

Multichannel RPL Variant

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

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

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

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 [?]. 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 [?], 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 [?] claimed that channel 11, 15, 25 and 26 are free from Wi-Fi, [?] channel 11, 19 and 25, Chrysso [?] channel 11, 14, 20, 22 and 26, and MiCMAC [?] 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 network 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 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 2 presents related work to multichannel protocols. Section 3 describes the key idea of our proposed protocol and the high-level design. The implementation of the protocol in Contiki is presented in Section 4 and the experimental results in Section 5. Finally, we conclude in Section 6.

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 Chryso, 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) [?] 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 its own control messages.

Chryso [?] 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, Chryso calls the scan mode to enable neighbour discovery over multiple channel. Chryso concentrates on data collection while our work tries to improve RPL single channel into multichannel without dealing with the MAC layer.

MiCMAC [?] is an asynchronous protocol, ContikiMAC [?] 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. Our work is a variant of RPL.

III. MULTICHANNEL RPL PROTOCOL

ContikiMAC [?] is the default low power listening MAC protocol in Contiki. It was proved to be efficient in a single channel [?][?]. 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 [?].

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.
- 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.

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

Figure 1 shows the design of Multichannel RPL. The states are explained in the next sections.

B. Channel Selection

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 [?][?][?] 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.

Once the topology has formed and stabilised, LPBR sends a channel change message which random channel is chosen for the node to change into. LPBR chooses random channel as it does not have information regarding the condition of the channels. It will build its knowledge of the channels based on the feedback it receives from the nodes. If LPBR has the knowledge of the channels, it decides on the best channel for the node to maintain high transmitting and receiving rate. LPBR set a timer for the node to reply. If it times out, no changes in done to the channel.

The channel that is selected also needs to (fulfil?) two hops colouring strategies which is explained in detail in the next section. This is done in order to eliminate the risk of next hop neighbour using the same channel which could lead to congestion on the channel.

C. Channel Quality Checking

The channel is checked at initialisation in order for the LPBR to build its knowledge on the channels quality. The channel quality checking is invoked 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. However, the condition of the channel might have been changed since the last it was checked. The node's neighbours will probe the node and collect the information about the send successes of the channel. The results from probing is used to decide if the new channel is better than the current channel. LPBR is informed and updates its knowledge on the condition.

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 invoked on three (cases? situations?); on initialisation, when receiving LPBR channel change message and when the packet delivery ratio (PDR) drops. ***NOT SURE

**** 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

As mentioned in section 3.2, 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 its current and new channel in its table to allow the channel to be restored if required. 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 restored to its current channel. It removes the new channel value from the table 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 update 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.

It is possible that the node itself ask for a channel change if its PDR is below a threshold. LPBR checks and updates the information of the channels and sends a change channel message. The steps as previously explained are repeated. ***NOT SURE? LPBR asks for the transmission and receiver rate periodically from all the nodes to decide a channel change.

LPBR sends a change channel message after ***some time to check the condition of the node's channel and to get updates on the channels. The node's neighbours will start probing on the channel which the result is used to decide the changes.

E. LPBR Strategies

There are two strategies that LPBR uses when sending a change channel message to the nodes; random change and two hop colouring strategy.

1) *Random Channel Selection* -: At initialisation, LPBR uses random change to decide which node and what channel the node should try to change into. The neighbours of the node will send probe messages to check the channel condition and the 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 table based on the probing result. LPBR will have an overall information of channels which will affect its decision when sending a channel change message to the node. The advantages of this is LPBR can be certain that the channel that it choose for a node to change into will have better throughput than the current channel. Through probing result, LPBR knows how much better a certain channel is compared to other channels as this information is kept in LPBR table.

2) *Two Hops Neighbour Strategy* -: 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 send on the correct transmitting channel and return to the listening channel. When a node gets the channel to change into, it first checks its 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.

IV. IMPLEMENTATION

how multichannel is implemented? in RPL? route discovery? neighbour discovery? types of tables being kept? LPBR tables? other nodes tables?

V. EVALUATION

ours, RPL + ContikiMAC, MiCMAC

A. Experimental Setup

//methodology? key metrics?

B. Comparison

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

VI. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES