Multichannel RPL Variant

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. Low power radios such as IEEE 802.15.4 are a relatively short range transmission standard radio technology in the 2.4Ghz band. Unfortunately, the frequency band is shared with WiFi and Bluetooth which cause problem for Wireless Sensor Networks that require minimal packet loss, interference and delay. 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 random channel selection and two hops neighbour strategy to avoid collision. By basing our protocol in routing protocol for low power and lossy networks (RPL), packets can be sent to the destination the same way as a single channel RPL but with less loss. RPL is a gradient based routing protocol forming any-to-any routing for low power IPv6 networks rooted at a single destination called the Low Power and lossy network Border Router (LPBR) with no cycles. The topology is formed by choosing the minimum rank which is the distance from the node to the root based on the minimum expected transmission count (ETX) metric. All nodes are battery operated except for the LPBR. This enables a centralised channel switching processes at the LPBR. The channel switching processes 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, significant higher throughput making better use of the total available spectrum and link stability.**

I. INTRODUCTION

Low power radios such as IEEE 802.15.4 is a relatively short range transmission standard radio technology in the 2.4Ghz band. Unfortunately, the frequency band is shared with WiFi and Bluetooth which cause problem for WSN that require minimal packet loss, interference and delay. The problem arise as all 16 channels overlap with channels used by WiFi. However, the overlapping problem can be solved if WiFi only uses the European non-overlapping channel set which are channels 1, 7 and 13, leaving channels 15, 16, 21 and 22 to be used by 802.15.4 [1]. Another solution is to use multichannel.

Multichannel communication in wireless networks can alleviate the effects of interference which as a result, improve the network efficiency and stability, link reliability and minimise latency. It also enables communication between nodes to occur simultaneously without the risk of collision. However, not all channels are free from interference, thus, the need to hop to another channel when the quality of the channel deteriorates. There are two types of channel hopping [15], blind channel hopping and whitelisting. In blind channel hopping, the node will hops over all available channel. Whitelisting on the other hand, filters out the worst channel. Many studies make use of channel whitelisting such as [15] claimed that channel 11, 15, 25 and 26 are free from Wi-Fi, [17] channel 11, 19 and 25, Chrysso [7] uses channel 11, 14, 20, 22 and 26, and MiCMAC [2] uses channel 15, 20, 25 and 26. From these studies, it can be concluded that it is impossible to determine a single channel that is not affected by interference. Our proposed work takes into account all available channels to utilise the spectrum and checks the condition of the channels before hopping as interference varies over time.

There are many studies that were done in multichannel where most of them concentrated on using multichannel MAC layer. Despite there are many multichannel MAC layer that are available in the literature, multichannel is still not widely implemented even though it has many potential benefits for wireless networks. This might be due to the complexity of the solutions to be implemented.

In this paper, we propose a multichannel RPL variant. As most multichannel is implemented in MAC layer, our work concentrates on the application layer. The nodes are given different channels by the Low Power and lossy network Border Router (LPBR) after the topology is formed to avoid collision in a single channel. LPBR has the knowledge of the whole topology which enables it to assign channel to the nodes. As a result, synchronisation is not require. The nodes communicate on the transmission channel and are always listening on their listening channel. The control messages are sent to the nodes on their listening channel as unicast which eliminate the need for a separate control channel. In Contiki, a fast turnaround time is supported where the channel switching delay of $128\mu s$ is negligible.

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 mechanism can be classified into two categories; synchronous and asynchronous. Multichannel synchronous protocols for such as MC-LMAC, Y-MAC, MuChMAC, EM-MAC and TSCH depend on the neighbouring nodes to synchronise with each other while multichannel asynchronous protocols such as Chrysso, MiCMAC and our protocol are independent.

**NOT YET ADDED - Multichannel protocol such as MC-LMAC, Y-MAC, MuChMAC, EM-MAC. What these works do?

Timeslotted Channel Hopping (TSCH) [10] focuses on the MAC layer which can be used with RPL. It uses time synchronisation and channel hopping for ultra low power operation and high reliability. TSCH schedules the communication in time and frequency slot. However, it requires tight synchronisation to the neighbour nodes. In order to use TSCH with RPL, there are several problems to be addressed in term of the topology and network maintenance as RPL determines the multihop route and has it's own control messages.

Chrysso [7] is a multichannel protocol for data collection applications. The nodes are organised into parent-children groups where each parent-children uses two channel for transmitting and receiving packets. 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 channel. Chrysso concentrates on data collection while our work tries to improve RPL single channel into multichannel without dealing with the MAC layer.

MiCMAC [2] is an asynchronous protocol, ContikiMAC [4] 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. MiCMAC can be used with RPL without any changes to RPL. However, RPL might not formed properly because the nodes need to be on the broadcast channel at different time for different node according to the trickle timer to receive RPL control messages which are sent through unicast and broadcast.

Our proposed protocol takes into account RPL topology formation scheme and the control messages exchange between nodes that take place frequently to maintain the quality of the tree. The RPL control messages are sent through unicast in order to reduce unnecessary transmitting in broadcast. Our work makes use of RPL topology formation and improves on the channel within the topology formed.

III. Multichannel RPL Protocol

ContikiMAC [4] is the default low power listening MAC protocol in Contiki. It was proved to be efficient in a single channel [2][5]. ContikiMAC uses periodical wakeups 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 is detected.

Routing protocol for low power and lossy networks (RPL) 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 root has the overall view of the network. The other nodes however, only has knowledge of its neighbours and default router. 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. RPL finds 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 [6].

Multichannel RPL concentrates on finding the best channels for nodes to listen and transmit on, given policies that needs to be complied.

A. *Overview*

The design of Multichannel RPL are based on several crucial observations:

- Channel assignment - as the nodes are battery powered, the decision of selecting the channel is left to LPBR. This reduces the processing on the nodes and enable load balancing within channels as LPBR has a complete knowledge of the topology. LPBR keeps track of the channel conditions based on the feedback it receives from the nodes. All intelligence is done at LPBR.
- Interference - external interference cannot be predicted, thus channel cannot be allocated beforehand as it varies over time. It is impossible to determine a single channel that is free from interference.
- Frequency diversity - RPL only considers a single channel. Applying multichannel to the existing RPL may hinder neighbour detection and RPL control messages. We solved this by enabling unicast in neighbour detection and RPL control messages. We assume that no new nodes should join the topology after the initial setup.

Multichannel RPL focuses on the application layer of the protocol. The selection of channels to the nodes are decided once the topology tree has been formed by LPBR.

B. *Channel Selection Strategies*

At initialisation, during the tree topology formation, all nodes are on the same channel, which is channel 26 by default. This is because nodes that are on different channels might not be detected which result in unoptimised topology. Channel 26 is used because it is the channel that the studies in [7][2][15] studies have in common. By default, in a single channel case, channel 26 is used as it usually does not overlap with WiFi and is relatively in a cleaner frequency than the other channels.

There are two strategies that LPBR uses when deciding on a channel change; random change and two hop colouring strategy.

1) Random Channel Selection -: At initialisation, LPBR uses random channel to decide which node and what channel the node should try to change into. The neighbours of the node will send probes messages to check the channel condition and the probing result is used to decide on the changes. This strategy is used when LPBR has no knowledge of the suitable channel for the node. By forcing the neighbour-node pair sending probing messages, LPBR could build its knowledge based on the probing results. LPBR will have an overall information of channels which will affect its decision in the future. The advantages of having these information is that LPBR can be certain that the channel that it chooses for a node to change into will have better throughput than the current channel. Through probing results, LPBR knows how much better a certain channel is compared to other channels as this information is only accessible by the LPBR.

2) Two Hops Neighbour Strategy -: ////DID MANY CHANGES, TO BE REWRITTEN!!! In two hops neighbour strategy, similar to random channel selection, LPBR choose a random channel for a node unless LPBR has knowledge of the channels condition as mentioned in Random Channel Selection section. The node checks if there is another node that is two hops away using the same channel. In RPL, the node routing table and neighbour table do not contain the information of all existing nodes in the network but only the nearest nodes that are in range. In order to do the checking, both routing table and neighbour table are used. The routing table is referred to when forwarding the channel value to the specific node while the neighbour table keeps the information on the node and neighbours listening and transmitting channels. This is important as the node should always sends on the correct transmitting channel and return to the listening channel. When a node gets the channel to change into, it first checks it's own neighbour table if any of the neighbour is currently on the channel. If there is none, the node asks the neighbours to check their neighbour table. The decision of whether the channel can be used is passed back to LPBR. LPBR will then use the channel or decide on another channel depending on the reply from the node and neighbours, and the steps explained in section 3.4 proceed. If LPBR could not get a two hops free channel, it will use the default channel which is channel 26.

C. Channel Quality Checking

The channel is check at initialisation in order for the LPBR to build it's knowledge on the channels quality. The channel quality checking is invoke each time the node receives the channel change message from LPBR. The node informs the neighbours of the new channel it might be listening on. The node's neighbours will probe the node and collects the information on the success or failure rate of the channel. The results from probing is used to decide if the new channel is better than the current channel. LPBR is informs and updates its knowledge on the channel condition.

////FUTURE WORK? Even though external interference varies over time, it is unlikely that the channel quality fluctuates frequently within minutes that would affect the receiving success rate. Thus, the channel quality check is invoke on three (cases? situations?); on initialisation, when receiving LPBR channel change message and when the packet delivery ratio (PDR) drops. ***NOT SURE

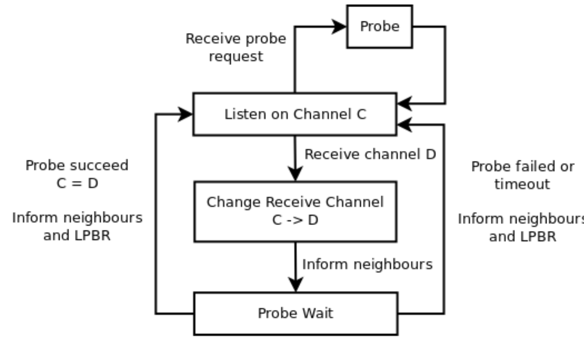

Fig. 1. Channel switching process

**** Each node keeps the counter of packets it has sent and received. It will send the information to LPBR periodically in order for LPBR to get a full view on the condition of the channels. LPBR will then decide if the node needs to change to another channel depending on the sent and received information. If the node does not perform well based on the number of received packets less than the number of packets being sent, LPBR will decide on a new channel the node needs to change into. The node will go through the channel changing processes to decide if the new channel is performing better than the previous channel before (settling?) for the new channel.

D. Channel Switching

LPBR decides the channel that a node should be listening on either by random selection or through LPBR knowledge of the channels. Figure 1 shows the states in channel switching. LPBR sends the change channel message to the node on the receiving node current listening channel. The node saves it's current and new channel to allow the channel to be restored if require. The node contacts its neighbours, informing the new channel value. The neighbours will in turn, send probing messages to check the new channel condition. Different neighbours might have different success rate, thus the final decision of switching is based on the threshold that is set.

If the threshold is not met or the probing has timed out, the node is restores to its current channel and informs LPBR of the updated channel condition. Otherwise, the node sends a change channel message to the neighbours and waits for acknowledgement. The node and the neighbours updates their table. The node informs LPBR of the changes and listens on the new channel for data packets. If the acknowledgement does not arrive before the timeout, the change channel message is retransmitted. The neighbour node is ignored if the retransmissions has timed out.

IV. Evaluation

We describe the experiments that we run to evaluate our multi-channel tree protocol and comparing the results (compare what??) against single-channel tree protocol and MiCMAC.

A. Experimental Setup

We evaluate the protocol in Cooja simulated environment with emulation of TMote sky nodes that feature the CC2420 transceiver, a 802.15.4 radio. The nodes run on IPv6, using UDP with standard RPL and 6LoWPAN protocols. The network consists of (explain border router etc????) that always stays on and it is the sink of the tree, also the root node of RPL. (how many other nodes??) are duty cycled and acts as UDP clients that send packets to LPBR. We simulated a controlled interference node that generates semi-periodic bursty interference on several channels (check??) to measure the benefits of multi-channel operation of changing channel when failure happens within a given period.

- throughput?

B. *Effect of Multi-channel*

//with existings - better? worse? what about RAM, ROM used?

C. *Resilience to External Interference*

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] IEEE standard for information technologytelecommunications and information exchange between systems local and metropolitan area networksspecific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[2] 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.

[3] 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.

[4] Adam Dunkels. The ContikiMAC radio duty cycling protocol, 2011.

[5] 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, page 2:12:14, New York, NY, USA, 2013. ACM.

[6] Omprakash Gnawali. The minimum rank with hysteresis objective function, rfc6719. 2012.

[7] 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.

[8] Philip Levis, T Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. Rfc6206: The trickle algorithm. *Internet Engineering Task Force (IETF) Request For Comments, http://ietf. org/rfc/rfc6206. txt*, 2011.

[9] 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, page 70:170:2, New York, NY, USA, 2013. ACM.

[10] Luigi Alfredo Grieco Thomas Watteyne, Maria Rita Palattella. Using ieee802.15.4e tsch in an lln context: Overview, problem statement and goals. 2014.

[11] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (rpl), rfc6552. 2012.

[12] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Osterlind, Joel Hglund, and Adam Dunkels. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, page 479480, New York, NY, USA, 2010. ACM.

[13] Tsvetko Tsvetkov. Rpl: Ipv6 routing protocol for low power and lossy networks. *Sensor Nodes–Operation, Network and Application (SN)*, 59:2, 2011.

[14] J Vasseur, M Kim, K Pister, N Dejean, and D Barthel. Routing metrics used for path calculation in low power and lossy networks. *draft-ietf-roll-routing-metrics-19 (work in progress)*, 2011.

[15] 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*, PE-WASUN '09, page 116123, New York, NY, USA, 2009. ACM.

[16] 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. *IETF ROLL WG, Tech. Rep*, 2012.

[17] 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*, pages –, April 2008.