

Do Not Lose Bandwidth: Adaptive Transmission Power and Multihop Topology Control

Hyung-Sin Kim and Saewoong Bahk
Department of ECE and INMC

Seoul National University
Seoul, Republic of Korea

Email: {hskim, sbahk}@netlab.snu.ac.kr

Jeongyeup Paek
School of CSE

Chung-Ang University
Seoul, Republic of Korea

Email: jpaek@cau.ac.kr

David E. Culler
Department of EECS

University of California, Berkeley
Berkeley, CA, USA

Email: culler@berkeley.edu

Abstract—We show that a multihop wireless network can achieve better bandwidth and routing stability when transmission power and routing topology are jointly and adaptively controlled. Our experiments show that the predominant ‘fixed and uniform’ transmission power strategy with ‘link quality and hop distance’-based routing topology construction loses significant bandwidth due to hidden terminal and load imbalance problems. We design an adaptive and distributed control mechanism for transmission power and routing topology, *PC-RPL*, within the standard RPL routing protocol. We implement *PC-RPL* on real embedded devices and evaluate its performance on a 49-node multihop testbed. *PC-RPL* reduces total end-to-end packet losses ~ 7 -fold without increasing hop distance compared to RPL with the highest transmission power at heavy load, resulting in 17% improvement in aggregate bandwidth and 64% for the worst-case node.

Keywords-Low-power Lossy Network (LLN), RPL, IPv6, Bandwidth, Load Balancing, Transmission Power, Routing

I. INTRODUCTION

The subtlest aspect of routing in low-power and lossy wireless networks (LLNs) is topology formation. In modern routing protocols, such as RPL (IPv6 routing protocol for LLNs) [1] and CTP (collection tree protocol) [2], the goal of this process is to form a DAG (directed acyclic graph) rooted at one or more border routers, typically connected to LAN or WAN networks and thereby part of a private or public Internet. Each node discovers neighbors through communication events and computes certain statistics, such as hop count or expected transmission (ETX) [3], to select a small subset of neighbors that are closer to the roots to serve as parents [1][2]. The dominant traffic pattern is then generating and forwarding packets through parents toward border routers and beyond. All aspects of this process, link capacity, neighbor table size, routing table size, and queue size, are severely constrained.

Hyung-Sin Kim is now at the EECS department of UC Berkeley, CA, USA (email: hs.kim@berkeley.edu).

This work was supported in part by the ICT R&D program of MSIP/IITP Korea [B0717-16-0026], in part by NRF-2016R1A6A3A03007799 and NRF-2014R1A1A2056626 through the National Research Foundation of Korea, and also in part by NSF CPS-1239552.

The wireless network topology is primarily determined by a host of external factors, e.g., physical placement of the nodes, presence of obstacles, media attenuation, and multipath effects. All that a node can control is its transmission power and the logic it uses to compute statistics and apply thresholds in the routing topology formation process. Many studies have considered varying transmission power to increase available local bandwidth through spatial multiplexing, often in theoretical models or simulations [4]. Basically, the lower the transmission power, the sparser and more localized is the connectivity graph (i.e., spatial multiplexing). However, for collection patterns, which dominate in practice, it would be expected to bring no value, since every packet that is generated needs to arrive at a border router eventually; collection points are the bottleneck. Maximizing transmission power tends to maximize link reliability and minimize routing distance. Uniform transmission power setting for all nodes is also commonplace for simple deployment.

This paper shows, surprisingly, that substantial gains in delivered bandwidth (throughput) and fairness result from dynamic and non-uniform adaptation of transmission power in combination with adaptive topology formation. The gains do not come from spatial multiplexing but from hidden terminal mitigation and load balancing. We study this problem in the context of RPL (a potential building block of Internet of Things) on a substantial, multihop testbed comprised of nodes in wide use for over a decade [5]. Given the limited link rates of LLN, the bandwidth they provide is generally precious, and even more so when they need to deliver heavy traffic as in some of the upcoming large-scale applications, such as smartgrid [6][7]. We develop a new routing protocol *PC-RPL* (Power Controlled RPL) that uses purely local rules in a simple control loop to adapt topology formation and transmission power in concert to improve delivered bandwidth and fairness. In doing so, the rules recognize the interrelated effects of link loss, queue loss, and routing distance and balance them to eliminate loss.

In RPL, topology formation is driven by DIO messages,

distinct from application data packets, so the two forms of communication can easily utilize distinct transmission power settings. Maximum transmission power is used for (infrequent) DIOs to gain the most information about the node neighborhood. The set of candidate parents is pruned or expanded in a manner consistent with time-varying connectivity by adjusting the threshold used in parent selection. To reduce hidden terminal effects at its parent, a node may elect to prune its parent list and select a parent “closer to it” by increasing the threshold and then, adjust its data transmission power while maintaining link reliability. However, these benefits must be balanced against potentially increased path length and path loss. Further load balancing benefits are gained by analogous threshold adjustments vis-a-vis children, in order to cause some of the children of an overloaded parent to shed it in favor of a more lightly loaded, good alternative. Critical to these adaptive mechanisms is identifying how a node can reasonably infer whether loss it observes is due to hidden terminal or queue overload effects.

We develop and evaluate a localized, adaptive topology and transmission control protocol in stages. After defining our empirical methodology in Section II, we study uniform transmission power control with conventional topology formation (RPL) in Section III. This proves the existence of benefits of power control on collection traffic and permits a sequence of observations that lead to our adaptive, non-uniform strategy. It should be noted that the benefits of *a priori* transmission control, while present on any specific deployment, are not directly actionable because the control point depends on particulars of the deployment. If it is to be determined *in situ* by nodes observing particular events and taking particular actions, it is actually more natural to employ non-uniform adaptation, which is also much more effective. Deployments are typically non-uniform.

In Section IV we develop the *PC-RPL* adaptive topology and power control protocol and evaluate its effectiveness in Section V on a 49-node testbed. It provides a 7-fold reduction in end-to-end packet loss rate, resulting in 17% improvement in aggregate bandwidth and 64% improvement for the worst-case node. The control algorithm stabilizes quickly with key adjustments occurring in a minority of the nodes. Section VI discusses related work, and Section VII concludes the paper.

II. SYSTEM MODEL AND EXPERIMENTAL SETUP

To study bandwidth of a wireless multihop network, we configured a testbed environment as depicted in Fig. 1, where 48 nodes and one border router (marked with a star) are in an office environment. Each node is a TelosB clone [5] with an MSP430 microcontroller and a CC2420 radio (IEEE 802.15.4 radio with a maximum transmit power of 0dBm). The border router is connected to a Linux PC through a serial link. Embedded software is TinyOS 2.1.2 with an IPv6/6LoWPAN stack and a RPL implementation, *BLIP*

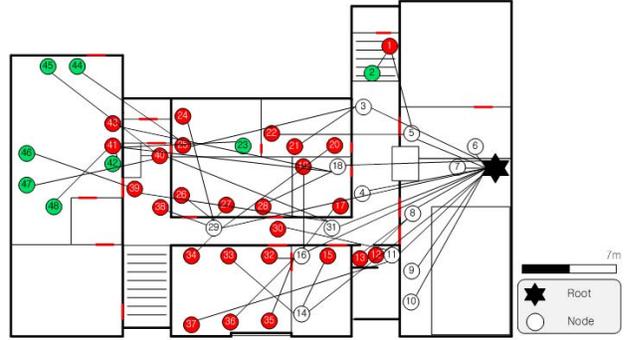


Figure 1. Testbed topology with a snapshot of routing paths given by RPL when using transmission power of 0dBm.

and *TinyRPL* respectively. Each node employs the TinyOS default CSMA with up to 5 link level retransmissions and a transmit queue size of 10 packets. To focus on the impact of transmission power and routing topology, we disabled the use of duty cycling mechanisms [8], which is common in many practical use cases [7][9][10] where LLN routers are wall powered. We also focus on upward traffic from individual nodes to the border router (i.e., data collection).

To evaluate the impact of a network protocol on bandwidth, we generate upward traffic at a rate from 720 packets per minute (ppm) (15ppm/node) to 2880ppm (60ppm/node). Our preliminary experiments with various 2-hop topologies (results omitted due to lack of space) show that these traffic loads can be delivered when multiple branches are effectively used without hidden nodes. In this setup, bandwidth degradation, if any, does not come from wireless capacity but from misbehavior of a network protocol. It should be noted that RPL does aim to support large-scale LLNs comprising hundreds and thousands of nodes [1], such as building automation [11] and industrial applications [12], which require nodes near the border router to deliver heavy traffic. Cisco’s CG-mesh solution for smartgrid is a good commercial example that connects 5000 LLN nodes to a single border router through RPL [7]. In such a large-scale network, one packet generation every ~ 7 minutes at each node leads to traffic load at the bottlenecked border router similar to our test scenario.

III. LIMITATIONS OF UNIFORM TRANSMISSION POWER

In this section, we provide an experimental measurement study of standard RPL while varying transmission power settings uniformly for all nodes.

A. Effect of Traffic Load

Fig. 2 plots relevant performance metrics of RPL with varying traffic load when all nodes use 0dBm transmit power, which is the maximum allowed by the IEEE 802.15.4 standard. Fig. 2(a) shows that RPL successfully delivers 99.7% of the generated traffic when traffic load is light (i.e., 720ppm). This shows that, in our testbed environment, RPL

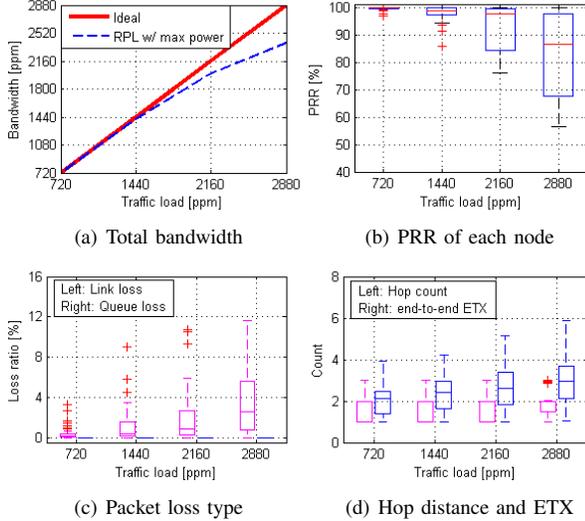


Figure 2. Performance of RPL with varying traffic load when all nodes exploit maximum transmit power (0dBm). RPL suffers from the hidden terminal problem, which not only degrades bandwidth but also churns routing topology.

establishes a reliable routing topology where the quality of each link is good enough to deliver a packet within 5 retransmissions. However, the gap between traffic load and achieved bandwidth increases with traffic load. Specifically, when traffic load is 2880ppm, bandwidth becomes 2400ppm, which means that the network loses 17% of packets. Furthermore, Fig. 2(b) shows that end-to-end packet reception ratio (PRR) from each node becomes significantly unfair as traffic load increases. Under traffic load of 2880ppm, the worst-performing node experiences only 57% PRR, while some nodes still achieve nearly 100% PRR.

What then is going wrong in this network? To figure this out, we divide losses into two types: link loss and queue loss, and plot them in Fig. 2(c). Here, we observe that as traffic load increases, most packet losses occur at links rather than at queues (Queue loss ratio is hard to see in Fig. 2(c) because all values are close to zero.) Further investigation reveals that this link loss comes from ‘hidden terminal problem’, which causes packet collision. As shown in Fig. 2(d) which plots both the hop distance and ETX from each node to the border router, the hidden terminal problem increases ETX with traffic load, while the hop distance remains constant (i.e., transmissions per hop increases). Even though RPL tries to adjust routing topology based on ETX, it fails to provide reliable routes as shown in Fig. 2(b); RPL cannot address hidden terminal problem but simply results in routing topology churn.

B. Effect of Transmission Power

Based on the above findings, we ask the question, “if there is severe hidden terminal problem in the network, can we alleviate it and improve bandwidth by adjusting

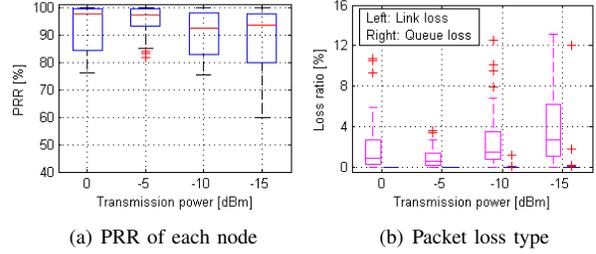


Figure 3. Performance of RPL with varying transmit power at traffic load of 2160ppm (45ppm/node). If all nodes use equal transmit power, simply reducing transmit power is not an effective solution to improve bandwidth.

the transmission power?” To seek an answer, we conduct additional experiments at a load of 2160ppm (45ppm/node) with varying transmit power from 0dBm to -15dBm, on all nodes. Fig. 3 plots performance with this configuration.

First, Fig. 3(a) shows that transmit power of -5dBm provides better PRR performance than 0dBm. The maximum allowed transmit power is not optimal; using a smaller transmit power can improve bandwidth. However, PRR degrades rapidly when transmit power is reduced too much. This provides an initial indication that we need a balance. “Adaptive” transmission power control has the potential to improve bandwidth.

To understand this phenomenon in more detail, Fig. 3(b) divides packet losses into link loss and queue loss. It shows that link loss decreases at -5dBm but increases again as transmit power is reduced further. When using transmit power -10dBm or -15dBm, link loss becomes larger than with 0dBm, despite fewer neighbor nodes. In addition, queue loss increases as transmit power is reduced. Importantly, severe queue losses occur only at a few nodes (1 ~ 2 nodes in Fig. 3(b)). By inspecting the routing topology, we identify the bottleneck nodes as the nodes with large unbalanced subtrees. For example, Fig. 4 depicts a snapshot of RPL’s routing topology when using transmit power -15dBm. Node 29 has a subtree with 27 nodes and suffers from very severe queue loss. This load imbalance comes from RPL’s use of only link quality and hop distance for parent selection. Note that real-world deployments are subject to uneven signal density due to physical deployment and wireless link characteristics. This provides an intuition that “non-uniform” transmission power control has the potential to improve bandwidth by load balancing.

As a final note, all results show that bandwidth degradation and PRR unfairness happen together (i.e., some nodes are still healthy while others are suffering), which implies that we may improve bandwidth by relieving a few suffering nodes from their problematic situations. These observations motivate us to design a distributed and adaptive control mechanism for transmission power and routing topology, which addresses both hidden terminal and load imbalance problems to not lose precious bandwidth. Specifically, in

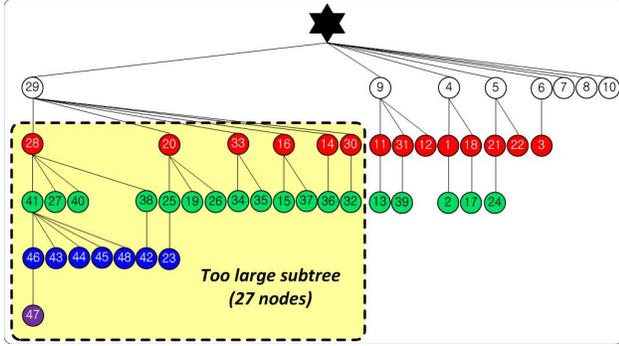


Figure 4. Routing topology of RPL when using transmit power of -15dBm, which clearly shows that RPL has a load imbalance problem.

this work, we design and implement this control mechanism within the standard RPL and call this, *PC-RPL* (*power controlled RPL*).

IV. PC-RPL DESIGN

In this section, we describe *PC-RPL* design in detail. *PC-RPL*'s design comes from the basic intuitions derived from our previous experimental studies.

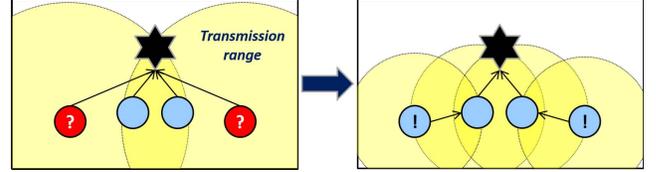
- Many link losses indicate that a node may be experiencing hidden terminal problem, and many queue losses indicate that it may have unbalanced load due to too many children.
- It may be possible to adjust transmit power and routing topology to achieve load balancing and hidden terminal mitigation, resulting in better bandwidth.

Based on these ideas, *PC-RPL* employs a new parent selection mechanism which uses adaptive RSSI thresholds and a reference RSSI value of a parent candidate node, in addition to the default rules in RPL (i.e., hop distance and ETX-based parent selection). *PC-RPL* controls these RSSI thresholds adaptively to mitigate hidden terminal problems and achieve load balancing. Furthermore, *PC-RPL* minimizes data transmit power according to the reference RSSI and transmission results (success or failure), which reduces hidden terminals and link congestion without sacrificing reliability.

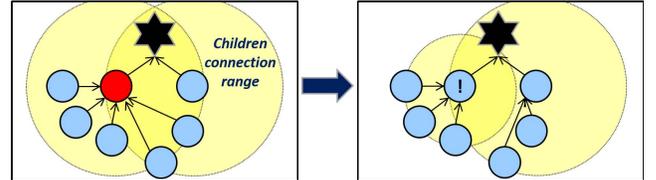
A. Key Concepts and Parameters

Compared to the standard RPL, the most distinct feature of *PC-RPL* is that it uses a reference RSSI value and adaptive RSSI thresholds for parent selection. We first define these key parameters and provide an overview of how they work for bandwidth improvement, before looking into the details of our algorithm.

Reference RSSI value: We define $RSSI_{ref}(k, n_k)$ as the 'reference RSSI value' that node k holds as the reference link distance information to its neighbor node n_k . To obtain this value, *PC-RPL* makes all nodes transmit RPL DIO messages with *maximum* transmit power (0dBm) without transmission power control. Maximum power for DIO allows the network



(a) Hidden terminal mitigation by adapting 'parent selection RSSI threshold' and transmission power



(b) Load balancing by adapting 'children control RSSI threshold' and detaching children nodes

Figure 5. Effect of *PC-RPL*'s two RSSI thresholds on routing topology.

to seek for more link connectivity, if needed, even if only a subset is utilized for data communication. In addition, given that the amount of data traffic is far greater than that of DIO traffic, the cost of using high transmit power for DIO messages can be easily amortized without causing link congestion.

When a node k receives a DIO message from a neighbor n_k , it uses the RSSI of the received DIO message as $RSSI_{ref}(k, n_k)$ and records this value in its neighbor table. Based on this $RSSI_{ref}(k, n_k)$, *PC-RPL* controls the three key parameters $RSSI_{th}^{PS}(k)$, $RSSI_{th}^{CC}(k)$ and $N_{desired}^{SST}(k)$ described below to build a multihop topology that is free from load imbalance and hidden terminal problems.

Parent selection RSSI threshold: $RSSI_{th}^{PS}(k)$ is the 'parent selection RSSI threshold' that node k maintains and uses for parent selection. Node k includes its neighbor node n_k in its parent candidate set \mathbf{P}_k when

$$RSSI_{ref}(k, n_k) > RSSI_{th}^{PS}(k). \quad (1)$$

PC-RPL uses this threshold for hidden terminal mitigation. As shown in Fig. 5(a), when node k suffers from hidden terminal problem, it increases $RSSI_{th}^{PS}(k)$ to select a 'closer' parent node. With the help of transmission power control (explained later), this simple local behavior eventually reduces transmission range of all nodes having hidden nodes, which alleviates the hidden terminal problem around the neighborhood of node k , and eventually in the whole network. Otherwise (if it is free from link loss), node k decreases $RSSI_{th}^{PS}(k)$ to select a longer distance (lower RSSI) node to reduce hop distance.

Children control RSSI threshold: $RSSI_{th}^{CC}(k)$ is the 'children control RSSI threshold' maintained and sent by node k , and used by its neighboring nodes (potential children nodes) n_k for parent selection. To this end, node k propagates

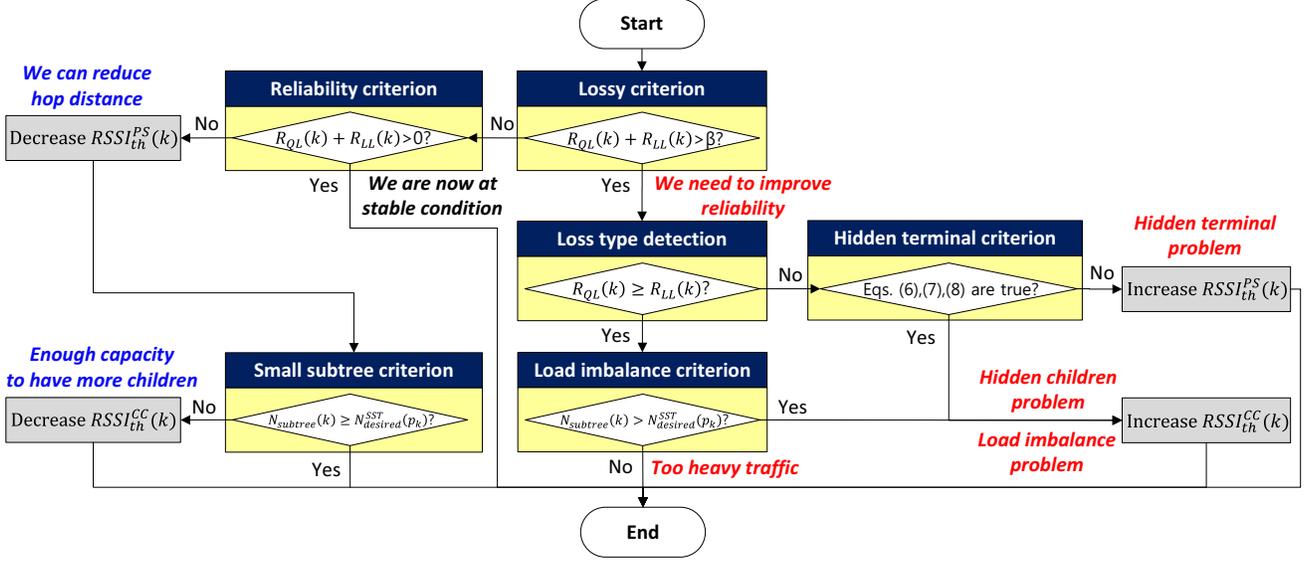


Figure 6. *PC-RPL*'s RSSI threshold control algorithm. Each node *self-detects* the link state it is experiencing: good or bad, hidden terminal or load imbalance problem, and handles each case differently in a fully decentralized manner.

$RSSI_{th}^{CC}(k)$ to its neighbor nodes through DIO messages. A neighbor (potential child) node n_k can add node k to its parent candidate set \mathbf{P}_{n_k} when

$$RSSI_{ref}(n_k, k) > RSSI_{th}^{CC}(k). \quad (2)$$

Note that $RSSI_{ref}(n_k, k)$ is the reference RSSI value that node n_k has for node k .

PC-RPL exploits this threshold for load balancing as shown in Fig. 5(b). When node k detects load imbalance (too many children) from frequent queue losses, it increases $RSSI_{th}^{CC}(k)$ to detach its children nodes (farthest located, first detached). Furthermore, it immediately transmits a DIO (with maximum transmit power) to rapidly propagate the increased $RSSI_{th}^{CC}(k)$ value to its children nodes. Then, a subset of the children nodes which have weak signal strength to node k (i.e., low reference RSSI value) are forced to change their parents, which results in reduced traffic load at node k .

However, queue loss may still occur even in a load balanced network if traffic load is higher than achievable bandwidth. In this case, increasing $RSSI_{th}^{CC}(k)$ may worsen performance by increasing hop count meaninglessly. To treat the overloaded situation differently, we use another parameter $N_{desired}^{SST}(k)$ described below.

Desired number of sub-subtree nodes: $N_{desired}^{SST}(k)$ is the desired number of sub-subtree nodes (of node k) for each child node of node k . Node k calculates $N_{desired}^{SST}(k)$ using the total number of nodes in its subtree¹, divided by the number of one-hop (direct) children nodes. Then, it propagates

¹Each RPL node can obtain its subtree size from the number of downward routing entries in the routing table.

$N_{desired}^{SST}(k)$ to its children nodes via DIOs². When node k experiences high queue loss rate, it distinguishes load imbalance from excessive traffic by using the $N_{desired}^{SST}(p_k)$ value received from its parent node p_k . Specifically, when its subtree size $N_{subtree}(k)$ is larger than $N_{desired}^{SST}(p_k)$, it detects load imbalance problem and increases $RSSI_{th}^{CC}(k)$. Otherwise, it assumes excessive traffic, and takes no action to maintain stability.

B. RSSI Threshold Control

Fig. 6 describes *PC-RPL*'s RSSI threshold control algorithm, which operates on each node in a fully distributed manner. *PC-RPL* classifies transmission results into three categories: success, link loss and queue loss. It accumulates these results and periodically runs the control algorithm.

First, it checks whether the total loss rate is larger than a 'loss rate threshold' $\beta = 5\%$, i.e.,

$$R_{QL}(k) + R_{LL}(k) > \beta \quad (3)$$

where $R_{QL}(k)$ and $R_{LL}(k)$ are queue and link loss rate respectively. If not, *PC-RPL* takes no action and will fallback to default RPL since bandwidth is enough to handle the traffic. Otherwise, *PC-RPL* detects that node k is suffering severe packet loss even though RPL's basic topology adaptation, link level retransmission, and *PC-RPL*'s data transmit power control (Section IV-D) put efforts for reliable packet delivery. To alleviate the problem, it tries to figure out which problem it is experiencing by using the 'loss type criterion':

$$R_{QL}(k) \geq R_{LL}(k). \quad (4)$$

²DIO has 16 reserved bits, from which we use 8 bits to deliver $RSSI_{th}^{CC}(k)$ and the rest 8 bits to deliver $N_{desired}^{SST}(k)$.

If Eq.(4) holds (more queue losses than link losses), *PC-RPL* checks the ‘load imbalance criterion’ to see if

$$N_{subtree}(k) > N_{desired}^{SST}(p_k). \quad (5)$$

If this condition is satisfied, *PC-RPL* infers that node k is experiencing load imbalance problem and increases $RSSI_{th}^{CC}(k)$ to reduce its subtree size. Specifically, it increases $RSSI_{th}^{CC}(k)$ ‘just enough’ to detach the farthest child node for the stability of routing topology. Otherwise, if the subtree sizes are already balanced, *PC-RPL* determines that the traffic load exceeds achievable bandwidth and does not take any action.

On the other hand, if Eq.(4) does not hold (more link losses than queue losses), *PC-RPL* makes a decision on how to alleviate the link losses. Specifically, it considers a ‘hidden terminal criterion’ consisting of the three conditions below to determine whether or not it should increase $RSSI_{th}^{PS}(k)$ to seek for an alternative parent:

$$RSSI_{th}^{CC}(k) \leq CCA_{th} \quad (6)$$

$$N_{subtree}(k) > N_{desired}^{SST}(p_k) \quad (7)$$

$$|\mathbf{P}_k| = 1 \quad (8)$$

where CCA_{th} is the clear channel assessment (CCA) threshold (-77dBm by default) used for CSMA.

If Eq.(6) holds, node k may have children nodes with reference RSSI less than CCA_{th} , which means that they cannot detect node k ’s data transmissions (to its parent node p_k) with CCA^3 . This may incur link losses due to collisions between node k ’s ACK receptions (from node p_k) and the children nodes’ data transmissions. In this case, increasing $RSSI_{th}^{PS}(k)$ is meaningless since hidden terminals are children nodes. If Eq.(7) holds, node k has large number of children nodes that are affected by its parent change, and thus, its decision may have critical impact on the stability of routing topology. Lastly, if Eq.(8) holds, further increase of $RSSI_{th}^{PS}(k)$ may incur route inconsistency, since node k has only one node in its parent candidate set. If all the three conditions are true, *PC-RPL* determines that the main cause of the link losses are due to having a large number of *hidden* children nodes, and decides to increase $RSSI_{th}^{CC}(k)$ to detach the farthest child node. Otherwise, *PC-RPL* detects that node k is suffering from hidden terminal problem, and decides to increase $RSSI_{th}^{PS}(k)$ ‘just enough’ to exclude the current parent node from the parent candidate set and obtain an alternative parent node with greater signal strength.

If node k experiences packet loss below the loss rate threshold (i.e., $0 < R_{QL}(k) + R_{LL}(k) \leq \beta$), it keeps all its thresholds unchanged to favor the stability of routing topology. On the other hand, if node k experiences no packet

³Given that CC2420’s receiver sensitivity is -95dBm, a node can receive a packet even though its RSSI is lower than CCA_{th} . Thus, it is possible that a node receives a DIO message from a neighbor node whose reference RSSI value is lower than CCA_{th} , and selects it as the parent node.

loss at all, it attempts to relax the two RSSI thresholds to improve efficiency (in terms of hop distance and forwarding traffic) while maintaining its current reliability. That is, *PC-RPL* reduces $RSSI_{th}^{PS}(k)$ ‘just enough’ to include the ‘closest’ neighbor node that can provide smaller hop distance than the current parent node p_k in the parent candidate set. Furthermore, if its subtree size is smaller than $N_{desired}^{SST}(p_k)$, it linearly (additively) decreases $RSSI_{th}^{CC}(k)$ to allow more children nodes. This helps other nodes which have large subtrees relieve their forwarding burden. Lastly, whenever a node detects route inconsistency, it re-initializes both $RSSI_{th}^{CC}(k)$ and $RSSI_{th}^{PS}(k)$ to -90dBm.

C. Parent Selection

Once all the threshold values are determined, parent selection process of *PC-RPL* is a straight-forward extension of the standard RPL. When a *PC-RPL* node k determines whether to include a neighbor node n_k to its parent candidate set \mathbf{P}_k , it first considers hop distance ($hop(n_k) < hop(k)$) and link layer ETX as in standard RPL. Additionally, *PC-RPL* requires the following condition to be satisfied.

$$RSSI_{ref}(k, n_k) > \max\{RSSI_{th}^{PS}(k), RSSI_{th}^{CC}(n_k)\} \quad (9)$$

By the definitions and the control mechanisms of the two RSSI thresholds, a *PC-RPL* node k will do its best to select a parent node such that it exploits a good-quality wireless link and avoids both hidden terminal and load imbalance problems.

However, it is possible that a node k needs to increase its $RSSI_{th}^{PS}(k)$ due to hidden terminal problem, but has only the current parent node p_k in its parent candidate set without any alternative. To allow this node to escape from hidden terminals, unlike RPL, *PC-RPL* allows this node to temporarily relax the hop distance condition to $hop(n_k) \leq hop(k)$ for the current selection process (i.e., temporary increase of parent candidates). The reason we relax this condition only *temporarily* is to avoid routing loops. Lastly, when a node experiences hop distance increase after changing its parent node, it immediately transmits a DIO message to fast propagate this information and avoid any potential routing loop.

D. Transmission Power Control

After a parent change to node p_k , a node k configures its *data packet transmission power* to p_k , $P_{tx}^{data}(k)$, as

$$P_{tx}^{data}(k) = P_{tx}^{DIO} - \left(RSSI_{ref}(k, p_k) - RSSI_{th}^{def} \right) \quad (10)$$

As aforementioned, DIO transmission power P_{tx}^{DIO} is fixed to 0dBm (max)⁴. Default RSSI threshold $RSSI_{th}^{def}$ is set to -77dBm (= CCA_{th}). By setting $P_{tx}^{data}(k)$ larger than this

⁴We also fix the transmit power for ACK packets (from a parent to a child, for upward traffic) to 0dBm (max). It is for successful ACK delivery to all children nodes without sophisticated child-by-child ACK transmission power control.

$RSSI_{th}^{def}$ value, we allow node k 's data transmissions to be detected by CCA of its parent node p_k (i.e., not hidden). This selection of $P_{tx}^{data}(k)$ allows the node k to use 'just enough' transmit power to reach its parent p_k while maintaining reliability⁵.

This initial configuration of $P_{tx}^{data}(k)$ based on the reference RSSI value is fast but needs further optimization since RSSI could be an imperfect metric due to external interference [13] and link asymmetry [14]. After this initial configuration, it uses the following set of rules to adaptively control the transmit power. If the node succeeds in transmitting M consecutive packets without any retransmission (where M is the 'good channel threshold', 20 by default in our implementation), it decreases the transmit power linearly by 1 dBm (or one level allowed by the radio) to probe for a lower reliable power. Otherwise, if a packet transmission fails, it increases the transmit power additively by 2 dBm (or two levels allowed by the radio) and also increases the good channel threshold exponentially by $M \leftarrow M * 2$. The purpose of this exponential increase of M is to reduce the transmit power more conservatively when a packet loss is followed by a recent transmit power increase, thus suppressing repeated cycles of increases and decreases. Otherwise, it maintains its current transmit power. Finally, whenever a node detects route inconsistency, it re-initializes the transmit power and M to the initial values.

The goal of our transmission power control is to use 'just enough' transmit power to reach its parent reliably. By coupling the *data transmission power* and *parent selection RSSI threshold* control together, *PC-RPL* mitigates both hidden terminal and link congestion as depicted in Fig. 5(a), resulting in bandwidth improvement.

V. EVALUATION

In this section, we evaluate the performance of *PC-RPL* on a testbed setup and compare it against RPL and QU-RPL [6][15], a queue-utilization based RPL which was recently proposed to tackle the load imbalance problem of RPL (but not hidden terminal issue). We also discuss the details of how *PC-RPL* adjusts its parameters to improve bandwidth.

A. Packet Delivery Performance

Fig. 7 plots the aggregate bandwidth of RPL (at 0dBm), QU-RPL (at 0dBm), and *PC-RPL*, and shows that *PC-RPL* provides better bandwidth than the others. At a load of 2880ppm, *PC-RPL* achieves 2810ppm of aggregate bandwidth (i.e., 17% more than RPL and 7% more than QU-RPL). In other words, *PC-RPL* reduces packet loss rate by 7 times compared to RPL and 3.5 times compared to QU-RPL.

Fig. 8 plots various performance metrics of RPL, QU-RPL, and *PC-RPL* at a traffic load of 2880ppm. Several

⁵If $RSSI_{ref}(k, p_k) < RSSI_{th}^{def}$, we set $P_{tx}^{data}(k)$ to 0dBm (max).

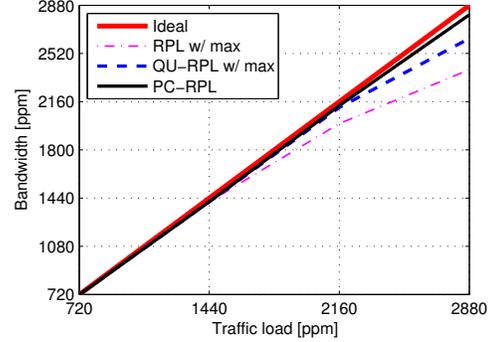


Figure 7. Bandwidth of RPL, QU-RPL, and *PC-RPL*. *PC-RPL* successfully delivers 97.5% of traffic with packet losses that are 7 times less than RPL and 3.5 times less than QU-RPL.

important observations can be made from the PRR results in Fig. 8(a). First of all, QU-RPL improves PRR significantly compared to RPL, at all transmit power settings. This implies that load balancing alone has a significant impact on improving bandwidth. Second, *PC-RPL* also provides dramatic PRR improvement over RPL, which is higher than the best case of RPL with uniform transmit power (-5dBm). In the perspective of PRR fairness, *PC-RPL* improves the PRR by 64% for the worst-case node compared to RPL with 0dBm. This shows that the use of adaptive and non-uniformly distributed transmit power achieves better bandwidth.

However, QU-RPL's PRR performance degrades as transmit power increases, and further investigation reveals that most of packet losses are link losses since QU-RPL cannot alleviate the hidden terminal problem. *PC-RPL* outperforms QU-RPL, even in the best case of QU-RPL with uniform transmit power at -10dBm. This is because *PC-RPL*'s adaptive control mechanism resolves both the load imbalance and hidden terminal problems. More importantly, our *PC-RPL* does not require a system designer to manually optimize transmit power, which is lacking in both RPL and QU-RPL.

B. Link and Routing Layer Behavior

Figs. 8(b) and 8(c) plot the hop count and end-to-end ETX of each node. From the former, we observe that the hop distance under both RPL and QU-RPL increases as transmit power decreases due to shorter transmission range. We can also see that QU-RPL requires larger hop distance than RPL, since it constructs a balanced tree topology by sacrificing hop distance. In contrast, *PC-RPL* does not increase hop distance compared to RPL and provides end-to-end ETX lower than both RPL and QU-RPL regardless of their transmit power levels by alleviating both the hidden terminal and load imbalance problems.

Figs. 8(d) and 8(e) show routing layer's behavior of the three protocols. We first observe that QU-RPL's parent change frequency is larger than that of RPL in almost all cases, since it detects the load imbalance problem and tries

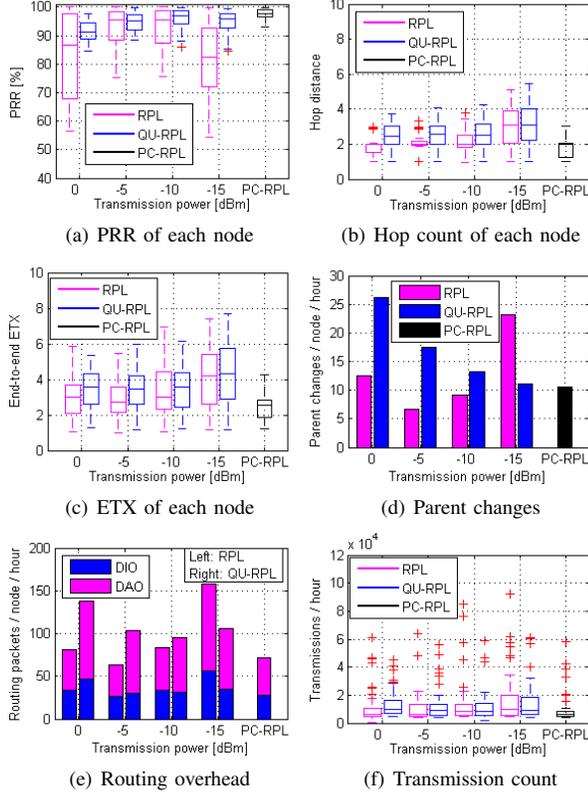


Figure 8. Performance of RPL, QU-RPL and *PC-RPL* with varying transmit power at traffic load of 2880ppm (60ppm/node). *PC-RPL* alleviates hidden terminal and load imbalance problems, and thus outperforms RPL and QU-RPL in terms of bandwidth and routing stability.

to avoid it by changing parent nodes. Combined with the results in Fig. 8(a), this implies that RPL's relatively fewer parent changes are not because it provides good and stable topology, but because it maintains a poor topology due to lack of knowledge. RPL also suffers from meaningless parent changes when link congestion becomes severe (at -15dBm).

Furthermore, QU-RPL's parent change frequency increases with transmit power, since QU-RPL cannot stabilize the routing topology due to the hidden terminal problem. In contrast, parent change frequency of *PC-RPL* is smaller than that of QU-RPL in all transmit power settings while maintaining a high PRR. This reveals that *PC-RPL* avoids meaningless parent changes while balancing load through jointly controlling transmission power and routing topology. *PC-RPL*'s parent change frequency is similar to the RPL's best case, which shows its efficient operation.

In addition, Fig. 8(e) shows that routing control overhead is roughly proportional to the parent change frequency. This implies that QU-RPL and *PC-RPL* achieve PRR improvement at the expense of more routing overhead. However, given that each node generates 3600 data packets per hour in our scenario, this increase in routing overhead (10~20

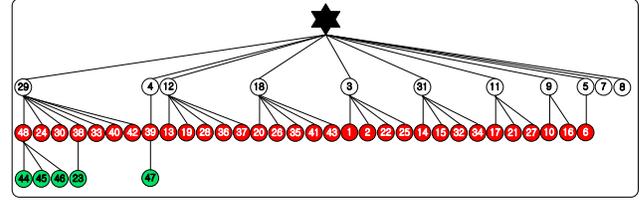


Figure 9. A snapshot of *PC-RPL*'s routing topology at traffic load of 60 ppm/node. Compared to Fig. 4, *PC-RPL* has taken load balancing action.

extra packets per hour compared to the RPL's best case) is a negligible cost compared to significant performance improvement.

Fig. 8(f) plots the total transmission count (including data and control packets) of each node. Interestingly, in all cases, QU-RPL requires more transmissions on average, but reduces that of the most bottlenecked (worst case) node compared to RPL. The former result comes from the larger hop distance of QU-RPL, and the latter from its load balancing effect. However, *PC-RPL* does not lose any of hop distance, ETX, and load balance, which reduces both transmission overhead for the bottlenecked node (-5.4%) and average transmission overhead (-4.4%) than the RPL's best case.

Finally, Fig. 9 is a snapshot of *PC-RPL*'s routing topology during an experiment. Compared to Fig. 4, this figure shows that the previously largest subtree (with 27 nodes) has been divided into four smaller subtrees where the largest one (i.e., node 29's subtree) has 11 nodes, resulting in a shallower and relatively more balanced network. This means that *PC-RPL*'s load balancing function has taken effect as desired.

C. Transmission Power and RSSI Thresholds Control

Fig. 10(a) plots the PRR experienced by *PC-RPL* for an hour of time by during an experiment. Although *PC-RPL* experiences some losses during the beginning of the experiment, when all threshold values and transmit power were uniform (default) among nodes, it improves bandwidth as time goes by through distributed and adaptive control of the two thresholds and transmission power.

Fig. 10(b) presents transmission power of each node during the same experiment. It clearly shows that *PC-RPL* constructs a multihop network with heterogeneous transmit power. In this experiment, *PC-RPL* reduces transmit power from 0dBm to -6.21dBm on average. Recalling that *PC-RPL* provides similar hop distance to RPL's 0dBm case (Fig. 8(b)) with reduced total number of transmissions (Fig. 8(f)), we can see that maximum transmission power does not necessarily result in minimal hop distance and transmission overhead. Furthermore, we can confirm that, in complex wireless environments, 'non-uniform' transmission power is needed for more balanced topology.

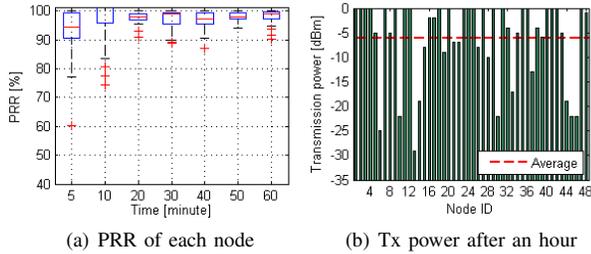


Figure 10. Detailed operation of *PC-RPL*. *PC-RPL* improves PRR as time progresses by adjusting the two RSSI thresholds and transmission power.

VI. RELATED WORK

Several studies have investigated topology and power control in wireless multihop network, and Santi [4] provides an excellent survey of those work. However, most of the work in [4] are graph-theoretical based on idealized graph models and node distributions without real implementation. The survey itself writes; “*despite many theoretical and simulation-based evidences of the effectiveness of topology control techniques ... there is little experimental evidence that topology control can actually be used to these purposes*”. We answer this statement by implementing a topology control scheme for improved bandwidth on real embedded devices, built into the Internet standard IPv6 routing protocol, and providing extensive experimental evidence of its effectiveness on real multihop network.

A few prior works have investigated transmission power control within LLNs using real implementations, either to manage network topology [16][17], or to reduce energy consumption and improve spatial reuse [18][19]. In both cases, transmission power control mechanisms were designed based on either RSSI [18][17] or trial-and-error active probing [16][19] to respect the resource constraints of LLN devices. *PC-RPL* combines ideas of both approaches, but goes beyond by explicitly tackling hidden terminal and load imbalance problems under high traffic scenarios.

The load imbalance problem of RPL has been investigated in several prior work [6][20][21][22][23]. LB-RPL [21] improves load balancing of RPL by allowing a node to prioritize its parent candidates based on their queue utilization. M-RPL [22] detects traffic congestion through RPL control messages and provides two parent nodes for traffic distribution. BD-RPL [23] restricts the subtree size of each node to relieve congestion, and QU-RPL [6][15] uses queue utilization as an indicator to resolve congestion and load imbalance in RPL. Only the latter two provide experimental evidence on testbeds comprising real embedded devices, and more importantly, none of these works examined the use of transmission power control over a real multihop LLN for jointly mitigating hidden terminal and load imbalance problems.

VII. CONCLUSION

We presented an adaptive and distributed mechanism for jointly controlling transmission power and routing topology in low-power multihop wireless networks. We have shown how routing topology and hidden terminal problem affect achievable bandwidth in multihop network, and identified cases where a uniform transmission power configuration can be improved for better bandwidth. We then implemented our control mechanism on top of the standard RPL, called *PC-RPL*, that aims to achieve better bandwidth compared to RPL. *PC-RPL* tackles the hidden terminal and load imbalance problems in multihop network by controlling the routing topology via transmission power and RSSI threshold control. We evaluated *PC-RPL* through extensive experiments on a real 49-node testbed, and our results showed that *PC-RPL* alleviates the packet loss problem which led to significant improvement in achievable bandwidth and routing stability.

REFERENCES

- [1] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” *RFC 6550*, Mar. 2012.
- [2] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection Tree Protocol,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys’09)*, 2009.
- [3] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing,” in *The 9th ACM International Conference on Mobile Computing and Networking (MobiCom’03)*, Sep. 2003.
- [4] P. Santi, “Topology Control in Wireless Ad Hoc and Sensor Networks,” *ACM Computing Survey*, vol. 37, no. 2, pp. 164–194, Jun. 2005.
- [5] J. Polastre, R. Szewczyk, and D. Culler, “Telos: Enabling Ultra-Low Power Wireless Research,” in *Proceedings of the 4th IEEE International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.
- [6] H.-S. Kim, J. Paek, and S. Bahk, “QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications,” in *Proceedings of the 12th IEEE International Conference on Sensing, Communication, and Networking (SECON’15)*, Jun. 2015.
- [7] Cisco Systems Inc., “Connected Grid Networks for Smart Grid - Field Area Network / CG-Mesh.” [Online]. Available: http://www.cisco.com/web/strategy/energy/field_area_network.html
- [8] D. Moss, J. Hui, and K. Klues, “Low Power Listening,” *TinyOS TEP 105*.
- [9] J. Ko, J. Lim, Y. Chen, R. Musaloiu-E., A. Terzis, G. Masson, T. Gao, W. Destler, L. Selavo, and R. Dutton, “MEDiSN: Medical Emergency Detection in Sensor Networks,” *ACM Transactions on Embedded Computing Systems (TECS), Special Issue on Wireless Health Systems*, 2010.
- [10] T. Group, “Thread Stack Fundamentals,” Jul. 2015. [Online]. Available: <http://threadgroup.org/>

- [11] E. J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks," *RFC 5867*, Jun. 2010.
- [12] E. K. Pister, E. P. Thubert, S. Dwars, and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks," *RFC 5673*, Oct. 2009.
- [13] H.-S. Kim, H. Cho, M.-S. Lee, J. Paek, J. Ko, and S. Bahk, "MarketNet: An Asymmetric Transmission Power-based Wireless System for Managing e-Price Tags in Markets," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*, Nov. 2015.
- [14] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio Link Quality Estimation in Wireless Sensor Networks: A Survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, pp. 34:1–34:33, Sep. 2012.
- [15] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 964–979, Apr. 2017.
- [16] G. Hackman, O. Chipara, and C. Lu, "Robust Topology Control for Wireless Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2008.
- [17] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON'04)*, 2004.
- [18] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, 2006.
- [19] J. Ko and A. Terzis, "Power Control for Mobile Sensor Networks: An Experimental Approach," in *Proceedings of the 7th IEEE International Conference on Sensing, Communication, and Networking (SECON'10)*, 2010.
- [20] H.-S. Kim, H. Im, M.-S. Lee, J. Paek, and S. Bahk, "A Measurement Study of TCP over RPL in Low-power and Lossy Networks," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 647–655, Dec. 2015.
- [21] X. Liu, J. Guo, G. Bhatti, P. Orlik, and K. Parsons, "Load balanced routing for low power and lossy networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013.
- [22] M. A. Lodhi, A. Rehman, M. M. Khan, and F. B. Hussain, "Multiple Path RPL for Low Power Lossy Networks," in *IEEE Asia Pacific Conference on Wireless and Mobile*, 2015.
- [23] F. Boubekeur, L. Blin, R. Leone, and P. Medagliani, "Bounding Degrees on RPL," in *ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 2015.

APPENDIX A.

PRELIMINARY FORWARDING STUDY

This appendix provides a brief background to set the traffic loads (720ppm to 2880ppm) for the experimental study in Section III. In the collection scenario, all traffic must converge at the border router and cross the serial link, so the available throughput of these flows, along with the wireless point-to-point flows, must be established to condition the study of wider multihop bandwidth.

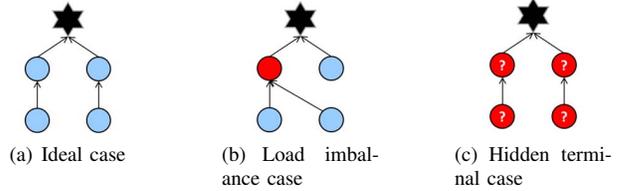


Figure 11. Various 2:2:1 topologies for bandwidth study of 2-hop networks.

We first measure the bandwidth when a node transmits packets to the border router without CSMA, resulting in 3380ppm, $\sim 20\%$ of the ideal bandwidth 16740ppm given that IEEE 802.15.4 supports maximum data rate of 250kbps. The bottleneck is at the transmitter side: processing delay of the embedded device. With CSMA, the bandwidth is degraded to 2815ppm due to the backoff delay of the transmitter. When increasing the number of transmitters to two, the bandwidth increases to 3600ppm and the bottleneck moves to the receiver side: serial link capacity of the border router. This shows that multiple transmitters achieve better bandwidth in LLN.

Next, we measure the bandwidth at the border router in various 2-hop topologies to have a glimpse of multihop characteristics. First, the bandwidth of a 2-hop line topology (1:1:1) is only half of that of the 1:1 single hop topology. This is because the 1-hop node has to relay all packets to the border router, but cannot access wireless channel frequently due to contention with the 2nd hop node, experiencing significant queue loss.

We measure the bandwidth when delivering traffic through multiple branches, by using the three types of 2:2:1 topologies depicted in Fig. 11. The ideally balanced topology in Fig. 11(a) achieves 3330ppm, which shows that using multiple branches improves bandwidth, as in the single hop case. The load imbalanced case in Fig. 11(b) degrades bandwidth to 2210ppm due to queue loss at the overloaded node, which reveals that load balancing highly impacts bandwidth when using multiple branches. Finally, the hidden terminal case (without CCA) in Fig. 11(c) achieves 3060ppm, which shows that hidden terminal causes bandwidth degradation due to packet collisions.

The maximum traffic load region used in Section III is 2880ppm, which is achievable when multiple branches are effectively used. Overall, this preliminary study provides three findings as follows:

- Load balancing among multiple branches improves bandwidth by addressing limited processing delay and queue size of embedded devices.
- Hidden terminal problem degrades bandwidth by causing packet collisions. Moreover, link losses imply the presence of hidden terminals.
- If a bandwidth degradation is observed in Section III, this implies that the network protocol may misbehave since wireless capacity is large enough.