Multichannel Cross-Layer Routing for Sensor Networks

Noradila Nordin
University College London
Email: noradila.nordin.12@ucl.ac.uk

Richard G Clegg
Imperial College London
Email: richard@richardclegg.org

Miguel Rio
University College London
Email: miguel.rio@ucl.ac.uk

***Abstract*—This paper proposes a new multi-channel tree building protocol for ad-hoc sensor networks. Our protocol alleviates the effect of interference which results in improved network efficiency and stability, link reliability and minimised latency. Our proposal takes into account all available channels to utilise the spectrum. It checks the condition of all the channels before deciding on a channel to switch into. The successful transmission rate of the channels are stored externally from the sensors which can be accessed when require. This information is used to limit the channels to be considered when channel switching is invoked. The channel that is selected is checked for any changes in its condition that might had taken place after it was checked previously before committing to the channel. The results and decisions are informed to the other nodes to update their neighbour table. We use two-hop colouring protocol to avoid collision. Our protocol is inspired by the routing protocol for low power and lossy networks (RPL). Packets will be sent to the destination the same way as a single channel RPL but with less loss. All nodes are battery operated except for the low power border route (LPBR). This enables a centralised channel switching process at the LPBR. The channel switching process take place after the topology is formed to further improve the transmission rate on the best paths. We implement and evaluate our solution using the Contiki framework. Our experimental results demonstrate an increased resilience to interference, and significant higher throughput making better use of the total available spectrum and link stability.**

I. Introduction

///add intro - comment and why it is an important topic

Sensor networks have to contend with an increasing number of devices that cause wireless interference. Organising the network topology around this interference becomes an enabler for increasing transmission efficiency at a smaller energy cost. Wireless Sensor Networks (WSN) typically use low power radios such as IEEE 802.15.4, a relatively short range transmission standard radio technology in the 2.4Ghz band. The standard allows transmission to occur on several different channels within this band. Unfortunately, the channels used by this technology often suffer interference, for example, from WiFi and Bluetooth. WSNs need to be able to operate reliably in the presence of such interference.

Multichannel communication in wireless networks can alleviate the effects of interference which, as a result, can improve the network efficiency and stability, link reliability and minimise latency. It also enables communication between physically proximate nodes to occur simultaneously without the risk of collision if the communicating nodes use different channels. However, not all channels are free from interference; thus, there is a need to hop to another channel when the quality of the channel deteriorates. Two commonly used types of channel hopping [24] are blind channel hopping and whitelisting. In blind channel hopping, nodes choose from all available channels. Whitelisting on the other hand, filters out those channels that may have bad interference properties. Many studies make use of channel whitelisting such as [24], [26], Chrysso [12] and MiCMAC [1]. However, they used different channels with channel 26 in common in their respective experiments as the interference would be on different channels depending on the locations.

It is notable that these protocols can use all available channels without whitelisting. However, they do not have a mechanism to check the channel condition before using it for packet transmission. In [1], MiCMAC sees its performance degraded when using more than 4 channels, thus the decision on specifying 4 channels to be included in their experiment. MiCMAC uses different wakeup channel each time it wakes up, thus, it sends packet on different channel each time. However, it might try to send on bad channels for a while before it finds a good channel to deliver the packet as the channel changes at each wakes up. Chrysso on the other hand, switches the affected nodes to a new set of channels upon detecting interference. It would require frequent channel switching if all channels are to be considered.

It is clear that it is impossible to find a single channel guaranteed free from interference and there is no consensus on the best channel to use. Our work takes into account all available channels to utilise the spectrum and checks the condition of the channels before hopping to avoid those channels with interference.

Several previous studies have developed a multichannel MAC layer but, despite the potential benefits none are yet widely implemented in real world deployments. The usual focus is on MAC layers that operate in an autonomous fashion. This paper focuses instead on a cross-layer multi-channel model where a centralised controller can make and communicate decisions about channels and this decision is implemented by the MAC layer. Our Multichannel Cross-Layer Routing Protocol (MCRP) provides feedback when a channel is subject to interference using a probing phase.

In this paper, we develop a cross-layer multi-channel protocol which allows a centralised intelligence to determine which channels each node. The protocol also introduces a probing phase that checks whether assigned channels are free of interference. This protocol is tested using a two-hop colouring protocol to reduce interference between physically proximate nodes trying to communicate on the same channel. The system is failed safe in the sense that the WSN functions if the central system which assigns channels fails temporarily or permanently.

We implement MCRP in Contiki and evaluate the protocol in Cooja simulated environment and in 26-nodes testbed FlockLab [15]. We demonstrate that MCRP avoids channels with interference which greatly reduces the effects of interference on the network.

The rest of the paper is organised as follows: Section II presents related work to multichannel protocols. Section III describes the key idea of our proposed protocol and the high-level design and the implementation of the protocol in Contiki. We describe and evaluate the experimental results in Section IV. Finally, we conclude in Section V.

II. Related Work

Radio duty cycling mechanisms can be classified into two categories; synchronous and asynchronous systems. A synchronous system is a system that requires a tight time synchronisation between the nodes. It uses time-scheduled communication where the network clock needs to be periodically synchronised in order for the nodes not to drift in time. Asynchronous systems on the other hand, do not require synchronisation but instead is a sender or receiver initiated communication. In asynchronous systems the nodes are able to self-configure without time synchronisation and this can have advantages. There are many studies done in multichannel for both categories. Multichannel synchronous protocols include MC-LMAC [11] which uses a time slot to transmit on a particular channel and Y-MAC [13], and TSCH [19] that depend on the neighbouring nodes to synchronise with each other. Multichannel asynchronous protocols such as EM-MAC [18], MuChMAC [4], Chrysso [12], MiCMAC [1] and our protocol are independent of time slot and synchronisation.

ContikiMAC [6] radio duty cycling mechanism is the default radio duty cycling protocol in Contiki that is responsible for the node wake-ups period. ContikiMAC is a power-saving radio duty cycling protocol. It was proved to be efficient in a single channel [1], [7]. ContikiMAC uses periodical wake-ups to listen to the neighbours' transmission packet. It has a phase-lock mechanism to learn the neighbours' wake-up phase to enable efficient transmissions and a fast sleep optimisation in case of spurious radio interference. The sender uses the knowledge of the wake-up phase of the receiver to optimise its transmission. When a packet is successfully received, the receiver sends a link layer acknowledgement. The sender repeatedly sends its packet until it receives a link layer acknowledgement from the receiver. ContikiMAC relies on retransmissions for reliable transmissions. A Carrier Sense Multiple Access, CSMA is a MAC protocol that performs retransmissions when the underlying MAC layer has problems with collisions. When the sender does not receive the link layer acknowledgement, ContikiMAC with CSMA will retransmit the packets three times before dropping it from the buffer queue.

MiCMAC [1] is a ContikiMAC [6] channel hopping variant. On every wakeup cycle, the channel is periodically switched according to a pseudo-random sequence. MiCMAC introduces channel lock for the channel reception at the sender. There is a dedicated broadcast channel for a duration at every wake up period. Chrysso [12] is a multichannel protocol for data collection applications. The nodes are organised into parent-children groups where each parent-children uses two channels for transmitting and receiving packets. Our work also uses two separate channels as in Chrysso. Both parent and children nodes can hop to another channel when interference is detected based on the channel switching policies. If a node loses connectivity, Chrysso calls the scan mode to enable neighbour discovery over multiple channels. Chrysso functionality comprises a set of channel switching policies that interface to both the MAC layer and the network layer.

In order to maximise the use of multichannel in improving packet delivery, routing topology plays a big role in providing an optimised routing tree to the network that is scalable and energy efficient. There are many studies that were done on routing protocol such as LEACH [2] that form clusters, PEGASIS [16] forming chains of nodes, CTP [9], and RPL which is designed largely based on CTP. Recent multichannel protocols such as MiCMAC is compatible with RPL as the routing protocol. Chrysso uses Contiki collect which is a CTP-like data collection protocol in Contiki. We choose to use RPL as it is the standard for IPv6 routing in low power and lossy networks.

RPL [25] is a gradient based routing protocol forming any-to-any routing for low power IPv6 networks. RPL topology is a Destination-Oriented Directed Acyclic Graph (DODAG), rooted at LPBR with no cycles. The topology is set up based on the routing metric [23] such as hop count and expected transmission count (ETX) metrics to calculate the distance of the nodes to the LPBR. The root has the overall view of the network. The other nodes however, only have knowledge of its neighbours and default router which is the parent. RPL is a rooted topology which any-to-any traffic is directed towards the root unless the common ancestor is found which the traffic is then routed downwards towards the destination. This strategy is used in order to scale large networks by reducing the routing overhead at the cost of increased hop count through common ancestor. In RPL terminology, the node distance to the root and other nodes is defined as the node's rank. By default, RPL uses ETX metric to find the path with the minimum number of transmissions that a node expect to successfully deliver a packet to the destination and switches only if it is less than the current rank to prevent frequent changes [8].

MiCMAC and Chrysso are fully distributed and allow the nodes to self configure and change to another channel when interference happens. The channels that the protocols can use are fixed to a subset of whitelisted channels. This contrasts with our protocol where the control decisions as to channel assignment happen behind the Low Power Border Router (LPBR). The LPBR, which is the point where a WSN connects to other networks, is fully powered and does not have the limited memory of low power nodes. In standard RPL this point in the system also has knowledge of the network topology that can be used in the decision making process. We are able to produce real time channel selection decisions where we can consider all available channels to be used in transmissions without blacklisting any of them.

Our proposed protocol takes into account RPL topology formation scheme and the control messages exchange between nodes that take place periodically to maintain the quality of the tree. We enable the RPL control messages to be sent through both unicast (for known neighbours) and broadcast only on the default channel (for new nodes to join) in order to reduce unnecessary transmitting through all channels. Our work makes use of RPL topology formation and improves on the channels of the nodes in the topology. Our protocol is a cross-layer protocol with a centralised co-ordinator that enables us to make real time decisions without being constraint by the layer capabilities. Our centralised LPBR has the intelligence in choosing the channel for the node as it has the full overview of the network. We also do not blacklist any channel as we do channel checking each time a channel change occur in order to have the update on the channel condition.

III. Multichannel Cross-Layer Routing Protocol

Multichannel Cross-Layer Routing Protocol (MCRP) concentrates on finding channels for nodes that are free from or have low interference. It allows the allocation of these channels in a way likely to minimise the chances of nodes which are physically near communicating on the same channel and hence reduce cross interference between different pairs of nodes.

A. Overview

The design of the multichannel protocol is based on several crucial observations:

- Channel assignment - Sensors have limited memory and battery capabilities. In order to maximise the sensors lifetime, a centralised LPBR that has larger memory and fully powered is used for decision making. LPBR has a complete knowledge of the topology which enables it to make good channel assignment decisions based on the criteria that are explained in the next section.
- Interference - External interference cannot be predicted, thus channels cannot be allocated beforehand as it varies over time and locations. It is impossible to determine a single channel that is free from interference at any location. Our protocol checks the channel condition each time before deciding on a channel change to reduce interference and maximise throughput.
- Frequency diversity - Multichannel increases the robustness of the network towards interference. However, applying multichannel to the existing RPL may hinder neighbour detection and RPL processes to maintain the network topology as it does not switch to the correct channel. We overcome this problem by enabling both broadcast on a default channel, and unicast, in neighbour detection and RPL control messages.
- RPL - as we are using multichannel, sending a broadcast on every channels would waste the bandwidth. RPL messages are instead being sent through unicast when the neighbours are known and broadcast in the default Contiki channel 26 in order to reduce unnecessary transmitting in broadcast. By sending a broadcast on only one channel which the new neighbours are going to start on, the nodes can be discovered and the channel changes processes can be done.
- Topology - The topology could change according to the routing metric as the how the usual RPL would work. The nodes can still change the parents as usual as all neighbours know each other new channels. The neighbours that are not part of the route do not probe the parent when making the channel decision. However, the neighbours are informed of any channel changes.

Our cross-layer multi-channel protocol focuses on the network and application layer of the protocol. This allows channel assignment decisions to be made thoroughly without being limited by the low layer complexity. The channel assignment processes take place only after the topology tree has been formed by RPL and stabilised. The system has two parts: a central algorithm which can be collocated with the LPBR and selects which channel each node should listen on; and a protocol which allows the network to communicate the channel change decision, probe the new channel and either communicate the success of the change or fall back to the previous channel.

B. Channel Selection Strategy

The system we propose is general and any algorithm can be used at the LPBR to assign channels. In this paper we use a two-hop colouring protocol to select a channel to be assigned to a node. Channels are chosen in a way that ensures no nodes within two hops of each other on the network are listening on the same channel. This allows fair load balancing on the channels and reduces channel interference that could occur when two nearby nodes transmit together on the same channel. The nodes used in this paper have a transmission range of approximately 50 metres indoors and 125 metres outdoors. It could be the case, therefore, that many nodes in a sensor network are in the transmission range of each other and potentially interfered with.

All nodes are initialised to channel 26 by default in Contiki. The usual RPL set up mechanism is used to exchange control messages that are required to form an optimised topology before channel assignments can take place. The nodes will only be on the same channel once during the initial setup. Channel 26 is chosen as this is the default initial channel in Contiki since it often has fewer interference problems with WiFi and other sources. The studies in [12], [1], [24] use

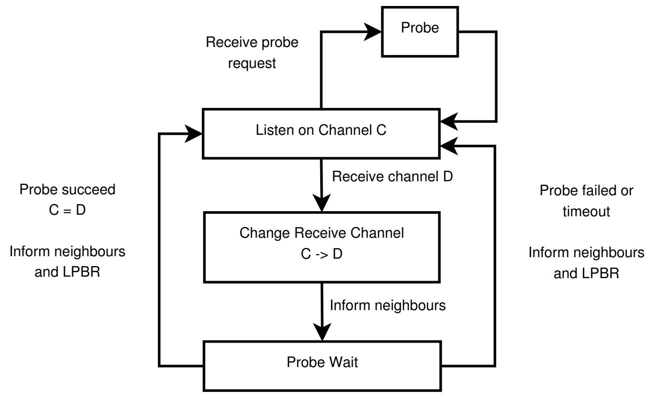

Fig. 1. Channel switching processes

several channels in their experiments and have channel 26 in common.

In the two-hop colouring protocol, the LPBR chooses a node to which it will assign a channel to listen on. This selection is random from channels 11 to 26 (the full range available). The protocol checks neighbours and neighbours of neighbours to see if any of those are listening on this channel already. If any are, a new channel is picked from the remaining list of available channels. If the LPBR has knowledge of existing bad channels then those channels can be blacklisted. Knowledge of channel interference (gained by probing, see later description) can be used to decide that a channel should not be used. If a channel is found then the channel switching protocol is used (see section III-C). If no channel can be found meeting these conditions the current channel is kept.

The node selection algorithm must only attempt one channel change at a time. The protocol ensures that the channel change attempt will always result in a message returned to the LPBR either confirming the new channel or announcing a reversion to the old channel. Until one or other of these happens, no new channel change will be made.

C. Channel Switching

Figure 1 shows the state machine for the channel switching protocol. As explained in the previous section, a choice of a new channel by the channel selection protocol causes a change channel message to be sent to the appropriate node. On receiving a channel change message a node N stores its current channel C and communicates to all its neighbours the new channel D it wishes to change to. Those neighbours will update their neighbour tables to ensure that they now send to node N on channel D. The node N now begins the channel quality checking process (see section III-D) with each neighbour in turn by sending them a probe request. If this process fails for any neighbour then the node reverts to channel C. Node N informs its neighbours of this reversion to channel C and informs the LPBR of the channel checking results and its reversion to listening on channel C. If all channel quality checks succeed the node N. It is important to emphasise that the network remains fully functional and connected at all stages of this protocol (however, the channel checking process uses probe packets that might interfere with other transmissions temporarily).

D. Channel Quality Checking

In describing the channel quality checking process it is worth emphasising the RPL distinction between neighbours and tree neighbours. In RPL, node neighbours are all nodes that a given node knows it could transmit to. Tree neighbours are the nodes that a node does transmit to through the topology formed by the RPL protocol.

The channel quality checking is invoked each time a node changes channel after receiving a message from the LPBR. A node N changing to channel D informs all neighbours in turn, of the new channel D it will be listening on as described in the previous section. It then enters the Probe Wait state and begins channel quality checking with each tree neighbour in turn.

In the Probe Wait state node N sends a Probe message to each neighbour in turn. That neighbour will respond to the message by sending eight packets to N on the new channel D. If this process times out (because of some communication failure) or the number of probe packets received is below a threshold (currently set to 16) then node N immediately exits Probe Wait and reverts to channel C its previous channel.

The buffer can accommodate eight packets at a time. As the packets might not be sent immediately due to wakes up and collisions, sending more packets would have the risk of being dropped. Increasing the buffer is not ideal as the sensor has limited memory capability. However, as the numbers of probing packets are small, the estimate value might not be accurate as the representation of the channel condition. The condition of the channel is further investigated through the number of retransmissions and packet collisions of the probing packets.

The threshold is set to be 16 to ensure that moderate interference channels are avoided. Mild interference channels usually fall below the threshold. If the threshold is not met, none of the node would change the channel as the node would be better off with its current channel.

All neighbours are informed of the change back to channel C and the LPBR is informed of the quality check failure with a summary of all probes received. If, on the other hand, all channel quality checks succeed the change to channel D becomes permanent for node N and it informs the LPBR of the results of the probing (numbers of packets received) and the channel change.

Probing is an important part in making the channel change decision. It gives a quick overview of the channel condition based on the number of probing messages received. It is worth noting that probing is only done between the node and the tree neighbours. Neighbours that are not tree neighbours will not use the node as a route during their transmission thus, there is no need for probing to take place with those neighbours. However, the neighbours still need to know the channel value as RPL control messages are sent to neighbours directly without using the routes.

IV. Evaluation

The authors present a centralized solution that wastes a lot of energy on probing. In the evaluation section, they do not compare their proposal against any of the existing solutions.

Also, the increase in overhead is not at all studied. ///need to have more details!

In the evaluation section, the proposed solution is not evaluated against the existing ones! Even the comparison against standard RPL is done using different parameter settings. ///scenario 1 and 2 -¿ single channel would have worse result

* The approach proposed by the authors is centralized, and requires communication from the LPBR to the nodes and viceversa. The latter appears to occur upon each channel switching. The communication overhead is never evaluated in the paper: only a passing mention to packets is provided, which is only a part of the picture from an energy standpoint.

* The evaluation uses end-to-end packet delivery as the main performance metric. However, the authors fail to state the key parameter affecting this metric, i.e., the diameter of the network.

* I would have expected that MCRP is able to identify the good channels and use them. Therefore, in the mixed scenario 2, I would have expected MCRP to exploit the 4 good channels, leading the performance at least in between the one of good and mild. Instead, performance is between mild and moderate... why?

* Moreover, in scenario 2 performance still appears to degrade over time as shown in Fig.3, which doesn't happen in scenario 1. Why? It seems that in this latter case MCRP provides only marginal advantages over a single channel (and I would argue it uses more energy, see above)

V. Conclusion

Acknowledgments

Noradila Nordin is a King's Scholar sponsored by the Government of Malaysia.

References

[1] B. Al Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multi-channel. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 2–9, May 2014.

[2] Asaduzzaman and Hyung Yun Kong. Energy efficient cooperative leach protocol for wireless sensor networks. *Communications and Networks, Journal of*, 12(4):358–365, Aug 2010.

[3] Carlo Alberto Boano, Thiemo Voigt, Nicolas Tsiftes, Luca Mottola, Kay Römer, and Marco Antonio Zúñiga. Making sensornet mac protocols robust against interference. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 272–288, 2010.

[4] Joris Borms, Kris Steenhaut, and Bart Lemmens. Low-overhead dynamic multi-channel mac for wireless sensor networks. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 81–96, 2010.

[5] Thang Vu Chien, Hung Nguyen Chan, and Thanh Nguyen Huu. A comparative study on operating system for wireless sensor networks. In *2011 International Conference on Advanced Computer Science and Information System (ICACSIS)*, pages 73–78, December 2011.

[6] Adam Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report T2011:13. ISSN 1100-3154 http://dunkels.com/adam/dunkels11contikimac.pdf, 2011.

[7] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, 2013.

[8] Omprakash Gnawali. The minimum rank with hysteresis objective function, RFC6719. https://tools.ietf.org/html/rfc6719, 2012.

[9] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, 2009.

[10] IEEE. IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[11] Ozlem Durmaz Incel, Lodewijk van Hoesel, Pierre Jansen, and Paul Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Netw.*, 9(1):73–94, January 2011.

[12] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso - a multi-channel approach to mitigate external interference. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 449–457, June 2011.

[13] Youngmin Kim, Hyojeong Shin, and Hojung Cha. Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 53–63, April 2008.

[14] Philip Levis, T Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. RFC6206: The trickle algorithm. https://tools.ietf.org/html/rfc6206, 2011.

[15] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 153–166, New York, NY, USA, 2013. ACM.

[16] S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–1125–3–1130 vol.3, 2002.

[17] Lanny Sitanayah, Cormac J. Sreenan, and Szymon Fedor. A cooja-based tool for maintaining sensor network coverage requirements in a building. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 70:1–70:2, 2013.

[18] A. Sivanantha, B. Hamdaoui, M. Guizani, Xiuzhen Cheng, and T. Znati. Em-mac: An energy-aware multi-channel mac protocol for multi-hop wireless networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 1159–1164, Aug 2012.

[19] Luigi Alfredo Grieco Thomas Watteyne, Maria Rita Palattella. Using IEEE802.15.4e time-slotted channel hopping (TSCH) in an internet of things (IoT): Problem statement. https://tools.ietf.org/html/rfc7554, May 2015.

[20] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (RPL), RFC6552. https://tools.ietf.org/html/rfc6552, 2012.

[21] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Osterlind, Joel Höglund, and Adam Dunkels. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 479–480, New York, NY, USA, 2010.

[22] Tsvetko Tsvetkov. RPL: IPv6 routing protocol for low power and lossy networks. *Sensor Nodes–Operation, Network and Application (SN)*, 59:2, 2011.

[23] J Vasseur, M Kim, K Pister, N Dejean, and D Barthel. Routing metrics used for path calculation in low power and lossy networks. https://tools.ietf.org/html/rfc6551, 2012.

[24] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings*

of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, pages 116–123, 2009.

[25] T Winter, P Thubert, T Clausen, J Hui, R Kelsey, P Levis, K Pister, R Struik, and J Vasseur. RPL: IPv6 routing protocol for low power and lossy networks, RFC 6550. https://tools.ietf.org/html/rfc6550, 2012.

[26] Yafeng Wu, J.A. Stankovic, Tian He, and Shan Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, April 2008.