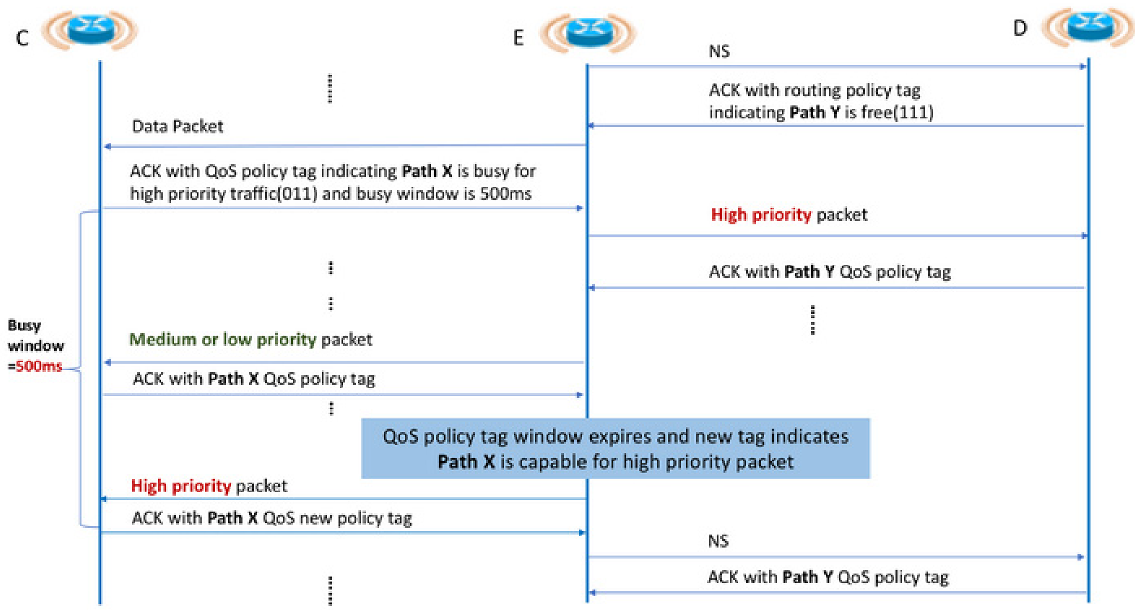

Figure 3

In summary, techniques are described herein for providing an intelligent and dynamic routing policy for QoS based on RPL DAG. This helps mitigate the bottleneck effect in a connected grid mesh by forecasting the capacity of the routing path. Each sender device may be able to forward packets based on QoS requirements to the proper next hop before RPL DAG updates by ETX change. With this approach, the QoS of latency sensitive or low packet loss tolerance services can be better satisfied in the connected grid mesh network.